MIRAPOINT®

Mirapoint Administration Protocol Reference

You can use the Mirapoint administration protocol or the command-line interface (CLI) to manage your system. The low-level administration protocol is intended for scripting, which provides a way to do batch administration tasks over the network (for example, using the PERL interface described in Appendix B, Automating Administration with Perl). The CLI is optimized for interactive use and has a more forgiving syntax. The low-level protocol requires you to provide all command parameters. (Chapter 1, About the Protocol explains the administration protocol in detail.)

Access to the two management mechanisms:

- Mirapoint administration protocol: port 10143
- CLI: port 10144 or default telnet port

This book serves as the reference manual for the administration ptotocol. Other sources of information:

- For an HTML version of this manual and previous Messaging Operating System (MOS) releases: http://support.mirapoint.com
- For a list of valid administration commands: Type *tag* `Help` within the administration protocol.
- For documentation about the CLI, type `Help` inside the CLI.

Protocol 4.1.5
March 2009
Part No. 010-00713f

This manual supports MOS 4.1.3-GA and subsequent releases until replaced by a newer edition.

Contents

Contents 5

Preface 31

About the Mirapoint Documentation 31

Typographic Conventions 31

Revision History 32

Version 4.1.3—September 2008 32
Version 4.1.2—August 2008 32
Version 3.10.2—August 2008 32
Version 4.0.4—October 2007 33
Version 3.8.3—November 2006 33
Version 3.8.2—September 2006 33
Version 3.8.1—July 2006 33
Version 3.8.0—April 2006 33
Version 3.7.4—February 2006 34
Version 3.7.2—November 2005 34
Version 3.7.1—September 2005 34
Version 3.7.0—August 2005 34
Version 3.6—March 2005 34
Version 3.5—June 2004 35
Version 3.4—December 2003 36
Version 3.3—January 2003 37
Version 3.2—June 2002 37
Version 3.1.1—March 2002 38
Version 3.1.0—January 2002 38
Version 2.9.3—November 2001 38
Version 3.0—October 2001 39
Version 2.9.2—August 2001 39
Version 2.9.1—June 2001 39
Version 2.9—April 2001 39
Version 2.8.1—January 2001 40
Version 2.8—December 2000 40
Version 2.7—September 2000 40
Version 2.5—April 2000 41
Version 2.0—January 2000 42
Version 1.6—September 1999 42
Version 1.5—August 1999 42
Version 1.4—May 1999 43
Version 1.3—March 1999 43

Version 1.2.5—February 1999 43
Version 1.2.4—December 1998 43

1
About the Protocol 45

Connecting and Logging In 46
Using a Perl Interface 47
Commands 47
Domain Sensitivity 47
Command Format Restrictions 47
Quoted Strings 47
Literal Strings 48
Optional Arguments 48
Responses 49
Using Patterns 49

2
General Commands 51

Connecting and Disconnecting 51
The Login Command 51
The Logout Command 52
Getting Help 52
The Get Command 52
The Help Command 53
The Set Command 53
Getting the System Software Version 54
The Version Command 54
Halting and Rebooting the System 55
The Halt Command 55
The Reboot Command 55
The Softreboot Command 56

3
The Admin Command 57

Subcommands 57
Get 57
Set 58

4

The Antivirus Command 61

Subcommands 61
Count 61
Get 61
Getmessage 62
Getversion 63
List 63
Set 64
Setmessage 66
Update 67

5

The Auth Command 69

Authentication Schemes 69

Subcommands 70
Count 70
Get 70
List 71
Set 72

6

The Autoreply Command 73

Subcommands 74
Clear 74
Get 74
Getexpiration 75
Getinterval 76
Getmode 77
Getstart 77
Set 78
Setexpiration 80
Setinterval 81
Setmode 81
Setstart 82

7

The Backup Command 85

Backup Levels 85

Subcommands 86

Abort 86
Full 86
Incremental 87
LevelX........ 88
Media........ 89
Selective 90
Status 90

8
The Brand Command 93

Subcommands 93
Add 93
Assign 94
Assigned........ 94
Change........ 95
Count........ 96
Countdomain 96
Delete........ 97
Get 97
List........ 98
Listdomain 98

9
The Calendar Command 101

Calendars 101

Subcommands 102
Addsubscribed........ 102
Countsubscribed 102
Deletesubscribed 103
Downsync 104
Exportcal 105
Get 106
Importcal 106
Listsubscribed 107
Set 108
Upsync 110

10
The Cluster Command........ 113

Configuring an N+1 Failover Cluster 113
Powering Down for Maintenance........ 114

Subcommands....114
Add....114
Attach....114
Count....115
Countmember....115
Delete....116
Detach....116
Get....117
List....117
Listmember....118
Scan....118

11
The Conf Command....121

Subcommands....121
Count....121
Disable....121
Enable....122
Enabled....125
Export....125
Get....126
Import....127
List....127
Set....128

12
The Cos Command....131

Disabling a Service....131
LDAP Extensions....132
Subcommands....132
Disable....132
Enable....133
Enabled....134
Getstatus....135
List....137

13
The Customercare Command....139

Subcommands....139
Get....139
Set....139

14

The Diag Command 141

Subcommands 141
Abort 141
Changer 141
Clear 142
Disk 142
Get 143
Netstat 144
Ping 145
Rescan 146
Set 146
Smtpconnect 147
Status 148
Tape 149
Tcptraceroute 150
Traceroute 151

15

The Dir Command 153

Directory Command Overview 153
Configuration Subcommands 154
ExportConfig 154
ImportConfig 155
Option Subcommands 156
Get 156
GetLogging 156
Set 157
SetLogging 158
Database Subcommands 159
AddDb 159
CountDb 159
DeleteDb 160
ListDb 160
AddDbSuffix 161
CountDbSuffix 162
DeleteDbSuffix 162
ListDbSuffix 163
CountDbOption 163
GetDbOption 164
ListDbOption 165
SetDbOption 165
Bulk-Data Subcommands 167

- ExportLdif 168
- ImportLdif 168
- ModifyLdif 170

Schema and Index Subcommands 171
- AddSchema 171
- CountSchema 173
- DeleteSchema 173
- GetSchema 174
- ListSchema 174
- SetSchema 175
- AddIndex 176
- CountIndex 177
- DeleteIndex 177
- GetIndex 178
- ListIndex 178
- SetIndex 179

Replication Subcommands 180
- AddReplica 180
- CountReplica 181
- DeleteReplica 182
- ListReplica 182
- AddRepHost 183
- CountRepHost 183
- DeleteRepHost 184
- ListRepHost 184
- CountRepOption 185
- GetRepOption 186
- ListRepOption 186
- SetRepOption 187
- Replicate 188
- Promote 189
- Repstat 189

Access Control Subcommands 190
- AddACL 191
- CountACL 192
- DeleteACL 192
- GetACL 193
- ListACL 193
- SetACL 194
- AddAclEntry 195
- CountAclEntry 196
- DeleteAclEntry 197
- GetAclEntry 197
- ListAclEntry 198
- SetAclEntry 199

16

The DI Command 201

Subcommands 201
Add 201
Count 202
Delete 202
Importsendmail 203
List 204
Membership 205

17

The DIentry Command 207

Subcommands 207
Add 207
Addbulk 207
Count 208
Delete 209
List 209

18

The Dns Command 211

Name Servers 211

Subcommands 211
Add 211
Count 212
Delete 212
Flushcache 213
Get 213
List 214
Lookup 214
Set 215

19

The Domain Command 217

Delegated Domains 217
Domain Sensitivity and the Current Domain 217

Subcommands 218
Add 218
Count 218
Delete 219

Get....219
Getcurrent....220
List....221
Set....221
Setcurrent....223

20

The Exception Command....225

Subcommands....225
Add....225
Count....227
Countattrs....228
Countprotocols....228
Delete....229
List....229
Listattrs....230
Listprotocols....231
Test....231

21

The Failover Command....233

Subcommands....233
Addport....233
Countport....234
Deleteport....234
Get....235
Listport....235
Reboot....236
Set....236
Shutdown....237

22

The Filter Command....239

Filters....239
Actions....239
Rules....240
System Junk Mail Rule....243
Filtering to Reduce DSN in Queue....243
Subcommands....244
Add....244
Change....248
Clear....248

Count ... 249
Delete ... 250
Export ... 250
Fetch ... 251
Get ... 252
Import ... 252
List ... 253
Move ... 254
Set ... 255
Test ... 256

23
The Fwd Command ... 259
LDAP Interaction ... 259
Subcommands ... 260
Get ... 260
Set ... 260

24
The Getmail Command ... 263
Subcommands ... 263
Add ... 263
Change ... 264
Check ... 265
Count ... 265
Delete ... 266
Get ... 267
List ... 267
Rename ... 268
Set ... 269

25
The Http Command ... 271
Subcommands ... 271
Get ... 271
Set ... 271

26
The Imap Command ... 275
IMAP Service Modes ... 275

Normal Mode....275
Proxy Mode....275

IMAP Namespaces....275

Subcommands....276
Get....276
Set....277

27

The Kerb4 Command....281

Generating Srvtab Data....281

Subcommands....281
Add....281
Count....282
Delete....282
Get....283
List....283
Set....284

28

The Kerb5 Command....287

Generating Srvtab Data....287

Subcommands....288
Add....288
Count....288
Delete....289
Get....289
List....290
Set....290

29

The Key Command....293

Subcommands....293
Count....293
Delete....293
Get....294
Getpublic....295
List....296
New....296
Set....297

30
The Ldap Command 299
LDAP Server Specifications 299
LDAP Query Specifications 300
Query Specification Names 300
Using Variables in Distinguished Names and Filters 302
LDAP Object Class Specifications 303
LDAP Access Profiles 303
Changing Your Primary LDAP Server 303
Subcommands 304
Add 304
Addaccess 304
Addmailhost 305
Addquarantinehost 305
Count 306
Countaccess 306
Countobjclass 307
Countmailhost 307
Countquarantinehost 308
Countqueries 308
Delete 309
Deleteaccess 309
Deletemailhost 310
Deletequarantinehost 310
Flushcache 311
Get 311
Getaccess 312
Getmailhost 313
Getobjclass 314
Getquarantinehost 314
Getquery 315
List 316
Listaccess 317
Listmailhost 317
Listobjclass 318
Listquarantinehost 319
Listqueries 319
Search 320
Set 321
Setobjclass 323
Setquery 324
Testquery 325

31
The License Command 329
Subcommands 329
Apply 329
Count 329
Fetch 330
Hostid 330
List 331
Override 332
Revoke 332
Status 333

32
The Locale Command 335
Locales 335
Subcommands 336
Count 336
List 336
Get 337
Set 337

33
The Log Command 339
Data Formats 339
Identifiers 340
Subcommands 353
Addroute 353
Countroutes 355
Deleteroute 356
Diagwatch 356
Get 357
Listroutes 357
Set 358
Watch 359

34
The Mailbox Command 361
Access Control Lists 361
Subcommands 362
Add 362

Copy 363
Count 365
Delete 366
Get 366
Getacl 367
List 368
Msgexpirenow 369
Publish 370
Rename 370
Set 371
Setacl 372
Sync 373
Undelete 374

35
The Maildom Command 377

Mail Domains 377
Subcommands 377
Add 377
Count 378
Delete 378
List 379

36
The Mailq Command 381

Using Patterns 381
Subcommands 382
Count 382
Delete 382
Get 383
List 384
Reject 386
Release 387
Retryall 387
Transfer 388

37
The Message Command 389

Subcommands 389
Count 389
Countargs 390

Countfacilities....390
Countpolicies....391
Get....391
Getarg....392
Getpolicy....392
List....393
Listargs....393
Listfacilities....394
Listpolicies....395
Send....395
Set....396
Setarg....398
Setpolicy....399

38

The Model Command....401

Subcommands....401
Get....401

39

The Mon Command....403

Subcommands....403
Clear....403
Disable....403
Enable....404
Enabled....404
Getthresh....405
Setthresh....406
Status....407

40

The Mtaverify Command....409

Mtaverify Open Issues....409

Subcommands....410
Add....410
Addmisbehavingmailer....411
Count....412
Countmisbehavingmailer....412
Delete....413
Deletemisbehavingmailer....413
Flush....414
Get....414

List 415
Listmisbehavingmailer 415
Set 416
Test 418

41
The Ndmp Command 421

Subcommands 421
Clear 421
DeleteTs 422
Estimatesize 423
Export 423
Get 424
Getts 424
Merge 425
Set 426

42
The Netif Command 429

Subcommands 429
Addalias 429
Addlogical 430
Addroute 430
Addtrustedip 431
Bind 432
Bindings 432
Bindlogical 433
Countaliases 434
Countlogical 435
Countmedia 435
Countroutes 436
Counttrustedip 436
Deletealias 437
Deletelogical 437
Deleteroute 438
Deletetrustedip 438
Get 439
Getdomain 439
Getlogical 440
Listaliases 440
Listlogical 441
Listmedia 442
Listroutes 442
Listtrustedip 443

Set....443
Setdomain....445
Setlogical....446
Unbind....447
Unbindlogical....447

43
The Ntlm Command....449

Subcommands....449
Add....449
Count....450
Delete....451
Export....451
Getdefaultdomain....452
Import....452
List....453
Listdomain....453
Setdefaultdomain....454

44
The Ntp Command....455

Date and Time Strings....455
Time Zones....455
Subcommands....456
Add....456
Count....456
Delete....457
Get....457
List....458
Set....458
Synchronize....459

45
The Patternlist Command....461

Subcommands....461
Count....461
Delete....461
Export....462
Import....463
List....464

46
The Pop Command 465
POP Service Modes 465
Normal Mode 465
Proxy Mode 465
Subcommands 465
Get 465
Set 466

47
The Quota Command 469
Subcommands 469
Get 469
Getpolicy 470
Set 470
Setpolicy 471

48
The Radius Command 473
Subcommands 473
Add 473
Count 473
Delete 474
Get 474
List 475
Set 476

49
The Relay Command 477
Networks 477
Subcommands 477
Add 477
Count 478
Delete 478
List 479

50
The Replicate Command 481
Setting up Replication 481
Implications for Backup 482
Questions and Answers 482
Subcommands 483
Cancel 483
Checkswitch 484
Clear 485
Finalize 486
From 486
Get 487
Restart 487
Set 488
Status 489
Switchover 490
Sync 491
To 492
Version 493

51
The Reputation Command 495
Reputations 496
Subcommands 496
Addfixed 496
Countfixed 497
Deletefixed 497
Getaction 498
Getdelay 498
Listfixed 499
Query 499
Setaction 500
Setdelay 502
Update 502

52
The Restore Command 505
Subcommands 505
Abort 505
All 505
Media 507
Selective 508

Status 508

53
The Role Command 511

Roles 511

Subcommands 511
Add 511
Count 512
Delete 513
List 513

54
The Schedule Command 515

Subcommands 515
Add 515
Count 516
Delete 516
Getcommand 517
Getdatestring 517
Getfrequency 518
List 518

55
The Service Command 521

Subcommands 521
Disable 521
Enable 522
Enabled 522
Listservices 523
Start 524
Started 524
Stop 525

56
The Smtp Command 527

Mirapoint Added Headers 528

Specifying Periods of Time 529

Subcommands 529
Addfilterhost 529

Addlistener 529
Addrblhost....530
Addrtest....530
Countfilterhost 531
Countlistener 531
Countrblhost 532
Deletefilterhost 532
Deletelistener 532
Deleterblhost 533
Disablelistener 533
Enablelistener 534
Flushloginbeforesmtp....534
Get....534
Hoststat....538
Listfilterhost 539
Listlistener 539
Listrblhost 539
Set....540
Testrblhost....548

57
The Snmp Command 551

Access Profiles....551

Traps 551

Subcommands....551
Add....551
Addtrap 552
Count 553
Counttrap....553
Delete 554
Deletetrap....554
Get....555
List 555
Listtrap....556
Set....557

58
The Spf Command 559

SPF Records in DNS 559

Subcommands....560
Add....560
Count 561
Delete 561

Get 562
List 562
Set 563
Test 563

59
The Ssh Command 565

Subcommands 565
Get 565
Set 565

60
The Sshkey Command 567

Secure Shell (SSH) Keys 567
Subcommands 567
Add 567
Count 568
Delete 569
List 569

61
The Ssl Command 571

Subcommands 571
Getauthrequired 571
Getcert 572
Getcsr 572
Getintca 573
Getmaxkeylen 573
Getminkeylen 574
Getversion 574
Newcert 575
Setauthrequired 576
Setcert 576
Setintca 577
Setmaxkeylen 578
Setminkeylen 578
Setversion 579

62

The Stat Command 581

Subcommands 581
Get 581

63

The Storage Command 601

Identifiers for Disks and Arrays 601

Subcommands 602
Addarray 602
Addspare 602
Alarm 603
Arrays 603
Battery 605
Configure 605
Configsan 606
Checkadd 607
Countpaths 608
Countpathstates 608
Countsan 609
Countsanport 609
Delete 610
Disks 611
Get 612
Getnas 613
Getport 613
Listpaths 614
Listpathstates 615
Listsan 616
Listsanport 617
Scan 617
Set 618
Space 619

64

The Trustedadmin Command 621

Network Specifiers 621

Subcommands 621
Add 621
Count 622
Delete 623
List 623

65
The Trustedhost Command 625
Subcommands 625
Add 625
Count 626
Counttypes 627
Delete 627
List 628
Listtypes 628

66
The Uce Command 631
Antispam 631
Rapid Antispam 632
Networks 632
Subcommands 633
Add 633
Addexception 633
Count 635
Countexception 635
Delete 636
Deleteexception 637
Exportexceptions 637
Getoption 638
Importexceptions 639
Junkmailsummaries 640
List 640
Listexception 641
Removerulegroup 642
Report 643
Setoption 643
Update 645
Version 645

67
The Update Command 647
Subcommands 647
Install 647
List 648
Uninstall 648

68

The Url Command 651

URL Classes and Instances 651

Subcommands 651

Add 651

Count 653

Delete 654

List 654

69

The User Command 657

Periods in User and DL Names 657

Note 657

Using Patterns 658

Subcommands 658

Add 658

Count 660

Delete 660

Get 661

List 663

Rename 664

Set 665

70

The Webmail Command 669

Subcommands 669

Get 669

Set 670

71

The Wordlist Command 673

How Wordlists Work 673

Subcommands 674

Count 674

Delete 675

Export 675

Import 676

List 677

A

Sample CLI and Protocol Sessions 679

Examples of CLI Use 679

Adding User Accounts 681

Sample Session 682

Adding Mailboxes 682

Sample Session 682

Creating Distribution Lists 683

Sample Session 683

B

Automating Administration with Perl 685

Overview 685

Installing Net::MirapointAdmin 685

UNIX-based Systems 685

Windows-based Systems 686

Method Reference 687

High-Level Interface 687

Low-Level Interface 691

Examples 691

Connection and Authentication 691

Commands and Responses 692

Exceptions 694

Debugging 694

C

Single Sign-On Portal Access 697

Single Message Server 697

Multi-Tier Environments 698

Implementing Multi-Tier SSO 699

Preface

About the Mirapoint Documentation

Documentation for all Mirapoint products is available through the Mirapoint Technical Library (MTL) on the Customer Support website:

`http://support.mirapoint.com/secure/MTL/MTL`

The MTL provides the Hardware and Software documentation for all supported Mirapoint releases and appliances, and the Support Knowledge Base. The Support site is accessible to all customers with a valid Support Contract. If your company has this but you need a Support login ID, email support-admin@mirapoint.com.

Typographic Conventions

Table 1 explains what different fonts indicate in this book.

Table 1 Typefaces Used in This Book

Typeface	Use	Example
Regular	Ordinary text	The email server organizes mailboxes hierarchically.
Bold	Definitions	A **mailbox** is a container that stores messages.
Italic	Emphasis and titles	Specify *at least two* DNS servers. See the *Administrator's Guide*.
`Typewriter`	Screen display text and command names	`Enter your IP address:`
`Typewriter Bold`	Text that you must type exactly as shown	**`Dir Listdb`**
`Typewriter Italic`	Variables that you substitute and type	*`your_IP_address`*

Revision History

Version 4.1.3—September 2008

- (New) `Storage Get Controller` command.

Version 4.1.2—August 2008

- Deprecated `Diag Rescan` and `Storage Scan` commands.
- (New) `Smtp Get Dsndisable` and `Smtp Set Dsndisable` commands.

Version 3.10.2—August 2008

- (New) `Smtp Get Dsndisable` and `Smtp Set Dsndisable` commands.
- (New) Subcommands for autoreply enhancements: `Autoreply Getexpiration/Getinterval/Getstart/Setexpiration/Setinterval/Setstart`.
- (New) Subcommands to help battle spam cannons: `Webmail Set Maxnumrecips/Maxmsgrate/Maxreciprate`.
- (New) `Schedulemode` command to enable Enterprise Calendaring—functionality where SMTP mail messages perform calendar updates.
- (New) Edited information for `Timeout` command.
- `Multienginebulkonly` to optimize the results of dual Antispam scanners. Also SMTP `Bannerdelay` to drop connection if sender fails to wait.
- Encryption for IMAP and POP connections with support for STARTTLS.
- Merged IMAP namespace (Cyrus and UW) works well for many email clients.
- `Storage Configsan Expandlun` enables dynamic SAN storage extension.
- NTLM authentication for Outlook; see Chapter 43, The Ntlm Command.
- Security features `Diag Smtpconnect`, `Smtp Set Identhost`, and `Ssh Set` port.
- Helpdesk administrator can now control UCE whitelists and blacklists.
- New `Patternlist` command to eventually replace `Wordlist` facility.
- `Ndmp Get Softwareversion` for NDMP validation across releases.
- `Failover Set Timeout` to delay standby takeover in a cluster with slow I/O.
- New `Complete` multiple MX lookup for `Mtaverify Set Reversemx`.
- `Storage Set Rebuildrate` to control speed of rebuilding RAID storage.
- License names normalized between Mirapoint price list and system software.
- Deprecated `Dir AddDbimage` and related LDAP image commands.
- Deprecated NIS authentication and related `Nis` commands.

Version 4.0.4—October 2007

- Multipath SAN support to avoid reboot in case of switch or fabric failure; see Chapter 66, The Storage Command.
- Removed all previously deprecated commands except `Virtdom(entry)`.

Version 3.8.3—November 2006

- Better documentation of `Wordlist` filtering.

Version 3.8.2—September 2006

- `Model Get Compliance` to check RoHS status.
- Rapid Antivirus quarantine with autorelease controlled by `Quarantinedelay`.

Version 3.8.1—July 2006

- BakBone NetVault is now a supported NDMP data management application.
- Local storage, and NAS or SAN, may exceed 1 TB (terabyte).
- Multiple listeners for SMTP connections; see `Smtp Addlistener`.
- Disclaimer filtering and `in-ldap-group` match type.
- Multiple message send, new `Message` locales (for Junk Mail Manager).
- NDMP history speedup.

Version 3.8.0—April 2006

- LDAP autoprovisioning now works for Junk Mail Manager (JMM) accounts.
- External server monitoring is now disabled by default, but administrators can enable it with the `Mon Enable Netmonping` command.
- New GUI terms for whitelist, blacklist, and whitelistTo: Allowed Senders, Blocked Senders, and Allowed Mailing Lists.
- Authenticated POP support (APOP) by setting secret keys with `Key New Pop` and creating trust relationships with `Trustedhost Add Popgroup`.
- Two new `Smtp Set Nomasquerade` options for `Disposition-Notification-To` and `Return-Receipt-To`.
- `Conf Enable Ldapuuid` to store the Mirapoint unique user ID (UUID) in LDAP. Related to this for multi-tier support is `User Get Uuid` to retrieve the UUID.

Version 3.7.4—February 2006

- MailHurdle triplet database can be divided between multiple servers by running `Mtaverify Add` on participating servers. Also, `Mtaverify Set Allowentireip` defaults to On, allowing all messages from an IP address once one has passed.
- New `FILTER.NOTIFY.*` facility enables user-definable message notification for domain level filters.
- The `Smtp Set Rbl Ignorerelays` option tightens RBL by checking all hosts even if they are in the relay list.
- Additional IMAP Proxy statistics available through the `Stat` command.

Version 3.7.2—November 2005

- Multiple Antivirus packages can be run on one system, Sophos then F-Secure.
- Added `Netif Set Primaryport` to change the default network interface.
- New `Smtp Set Maxmsgcnt` to limit number of messages in a connection. Also `Smtp Set Dsnretrycnt` to limit delivery status notifications.
- More header fields supported by `Smtp Set Nomasquerade`.
- New logging event `UCE.MESSAGE.JUNKMAIL` for greater convenience.

Version 3.7.1—September 2005

- Ethernet NIC failover controlled by `Netif Addlogical` and related commands. This is a software enhancement that works on all multi-NIC systems.
- Automatically generated suspect list for MailHurdle due to Antispam scanning. See `Smtp Set Ucesuspectlist` and `Mtaverify Set Checksuspectlist`.
- Streamlined MailHurdle checking that can ignore recipient(s) triplet content. See `Mtaverify Set Ignorerecipients`.
- Enhanced mail queue management with `Mailq Transfer` number of tries.

Version 3.7.0—August 2005

- Corporate Edition WebMail and Calendar are available as `EnterpriseUi` in the `Conf`, `Cos`, and `Http` commands. License required.
- Support for F-Secure virus scanning has been added as an alternative so Sophos. Once licensed you can control operation with `Antivirus Set Fsav`.
- Tivoli Storage Manager is supported as one of the backup options for `Ndmp Set`.
- The Legato NetWorker backup service cannot be enabled.

Version 3.6—March 2005

- New `Ndmp` commands to perform selective restore from image: `Merge`, `Set Dma`, `Set Logprotocol`, `Export`, `Clear`.
- Deprecated the `Networker` command.

- Protocol extensions, mostly to the `Ldap`, `Uce`, and `User` commands, to support Junk Mail Manager.
- New `Message` command provides localizable welcome message formatting.
- Box-wide user count (including domains) with `User Count *@*`.
- Inserted SMTP header line `X-XMS` to track message filtering state. New `Smtp` setting `Nomasquerade` to ignore certain header fields.
- New `Key` command to manage secure keys for login to Mirapoint systems.
- The `Trustedhost` command now supports type `Mtagroup`.
- `Calendar Addsubscribed` (and so on) to manage calendar subscriptions. Number and size of external subscriptions can be set.
- Regular expression support for filtering using `Regexmatches` keyword, and the ability to alter spam score using the `Modspamscore` action.
- `X-junkmail-info` header line showing abbreviated Principal Edition score components.
- Ability to insert arbitrary header line using `Extraheader=` filtering option.
- Outbound spam scanning can be enabled using `Conf` option, `Antispam-all`.
- `Uce Removerulegroup` command to delete an installed Antispam rule group.
- `Mtaverify Addmisbehavingmailer` provides access to known-good mailer list.
- New Quarantine administrator role to manage domain-level quarantine filters.
- `Url Add` searching additions, MatchUser and MatchGroup.
- Multiport failover monitoring with `Failover Addport` (and so on).
- New set of `Cluster` commands to support N+1 failover, in which a group of servers can economically use one spare head to support initial-system failover.
- IMAP performance improvements, and shared folders accessible by WebMail, neither of which affect external protocol.

Version 3.5—June 2004

- New `Fastpath Direct` option for SMTP to support DirectPath™2 operation, and a `Storage Set IdeCache` option to disable IDE caching (the default).
- MailHurdle antispam facility controlled by the new `Mtaverify` command, with new `Trustedhost` command for security, and `Uce` recipient whitelisting.
- New `X-DSN` headers that can be filtered against to trim away bounce messages.
- `License Fetch` for all platforms, not just RazorGate 100.

Version 3.4—December 2003

- Remote filtering and filtering out of the queue, so a reject becomes a bounce. Related to this is a new `Cos Antivirus` setting, and event `DIAG.FILTER.STATUS` for the `Log` command.
- New `Ldap Setquery` specifications for mail group `allowedBroadcaster` and `allowedDomain`, for restricting who can send to an LDAP mailing list.
- Recursive mailbox copy with proxy authentication and SSL: see parenthesized list items in the various arguments to the `Mailbox Copy` command.
- Fallback from primary to secondary name in `Ldap Setquery user:Fullname`, for instance from `cn` to `displayname` or vice versa.
- Save and restore of system configuration with `Conf Import` and `Export`.
- APOP (authenticated POP) support, including fallback, with `Pop Auth Set`.
- New `Exception` command generalizing protocol exceptions, for SMTP only.
- Group Calendar without LDAP, with the `Calendar Set Groupmode` command. Better repeating event modification, and weekly or monthly event summaries.
- The `Ldap Setquery` query specifications now all default to the contents of the `User:Publishedname` specification, to promote consistency and reduce typing.
- For WebMail and related applications, setting `Conf Enable DerivedomainUrl` automatically supplies the delegated domain when users log in.
- POP and IMAP protocol logging, `DIAG.POP.PROTOIN` and `DIAG.IMAP.PROTOIN`. Also logging `MTA.MESSAGE.NEWMAIL` when messages arrive.
- Monitoring of system resource usage by task categories (no related commands). New `Mon Status` command to show outstanding alerts. New panic diagnostics.
- Remote monitoring of various types of servers, controlled from `Mon Setthresh`.
- System reset with `Softreboot`, `Dns Flushcache`, `Diag Tcptraceroute`.
- Junk mail delivered to POP mailboxes can be tagged “Spam” in the subject line; see the `Uce Setoption Spamprolog` command. Spam threshold now adjustable.
- The LDAP attribute `miDefaultJunkmailFilter` governs the automatic junk mail rule for `Cos Enable Antispam`. Logging of antivirus and antispam filters.
- User-level blacklisting now allowed by `Uce Addexception` command.
- New `Wordlist` command to support corporate or banned-word lists in filters.
- Ability to filter binary MIME parts using the `:bodydecodedbinary` keyword.
- HTTP `Ldapproxy` mode so WebMail servers can be kept behind the firewall, with optional SSL security and URL routing.
- POPS (secure POP using SSL) support for `Getmail`, with fallback to POP.
- The `Update Install` command now consults `Conf Set Httpproxy`, if applicable.
- Antivirus software can now be disabled and enabled with the `Conf` command. Scan failures are logged as an `ANTIVIRUS.*.SCANFAILURE` event.
- `Conf Enable Uidmigrate` preserves POP user IDs during system migration.

- New `Getmail` statistics for requests, messages, checks, and various failures.
- `Storage Checkadd` previews result of `Addarray` or `Addspare` command.
- New `Wiretap` action for the `Filter` command, like the `Redirect` action except that the default is to deliver, and bounces are not returned to the sender. Also a `Quarantine` action, and new priority setting for domain filters.
- The `Log Watch` system can now save data for seven days, rather than just one. Events were added for Administration protocol, system alerts, user and domain filter actions, remote delivery, overquota notification, and session ID logging.
- `Model Getserial` so customers can remotely determine hardware serial number.
- `Conf Enable Ldapgui` to enable the LDAP user provisioning web-browser tools.
- Fastpath improvements; see `Smtp Set` (near the end) for current limitations.
- New `Storage Battery` commands to manage RAID battery calibration.
- New `Mailbox Addindex` and `Deleteindex` commands for archive support.
- COS global default with the "@" name in LDAP.
- More settings, including All and Local, for `Smtp Set Ldapmasqsender`.
- Unannounced setting for `Locale Set`, useful for Asian character encodings.
- New Miradmin settings for `Http Set Root`.

Version 3.3—January 2003

- Antispam interfaces: `Uce` subcommands for whitelist and blacklist exceptions, and spam recognition updates. This is `Conf`-enabled and COS-controlled.
- Message undelete without LDAP via `Mailbox Set Undeletequota` command.
- Quota `Setpolicy` subcommand to override and change overquota messages.
- Autoreply `Setmode` subcommand to have automatic replies sent to all senders.
- HTTP `Set Cookies` subcommand setting for strengthened browser validation.
- SMTP set to preserve CNAME during routing; send Radius user name as typed.

Version 3.2—June 2002

- Directory Server enhancements: slave to master promotion, chaining and proxy authentication, RDN integrity, replication logging, password hash, user ACL.
- New `Getmail` command to fetch external POP mail.
- SMTP fast path for greater message delivery performance.
- Additional `Calendar Set` flags to control group calendars.
- Autoreply and WebMail settings stored in LDAP.
- New `Filter` rules for selectively deleting certain types of attachments.
- `Autoreply` and `Fwd` (forward) commands no longer mutually exclusive.
- The COS licensing requirement was dropped.

Version 3.1.1—March 2002

- New `Stat Get` cache, battery time, battery status, fan, and CPU temperature (for main and standby units) to support Series 400 and 4000 systems.
- New `Storage` commands for preliminary testing of SAN support.

Version 3.1.0—January 2002

- Ability to filter encoded attachments and international character sets using the `:bodydecoded` keyword of the `Filter Add` command.
- Support for login before SMTP in multi-tier environments. This feature requires a MIM schema including `miMembername` on the LDAP server, then you must run several commands including `Smtp Set Loginbeforesmtp` on message proxies. See the *Administrator's Guide* for details.
- NFS file systems can be mounted using the `Storage Mountnas` command, and prepared for NDMP backup with the `Storage Nasprep` command.
- Virus checking for each user can be controlled with the COS `antivirus` setting.
- Administrators can enable user-transparent LDAP password updating with the `Conf Enable Ldappassupdate` command.
- Administrators can allow users to browse and undelete discarded messages by creating a `mailUndeleteQuota` LDAP attribute and COS-enabling `msgundelete` for that user. A new `Mailbox Undelete` command is also available.
- New `Stat Get` variables are available to study SMTP performance.
- The `Dir AddDbimage` command takes a snapshot of the LDAP database, while `Dir RevertDbimage` rolls back the database to the time of a snapshot.
- You can arrange autoreply for messages sent to a user's alternate address by setting the `user:Localaddr` LDAP query (`Ldap Setquery`).
- The `Fwd Set` command now accepts multiple forwarding addresses separated by commas, even in the GUI.
- New `Log` events exist for mailbox `MESSAGE.APPEND` and `MESSAGE.EXPUNGE`.
- You can set a daily limit on the number of delivery status notifications (bounce message loops) with the `Smtp Set Dsnlimit` command.
- Rescan of SCSI bus for tape devices possible using `Diag Rescan` command. Support for tape drives and autochangers was enhanced in the 3.0 release.
- Message aging (in the 3.0 release) was extended with a command to request immediate expiration of old messages, `Mailbox Msgexpirenow`.
- Ability to restore branding data with the `Networker Recover Brand` command.

Version 2.9.3—November 2001

- New `Brand` subcommands for publishing multiple brands for different domains.
- New `Antivirus` subcommands for controlling Sophos virus protection.
- New `Schedule` subcommands for running services at specific times (CLI in 3.1).

- Configuration of RBL servers using `Smtp Addrblhost` and related commands.
- Import and export of address books with `User Get` and `Set Wmaddrbook`.
- Backport of `Loginbeforesmtp` (see version 3.1.1).

Version 3.0—October 2001

- Gigabit Ethernet support was added into the `Netif` command (on port 3).
- User suspend allowed with LDAP, controlled by `Conf Enable Ldapexception`.
- Mail forwarding from LDAP, controlled with `Conf Enable Ldapforward`.
- Message aging was implemented, controlled with `Cos Enable Msgexpiration`.
- Deprecated virtual domains; see `Virtdom` and `Virtdomentry` commands.
- New `Diag` subcommands for `Changer`, `Set`, `Get`, and `Clear`.
- The `Ssl Getintca` and `Setintca` commands added to support Intermediate Certifying Authority Certificate. The `Ssl Newcert` command was modified to accept arguments with different formats.

Version 2.9.2—August 2001

- LDAP routing on MX records with `Smtp Set LdapMXrouting`.
- LDAP routing attempted for every message with `Smtp Set Ldaprouting All`.
- Reject bogus inbound mail with `Smtp Set SenderIsValidRecipient Reject`.
- Selective subdomain-level routing with `Ldap Set SubdomainRoutingLevel`.

Version 2.9.1—June 2001

- Support for Web calendar, WAP calendar, and iMode mail.
- Overquota handling may be set with `Quota Setpolicy` command.
- Console password can be set by `Conf Enable Consolelogin` command.
- Outbound content filtering distinguished by `local` and `nonlocal` keywords.
- SMTP security options `Recipientcheck` and `Senderisvalidrecipient`.
- Directory live import with "`l`" flag of `Dir ImportLdif`.
- May require reboot verification after `Conf Enable Verifyreboot`.
- Selectable IMAP namespace, Cyrus or University of Washington.

Version 2.9—April 2001

- New `Dir` command with many features to manage LDAP Directory Server.
- Additional `Stat Get` parameters to support M2500 and M300 hardware.
- `Loginbeforesmtp` setting to allow non-relay users to send mail after logging in.
- Many new event identifiers added to `Log` facility, plus `Diagwatch`.

- Extended flags for Fileinto and Keep actions of the Filter command.
- Smtp Addrtest for testing message addresses.
- Mailq Get Envelope for viewing envelopes of queued messages.
- Ability to set the local mail router with Smtp Set Lmr command.
- Full control of SMTP authentication with Smtp Set Smtpauth command.
- Alternate SMTP ports with keywords Listenport and Connectport.
- Locale Set Loginfooter command to control browser's list of languages.
- Fuser Ldapadd extension for interoperability with NDS E-directory.

Version 2.8.1—January 2001

- New Calendar command to manage user schedules and give email notification.
- Ability to insert "for" header using Smtp Set Receivedforhdr command.
- Offers language selection before login with Locale Set Loginfooter command.

Version 2.8—December 2000

- New Cos command to manage class of service facility, with subcommands Getstatus, Disable, Enable, Enabled, and List.
- Added Diag Netstat subcommand.
- Ability to forward message excerpts with the Filter command.
- Non-anonymous binding now available in Ldap command.
- Added License Count and License List to help manage Mirapoint licenses.
- New Log command to manage error logging subsystem, with subcommands Addroute, Countroutes, Deleteroute, Get, Listroutes, Set, and Watch.
- New items (CPU, memory, Ethernet, chassis) in Model command.
- Implemented static routes in Netif command, with Addroute subcommand.
- Added Nis Add, Count, Delete, and List to manage persistent NIS servers.
- Added Smtp Hoststat subcommand.
- New items (connection port, no relays for SmtpAuth) in Smtp command.
- New items (RAID monitoring, system load, idle) in Stat Get command.
- Added Uce Get and Uce Set for Trend eManager facility, since discontinued.

Version 2.7—September 2000

- Added the Cleartextout and Sslout security schemes to the Admin, Imap, Pop, and Smtp commands.
- Added the Customercare keyword to the Conf Enable and Disable commands.
- Added the Customercare command.

- Added the Ping and Traceroute subcommands to the Diag command.
- Added the Signature parameter to the Domain Get and Domain Set commands.
- Added support for domain-based filtering to the Filter command, and added the Continue and Reject keywords.
- New Farm features for Messaging Infrastructure Manager (MIM).
- Added the Http command.
- Added the Kerb5 command.
- Added the Search subcommand to the Ldap command.
- Enhanced syntax of host argument to the Ldap Add, Count, Delete, and List commands to allow secure protocol and port number to be specified.
- Added the User:Fullname, User:Loginid, and User:Quota query specifications to the Ldap command.
- Added the Autoprovision keyword to the Ldap Get and Set commands.
- Added the Override subcommand to the License command.
- Added the Copy, Get, and Set subcommands to the Mailbox command.
- Added the Retryall subcommand to the Mailq command.
- Added pattern matching support for the Mailq Delete and Reject commands.
- Added Abort, Recover, and Status subcommands to the Networker command.
- Added the Getpolicy and Setpolicy subcommands to the Quota command.
- Added the Radius command.
- Added the Ssl command.
- Added the Url command.
- Enhanced the password argument of User Add to specify password encoding.
- Added the Login parameter to the User Get and Set commands.
- Added the User Rename command.

Version 2.5—April 2000

- Desupported the Mon subcommands, deprecated in version 1.4.
- Added the Timeout parameter to the Admin Get and Admin Set commands.
- Added Passmin and Passchars parameters to the Auth Get and Set commands.
- Added the Level0 through Level9 subcommands to the Backup command.
- Added the Flushcache and Testquery subcommands to the Ldap command.
- Added the Cachetimeout, Compatv1routing, and Ldif parameters to the Ldap Get and Set commands.
- Added Mailgroup:Members, Mailgroup:Owner, and User:Groupmembership keywords to the Ldap Getquery and Ldap Setquery commands.
- Added the Getthresh and Setthresh subcommands to the Mon command.

- Added the Ndmp command.
- Added the Addalias, Countalias, Deletealias, and Listalias subcommands to the Netif command.
- Added the Networker command.
- Added the Listservices subcommand to the Service command.
- Added the Ldapmasqsender and Sendercheck parameter to the Smtp Get and Smtp Set commands.
- Added the Webmail command.
- Enhanced pattern support for the User List and Mailq List commands.
- Deprecated Ldap Set Userdn and Ldap Get Userdn commands.

Version 2.0—January 2000

Made available with system software release 2.0, when numbering was aligned.

- Deprecated the following Admin subcommands:
  - Admin Add—Replaced by Role Add Admin
  - Admin Count—Replaced by Role Count Admin
  - Admin Delete—Replaced by Role Delete Admin
  - Admin List—Replaced by Role List Admin
- Added the Conf command.
- Added the Dns Lookup command.
- Added the Domain command.
- Added the Filter command.
- Added the Add, Count, Delete, and List subcommands to the Kerb4 command.
- Added the Realm keyword to the Kerb4 Get and Set commands.
- Added the Role command.
- Added the Interval keyword to the Autoreply Get and Set commands.
- Added the Raid keyword to the Model Get command.
- Added Forward, Autoreply, and Filter to the User Get Rights command.

Version 1.6—September 1999

Made available with system software release 1.7.

- Added the Diag command.
- Added Ldaprouting and Maxrecip keywords to Smtp Get and Set commands.
- Added Tape keyword for device parameter to Backup and Restore commands.

Version 1.5—August 1999

Made available with system software release 1.6.

- Added the `Locale` command.
- Added the `Maildom` command.
- Added the `Model` command.
- Added the `Netif Get` and `Netif Set` commands.
- Added the `Sshkey` command.
- Added the `Server` keyword to the `Nis Get` command.
- Added the `Date` keyword to the `Ntp Set` command.

Version 1.4—May 1999

Made available with system software release 1.5.

- Deprecated all subcommands of the `Mon` command except `Mon Clear`.
- Added the `Admin Get` and `Admin Set` commands.
- Added the `Auth` command.
- Added the `Backup Media` and `Restore Media` commands.
- Added the `Security` keyword to `Imap Get` and `Imap Set` commands.
- Added the `Kerb4` command.
- Added the `Ldap` command.
- Added the `Netif` command.
- Added the `Security` keyword to the `Pop Get` and `Pop Set` commands.
- Added the `Auth` keyword to the `User Get` and `User Set` commands.

Version 1.3—March 1999

Made available with system software release 1.2.

- Added the `Mailq` command.
- Added the `Backup Abort` and `Restore Abort` commands.

Version 1.2.5—February 1999

Made available with system software release 1.1.

- Added the `Backup` and `Restore` commands.
- Added the `Trustedadmin` command.
- Added the `Report` keyword to the `Mon` command for weekly reports.

Version 1.2.4—December 1998

Base protocol made available with system software release 1.0.

1

About the Protocol

Using the Mirapoint administration protocol, you can write scripts to do batch administration tasks on your Mirapoint system over the network. Unlike the administration command-line interface (CLI), which is optimized for interactive use, the low-level administration protocol is intended for scripting.

The CLI supports command-line editing: press the up arrow to repeat previous lines, and the left arrow to edit the current line. The CLI also supports command completion when you press the Tab key. The protocol level supports neither.

Ways that the administration interface, which is script-friendly but uses a less-forgiving syntax, differ from the CLI include:

- Every command you issue must be preceded with a tag, which is essentially a simple text string (no blanks allowed) that the system will echo throughout the system administration logs.

  Tag considersations:

  - All your tags can be the same. "TAG" and "." are commonly used.
  - You can use one tag for per script, such as the name of the script.
  - You can vary the tag name to assist in script debugging.
- You cannot have trailing blanks after the last command argument; the system interprets the blank as the separator before the next argument.
- The administration interface does not prompt for parameters. It assumes that they are part of the command line.

In the CLI, your prompt will display in the following convention, depending on your particular environment:

server.company.com>

Administration Interface Examples

```
tag login administrator admin
* tag 2b188b38ec740820a3a1ggd
tag OK User logged in

1 Webmail Set Timeout 30
1 OK Completed

2 Autoreply Set "sophia" "Out of Office" "I will return 31 May"
2 OK Completed
```

CLI Examples

```
User: administrator
Password:
OK User logged in

eng1.companyx.com> webmail set timeout 30
OK Completed

eng1.companyx.com> autoreply set sophia
Subject: Out of Office
Enter Autoreply message, finish with '.' on a line by itself
.
OK Completed
```

All examples in this manual use the administration protocol syntax.

Connecting and Logging In

For the CLI, connect to the usual `telnet` port 23. For the administration protocol service, connect to port 10143. The Mirapoint server responds with:

```
* OK Mirapoint-hostname admind version server ready
```

In this response, *`Mirapoint-hostname`* is the hostname of your Mirapoint system, and *`version`* is the administration server version.

At this point you're connected to the administration server, but you haven't logged in—your connection is in a **non-authenticated** state. In this state, you can run only the `login` and `logout` commands (see Commands on page 47 for details). From this state, you enter **authenticated** state by using the `Login` command.

Each user account has at least one **privilege level** corresponding to **roles** that the user is assigned (see Chapter 53, The Role Command). Roles determine which commands you may issue:

- Administrator—Can issue all administration protocol commands.
- Domain administrator—Can issue a subset of administration commands relevant to a delegated domain (see Chapter 19, The Domain Command).
- Helpdesk administrator—Can issue a subset of administration commands for handling day-to-day user requests, such as changing passwords and adding members to distribution lists.
- Backup operator—Can issue all commands necessary to perform system backups. This is a read-only role that cannot change the system in any way.
- User—Can issue only commands affecting your own user account.

In this book, each command description includes a Privilege Levels section that lists the roles that may issue the command.

Using a Perl Interface

Mirapoint professional services wrote a Perl library called **Net::MirapointAdmin** for administering our messaging systems. For more information see Appendix B, Automating Administration with Perl on page 685.

Commands

A protocol command looks like one of the following, where:

```
tag command arguments
tag command subcommand arguments
```

- *tag* is a text string that identifies the command. You must use a tag for all commands. While the protocol does not require it, using a unique tag for each command is recommended. Alphanumeric characters are preferable in tags, although some non-alphanumeric characters are allowed.
- *command* is one of the commands described in this book. Command names are case-insensitive. You must specify a command; it is not optional.
- *subcommand* is one of the subcommands described in this book. Subcommand names are also case-insensitive. Most commands have subcommands.
- *arguments* is a command-specific list of space-separated strings. Each has its own fixed number of arguments. Not all commands take arguments.

Domain Sensitivity

Some commands behave differently if a delegated domain is current (see Domain Sensitivity and the Current Domain on page 217). In particular, commands that affect mailboxes, user accounts, and distribution lists are all domain-sensitive. Some commands are not allowed at all when a delegated domain is current.

In this book, each command description includes a Domain Sensitivity section that describes any effect a current delegated domain has on the command's behavior.

Command Format Restrictions

- No leading or trailing whitespace is allowed. There must be exactly one space separating *tag* from *command*, *command* from *subcommand*, and *subcommand* from *arguments*, which must also be separated by single spaces.
- Commands must terminate with CRLF (carriage return followed by newline).
- Command and subcommand names are case-insensitive, as are most arguments.

Quoted Strings

You can specify a command or subcommand argument that contains spaces, other whitespace characters (tab or newline), and open or close parentheses, by enclosing the argument in double quotes (" "). For example, see `User Set`. In the CLI though, open and close parentheses do not need quoting.

Server responses are sometimes quoted and sometimes not, depending on context.

Literal Strings

Some commands and subcommands need input strings of arbitrary length. For these commands, the command line must end with a number followed by a plus (+) sign enclosed in curly braces, such as `{42+}`. This **literal string count** gives the number of octets (eight-bit bytes) in the literal string to follow.

Following the literal string count, you must send a line-ending CRLF, followed by the specified number of octets, followed by another CRLF. The literal string may or may not contain CR or LF in any combination. Here is literal string example:

```
tg Autoreply Set demo "Nobody Home" {52+}
I've gone fishing, and will reply on Tuesday. --Bill
```

The literal string mechanism is derived from the IMAP standard, RFC 2060. It is possible to have two literal strings in the same command, as in this example:

```
tg Unknown Set demo {12+}
Nobody Home
 {52+}
I've gone fishing, and will reply on Tuesday. --Bill
```

The numeric count of bytes in the string literal includes any CR or LF characters but not the beginning and ending CRLF. Depending on your software, CR and LF characters might be added automatically. The CLI prompts for strings.

Optional Arguments

Each command takes a fixed number of arguments. However, some commands allow you to pass optional parameter values to the command within a single quoted argument. Each optional parameter in such a parameter lists look like this, where *name* is the optional parameter and *value* is what you want to assign to it:

(*name*=*value*)

An optional parameter list might look like this:

```
"(name1=value1)(name2=value2)(name3=value3)"
```

The description of each command that uses optional parameter syntax defines the allowable parameter names and specifies any restrictions on their allowed values.

Escaping Special Characters

Because certain characters can have special meanings within optional parameter names and values, the following escape sequences are interpreted this way:

- `\(`—represents a single open parenthesis (character
- `\)`—represents a single close parenthesis) character
- `\\`—represents a single backslash \ character
- `\"`—represents a double-quote " character

Within an optional parameter name or value, a single backslash character followed by any character other than those listed here generates a syntax error.

Responses

When a command completes, the server responds with one of these responses, where *tag* matches your command, and *message* briefly explains the outcome:

- *tag* OK *message*
- *tag* NO *message*

For example, a successful command usually generates the response:

```
tag OK Completed
```

In the CLI, *tag* is absent. If a command fails for any reason, such as a syntax or permissions error, it might generate one of the following responses:

```
tag NO Missing required argument
tag NO Permission denied
```

Multi-Line Responses

Many commands respond with information requested by the client, such as a list of users or mailboxes. These are called **multi-line responses**. Each line of a multi-line response is prefixed with a `*` character, followed by the command's tag. Multi-line responses are followed by one of the completion responses described above. For an example, see the `User List` command.

All response lines, including the completion response, are terminated with CRLF.

Variable-Length Lists in Responses

Some commands, such as `Storage Arrays`, include variable-length lists of fields in multi-line responses. Each such list is preceded by an open parenthesis (`(`) and followed by a close parenthesis (`)`). For an example, see `Storage Arrays`.

Literal Strings in Responses

Some commands, such as `Autoreply Get`, return literal strings within multi-line responses. In a response, a literal string is introduced by a number enclosed in curly braces, such as `{42}`, just before the CRLF at the end of a line; this is the number of octets (8-bit bytes) in the literal string following, not including terminating CRLF.

Using Patterns

Several protocol commands allow you to specify a **pattern** for user, mailbox, or other names. Patterns are case-insensitive except where otherwise noted and can contain these wildcard characters:

- ?—(non-mailbox names only) matches any single character. For compatibility with the IMAP4 protocol, ? is not interpreted as a wildcard in mailbox names.
- *—matches zero or more characters of any kind. For mailboxes, this includes mailbox hierarchy separators (‘.’).
- %—(mailbox names only) matches zero or more characters, not including mailbox hierarchy separators. This wildcard is provided for compatibility with the IMAP4 protocol—it is interpreted as a wildcard only in mailbox names.

For example, the pattern `ann?`, passed to `User List`, would match the initial names, and the pattern `jo*`, passed to `Mailbox List`, would match the mailboxes below:

```
anna anne
jo joe john jon jon.bulk jon.personal
```

General Commands

General commands include those for establishing connections, getting help, finding the version number, and halting or rebooting the system.

Connecting and Disconnecting

The Login Command

Lets users authenticate themselves so they can run administration commands. The commands a user can execute when authenticated (logged in) depend on the roles assigned to the user. When an authenticated user has administration privileges, `Login` responds with a hexadecimal administration session identifier, which is currently unused by the protocol.

Connections to the administration service are limited to a rate of 16 K each second. If too many connection attempts occur, the service shuts down for one minute. Also, the number of connections are limited to system memory size (in MB) minus 128 and divided by 10. Above this limit connections are refused.

If an interactive login is obtained by means other than administration service, the log message “ALERT!: Interactive login detected” appears.

Syntax

```
tag Login username password
```

where:

- *username* is a valid user name.
- *password* is the password for this user.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- User

Domain Sensitivity

None

Example

```
2 Login Administrator admin
* 2 246b54def2e3672cc71b904
2 OK User logged in
```

The Logout Command

Closes the user's connection to the administration service. To use administration commands, you must reconnect and log in again.

Syntax

```
tag Logout
```

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- User

Domain Sensitivity

None

Example

```
3 Logout
3 OK Completed
```

Getting Help

The Get Command

Shows the screen (page) height for CLI help output. Returned value `auto-paging` means screen height is derived from terminal information, often 24.

Syntax

```
Get Page
```

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- User

Domain Sensitivity

None

Example

```
Get Page
auto-paging
OK Completed
```

The Help Command

Responds with a list of valid administration commands.

Syntax

```
tag Help
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Help
* 4 Admin List pattern start end
* 4 Admin Add admin_name
* 4 Admin Delete admin_name
* 4 Admin Count pattern

...remainder of long response...

4 OK Completed
```

The Set Command

Changes the screen (page) height for help output in the CLI.

Syntax

```
Set Page height
```

where screen *height* is the number of lines after which output pauses.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- User

Domain Sensitivity

None

Example

```
Set Page 39
OK Completed
```

Getting the System Software Version

The Version Command

Responds with the version number of the system software that your Mirapoint system is running.

Syntax

```
tag Version
```

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- User

Domain Sensitivity

None

Example

```
5 Version
* 5 Version 3.4.0.28
5 OK Completed
```

Halting and Rebooting the System

The Halt Command

Synchronizes the file system, gracefully shuts down all system services, and shuts off the system power. After you type `Halt`, the system waits a minute before shut down, giving you time to log out. On dual-head systems, use `Failover Shutdown`.

Once a system is powered off, you can restart it only by pressing the power button on the system's front panel keypad.

Syntax

tag `Halt`

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Halt
6 OK System will halt in one minute
```

The Reboot Command

Synchronizes the file system, gracefully shuts down all system services, and reboots the system. After you type `Reboot`, the system waits one minute before rebooting, giving you time to log out. On dual-head systems, use `Failover Reboot` if you want to select the other head as primary.

Syntax

tag `Reboot`

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
7 Reboot
7 OK System will reboot in one minute
```

The Softreboot Command

Quickly stops and restarts Mirapoint services, as would the `Reboot` command, but more drastically. Many customer problems can be resolved with soft reboot.

Syntax

```
tag Softreboot
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Softreboot
8 OK Completed
```

3 The Admin Command

The functionality of certain deprecated `Admin` subcommands has been replaced by the `Role` command. See Chapter 53, The Role Command and the *Administrator's Guide*.

Subcommands

Get

Responds with the value of the specified parameter.

Syntax

```
tag Admin Get parameter
```

where *parameter* is one of:

- `Security`—The returned value is a quoted, space-separated list showing the data-privacy schemes allowed for administration protocol and command-line interface connections (see `Admin Set`).
- `Timeout`—The administration service idle timeout in minutes, effective for all future administrator logins. The administration service closes CLI or low-level administration protocol connections that remains idle for this period.

Privilege Levels

- Administrator
- Helpdesk administrator (`Admin Get Timeout` only)
- Domain administrator (`Admin Get Timeout` only)
- Backup operator
- User (`Admin Get Timeout` only)

Domain Sensitivity

None

Example

```
4 Admin Get Timeout
* 4 10
4 OK Completed
5 Admin Get Security
* 5 "cleartext ssl"
5 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

tag `Admin Set` *parameter value*

where:

- *parameter* is one of:
  - `Security`—The data-privacy schemes allowed for administration protocol and command-line interface connections. When security is set to SSL only, low-level administration clients are required to connect using SSL, using the secure protocol SSL/TLS. The following *value* argument must be a quoted, space-separated list containing any of these data-privacy schemes:
    - `Cleartext`—No encryption of data for incoming connections.
    - `Cleartextout`—No encryption of data for outgoing connections.
    - `Ssh`—Secure-shell encryption of data transmitted using the command-line interface, but not supported for low-level administration protocol connections.
    - `Ssl`—Secure Sockets Layer (SSL, versions 2 and 3) and Transport Layer Security (TLS, version 1) encryption of data for incoming low-level administration connections. Not supported for command-line interface connections. **Note**: Admin SSL (and Sslout) connections are not HTTP connections; they are low-level administration connections protected by the same SSL/TLS technology widely used by HTTPS connections. For information about how to protect HTTP admin connections, see the details for `Http Set` in Chapter 25, The Http Command.
    - `Sslout`—SSL for outbound low-level administration connections, but not supported for command-line interface connections. SSH and SSL are available only if you have licenses for them installed on your system.
  - `Timeout`—The administration service idle timeout in minutes, effective for all future logins. The administration service closes any CLI or low-level administration protocol connection that remains idle for this period. The default timeout is 10 minutes, and the minimum timeout is 3 minutes.
- *value* is the value to which you want to set *parameter*.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
**2 Admin Set Timeout 10**
2 OK Completed
**3 Admin Set Security "Cleartext SSL"**
3 OK Completed
```

4 The Antivirus Command

The `Antivirus` command configures the behavior of virus scanning libraries on a Mirapoint system. Different antivirus libraries are supported.

Antivirus scanning requires a license. Applying each Antivirus license automatically `Conf`-enables a scanner. Antivirus checking, always for both incoming and outgoing messages, can be allowed only for specific users by means of the `Cos` command.

If two or more antivirus libraries are licensed and enabled, they run in this order: Sophos Antivirus, F-Secure Antivirus, Rapid Antivirus.

Subcommands

Count

Counts the antivirus libraries currently active on the system. An antivirus library becomes active when it is installed and licensed.

Syntax

`tag` `Antivirus Count` *`pattern`*

where *`pattern`* must be either the null string (`""`) or asterisk (`*`).

Privilege Levels

Administrator

Domain Sensitivity

Top-level only: returns an error within a delegated domain.

Example

```
2 Antivirus Count ""
* 2 1
2 OK Completed
```

Get

Retrieves the settings that can be configured with the `Antivirus` command.

Syntax

tag `Antivirus Get` *vendor parameter*

where:

- *vendor* may be `Fsav`, `Rapid`, or `Sophos` to select F-Secure, Rapid, or Sophos antivirus scanning.
- *parameter* is one of those supported by set; see `Antivirus Set`.

Privilege Levels

Administrator

Domain Sensitivity

Top-level only: returns an error within a delegated domain.

Example

```
5 Antivirus Get sophos notifyrecipient
* 5 ON
5 OK Completed
```

Getmessage

Retrieves all message values configured by the `Antivirus` command.

Syntax

tag `Antivirus Getmessage` *vendor messagepart*

where *vendor* may be `Fsav`, `Rapid`, or `Sophos` to select F-Secure, Rapid, or Sophos antivirus scanning, and *messagepart* is one of those documented under `Antivirus Setmessage`.

Privilege Levels

Administrator

Domain Sensitivity

Top-level only: returns an error within a delegated domain.

Example

```
9 Antivirus Getmessage sophos recipientmessage
* From: administrator
* The %v virus was detected in attachment (%F) in email from %f (%d).
* Action taken: %a
9 OK Completed
```

Getversion

Retrieves antivirus software engine information. The output of this command is vendor-specific.

Syntax

tag `Antivirus Getversion` *vendor*

where *vendor* may be `Fsav`, `Rapid`, or `Sophos` to select F-Secure, Rapid, or Sophos antivirus scanning.

Privilege Levels

Administrator

Domain Sensitivity

Top-level only: returns an error within a delegated domain.

Example

```
6 Antivirus Getversion sophos
* 6 {1125}
Sophos Anti-Virus SAVI3 3.2.07.107
Pattern file: 3.96
Incremental patterns: ablnk-ae agob-adh ...
Mirapoint MIME engine: 041214
Mirapoint scan engine: 040709
Mirapoint AV updater: 2.0.3
Last updated: Sat Aug 13 09:15:06 2005
6 OK Completed
```

```
8 Antivirus Getversion fsav
* 8 {425}
F-Secure virus scanning  v3.5
Scan Engine: 5.300-0130
Pattern Number: 893
Scanner Version: 3.5-Build_mirapoint_1032
8 OK Completed
```

List

Shows the antivirus libraries currently active on the system. An antivirus library becomes active when it is installed and licensed.

Syntax

tag `Antivirus List` *pattern start count*

where *pattern* may be either the null string (" ") or asterisk (*), and *start* and *count* specify the beginning and duration points.

Privilege Levels

Administrator

Domain Sensitivity

Top-level only: returns an error within a delegated domain.

Example

```
4 Antivirus List "" "" ""
* 4 sophos
4 OK Completed
```

Set

Changes a given parameter for the antivirus library of a given vendor.

If you choose to notify system administrators, recipients, or senders about messages containing a virus, a default header from `administrator` with subject line saying “Virus Warning” precedes the notification message. Additionally, a customizable message body provides information about the contaminated email. The following fields may be substituted into the message:

```
%d Date
%F Attachment file name
%i Attachment index (sequence number)
%v Virus name
%t Recipients (Envelope-To)
%a Action (Use this as its own complete sentence)
%p Problem (“virus infected” or “scan failed”)
%f Sender (From)
%h Mail server hostname
%% Actual percent sign (%)
%? Anything not listed above generates syntax error
```

Syntax

tag `Antivirus Set` *vendor parameter value*

where:

- *vendor* may be `Fsav`, `Rapid`, or `Sophos` to select F-Secure, Rapid, or Sophos antivirus scanning.
- *parameter* is one of:
  - `Adminmessage`—Specifies a notification message sent to `virus-alerts` distribution list if `Notifyadmin` is enabled and a virus is found. Default: `The %v virus was detected in attachment (%F) in email from %f to %t. Action taken: %a`
  - `Notifyadmin`—If set to `ON`, sends an email to the distribution list named `virus-alerts` whenever a virus is detected. If set to `OFF`, no message is sent. The default is `OFF`.
  - `Notifyrecipient`—If set to `ON`, sends an email to the recipient when a virus is detected. If set to `OFF`, no message is sent. The default is `OFF`.
  - `Notifysender`—If set to `ON`, sends an email to the original sender when a virus is detected. If set to `OFF`, no message is sent. The default is `OFF`.
  - `Quarantineaddress`—If not null (`""`) sends a virus quarantine message, which is the original virus-contaminated message as a separate attachment,

to the given address. A quarantine address can help you analyze viruses. The default is not to send a quarantine message.

- `Quarantinedelay`—For Rapid Antivirus, hours of quarantine timeout. After this delay, messages are automatically released from quarantine, hopefully to be rescanned by other Antivirus libraries. The quarantine area is checked once an hour for potential releases, or if it contains more than 40,000 messages. Delay can be 1 to 24 hours; setting zero (0) resets to the default 8 hours.
- `Recipientmessage`—Specifies a notification message sent to the message recipient if `Notifyrecipient` is enabled and a virus is found. Default:
  ```
  The %v virus was detected in attachment (%F) in email from %f (%d).
  Action taken: %a
  ```
- `Scanlist`—List of attachments to scan. A value of "" means to scan all attachments. Otherwise, *`value`* should contain a space-delimited list of file extensions to scan. The default is to scan all attachments.
- `Sendermessage`—Specifies a notification message sent to the originating sender if `Notifysender` is enabled and a virus is found. Default:
  ```
  The message you emailed to %t, dated %d, contains the %v virus in
  the %F attachment.  Action taken: %a
  ```
- `Virusaction`—What action should be taken when a virus is found. The possible action values are as follows:
  - `Delete`—Using `Log` facility, record that a virus was located (identifier `ANTIVIRUS.*.VIRUSDELETED`) and discard the infected attachment.
  - `Ignore`—Log that a virus was found (`ANTIVIRUS.*.VIRUSFOUND`) and send the unchanged message, with virus, to its intended recipient.
  - `Clean:Delete`—If virus can be cleaned, log that a virus was removed (`ANTIVIRUS.*.VIRUSCLEANED`) and send the cleaned message to its intended recipient. If virus is not cleanable, log that a virus was located (`ANTIVIRUS.*.VIRUSDELETED`) and discard the infected attachment.
  - `Clean:Ignore`—If virus can be cleaned, log that a virus was removed (`ANTIVIRUS.*.VIRUSCLEANED`) and send the cleaned message to its intended recipient. If virus is not cleanable, log that a virus was found (`ANTIVIRUS.*.VIRUSFOUND`) and send the unchanged message, with virus, to its intended recipient.

- *`value`* is the setting for *`parameter`*. Maximum length of *`value`* data is 1 MB.

Privilege Levels

Administrator

Domain Sensitivity

Top-level only: returns an error within a delegated domain.

Example

```
2 Antivirus Set sophos notifyrecipient ON
2 OK Completed
```

Setmessage

Sets the message part inserted into infected emails or virus notifications. The data in Message-Part will be mime decoded and pre-processed with the % substitutions described below, and then re-encoded with the original encoding. The maximum length of the value parameter is one megabyte of data.

These message settings take precedence over their equivalents in `Antivirus Set`.

Syntax

`tag Antivirus Setmessage vendor messagepart value`

where *vendor* may be `Fsav`, `Rapid`, or `Sophos` to select F-Secure, Rapid, or Sophos antivirus scanning, and *messagepart* is one of those listed below. If value is given as null string (" ") then the various messages return to default or those established by `Antivirus Set`.

- `summarypart`—The single MIME part inserted at the top of infected emails.
- `deletedpart`—The single MIME part inserted in place of an attachment that was deleted because it was found to be infected.
- `adminmessage`—Specifies the notification message that gets sent to the administrator if `Notifyadmin` is On and a virus is found. This should be a complete RFC 822 message.
- `sendermessage`—Specifies the notification message that gets sent to the originator if `Notifysender` is On and a virus is found. This should be a complete RFC 822 message.
- `recipientmessage`—Specifies the notification message that gets sent to the recipient if `Notifyrecipient` is On and a virus is found. This should be a complete RFC 822 message.

Message data in RFC822 format is MIME decoded, then `%` substituted (see below) and returned to the original encoding. Supported `Content-Transfer-Encoding` types are `""`, `"7bit"`, `"base64"`, or `"quoted-printable"`. Any RFC822 part with a different encoding is passed through unchanged. If there is more than one infected attachment, any single LINE with the `%i` sequence is repeated for each attachment. Dynamic `%` substitutions are not possible within encoded headers (such as Subject). Multipart boundaries are replaced at run-time to guarantee uniqueness.

- `%d`—Date.
- `%F`—Attachment file name.
- `%i`—Attachment index.
- `%v`—Virus name (or reason for a scanner failure).
- `%t`—Envelope recipient (including Bcc, careful about making it user-visible).
- `%a`—Action, which should not be used as part of a sentence. Possible values: `cleaned`, `deleted`, `ignored`.
- `%p`—Attachment problem (replaced by "virus infected" or "scan failed").
- `%f`—Sender.

- `%h`—Mail server host name.

Privilege Levels

Administrator

Domain Sensitivity

Top-level only: returns an error within a delegated domain.

Example

```
9 Antivirus Setmessage sophos recipientmessage
* From: administrator
* The %v virus was detected in attachment (%F) in email from %f (%d).
* Action taken: %a
9 OK Completed
```

Update

Manually update the given antivirus engine's pattern files. This usually involves your Mirapoint system connecting to a Mirapoint (or F-Secure, Rapid, or Sophos) website to retrieve pattern update files. This command is useful for emergencies. Automatic updates are scheduled at regular intervals, hourly by default. Now that pattern updates no longer require restarting SMTP service (release 3.10 and later), Mirapoint recommends changing this interval to every 15 minutes.

Syntax

`tag` `Antivirus Update` *`vendor`*

where *`vendor`* may be `Fsav`, `Rapid`, or `Sophos` to select F-Secure, Rapid, or Sophos antivirus scanning

Privilege Levels

Administrator

Domain Sensitivity

Top-level only: returns an error within a delegated domain.

Example

```
7 Antivirus Update sophos
7 OK Completed
```

```
8 Antivirus Update fsav
8 OK Completed
```

```
9 Antivirus Update rapid
9 OK Completed
```

The Auth Command

The `Auth` command lets you specify the schemes that the Mirapoint system can use for authenticating user logins to various services. You can specify authentication schemes for the system's primary domain and (in certain cases) specifically for each delegated domain.

Authentication Schemes

An **authentication scheme** is a two-part string identifying an **authentication method** and an **authentication database** appropriate for the method. The two string parts are separated by a colon (`:`). Authentication schemes currently supported are:

- `APOP:LDAP`—Authentication is done using a login name and timestamp-coded password transmitted over the network. The POP server obtains the user's password from LDAP, codes it with the timestamp (**nonce**), and compares. APOP is specified in RFC 1725 and stands for Authenticated POP.
- `KERBEROS_V4:KERBEROS_V4`—Authentication is done using a Kerberos version 4 ticket transmitted encrypted over the network. The ticket is validated against the Kerberos database using the `Kerb4 Set` command.
- `KERBEROS_V5:KERBEROS_V5`—Authentication is done using a Kerberos version 5 ticket transmitted encrypted over the network. The ticket is validated against the Kerberos database using the `Kerb5 Set` command.
- `NTLM:NTLM`—Authentication is done using NTLM (NT LAN manager). For more information, see Help About Ntlm.
- `PLAINTEXT:KERBEROS_V4`—Authentication is done using a login name and password transmitted without encryption over the network. The login name and password are validated against the Kerberos database using `Kerb4 Set`.
- `PLAINTEXT:LDAP`—Authentication is done using a login name and password transmitted without encryption over the network. The password is compared to a record stored remotely in an LDAP database.
- `PLAINTEXT:LOCAL`—Authentication is done using a login name and password transmitted without encryption over the network. The password is compared to a record stored in the user database stored locally on the Mirapoint system.
- `PLAINTEXT:NIS`—Authentication is done using a login name and password transmitted without encryption over the network. The password is compared to a record stored remotely in the Network Information Service database.

- `PLAINTEXT:RADIUS`—Authentication is done using a login name and password transmitted without encryption over the network. The password is compared to a record stored remotely on the Radius server database. Radius authentication can be useful for non-ASCII login names.

Subcommands

Count

Responds with the number of available authentication schemes(see Authentication Schemes on page 69).

Syntax

tag `Auth Count` *pattern*

where *pattern* is currently ignored.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
2 Auth Count ""
* 2 9
2 OK Completed
```

Get

Responds with the value of the specified parameter.

Syntax

tag `Auth Get` *parameter*

where *parameter* is one of:

- `Default`—the default list of authentication schemes to be used in authenticating user account logins for the administration, POP, and IMAP services(see Authentication Schemes on page 69).
- `Passchars`—a set of characters, at least one of which must appear in new passwords for local users (see `Auth Set`).
- `Passmin`—the minimum required size of new passwords for local users (see `Auth Set`).

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

When a delegated domain is current, this command reports authentication schemes specified for the primary domain except for schemes requiring Kerberos or NIS.

Example

```
7 Auth Get Default
* 7 "apop:ldap kerb4:kerberos_v4 kerbe5:kerberos_v5 plaintext:local"
7 OK Completed
8 Auth Get Passchars
* 8 {27}
0123456789!@#$%^&*,./<>?;:|
8 OK Completed
9 Auth Get Passmin
* 9 8
9 OK Completed
```

List

Responds with a list of the available authentications schemes (see Authentication Schemes on page 69) in alphabetic order.

Syntax

tag Auth List *pattern start count*

where all parameters are currently ignored (specify arguments as "" or *).

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
3 Auth List "" "" ""
* 3 apop:ldap
* 3 kerberos_v4:kerberos_v4
* 3 kerberos_v5:kerberos_v5
* 3 ntlm:ntlm
* 3 plaintext:kerberos_v4
* 3 plaintext:ldap
* 3 plaintext:local
* 3 plaintext:nis
* 3 plaintext:radius
3 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

tag `Auth Set` *parameter value*

where:

- *parameter* is one of:
  - `Default`—the default list of authentication schemes to be used in authenticating user account logins for Mirapoint services.
    These authentication schemes are used to authenticate all users for whom the `User Get Auth` command returns ‘`default`’. You can override this setting for individual users using the `User Set Auth` command.
    The value for this parameter must be a quoted, space-separated list of authentication schemes. For a list, see Authentication Schemes on page 69).
    You may specify only one `PLAINTEXT` authentication method in this list. Plain text authentications include `PLAINTEXT:LOCAL`, `PLAINTEXT:LDAP`, and `PLAINTEXT:RADIUS`.
  - `Passchars`—a set of characters, at least one of which must appear in new passwords for local users. This command affects only subsequent `User Set` commands affecting passwords stored in the Mirapoint system’s local user database. Existing passwords are not affected.
  - `Passmin`—the minimum required size of new passwords for local users. This command affects only subsequent `User Set` commands affecting passwords stored in the Mirapoint system’s local user database. Existing passwords are not affected.
- *value*—the value you want to assign to *parameter*.

Privilege Levels

Administrator

Domain Sensitivity

This command is allowed only when no delegated domain is current.

Example

```
4 Auth Set Default "PLAINTEXT:LOCAL KERBEROS_V4:KERBEROS_V4"
4 OK Completed
5 Auth Set Passchars "0123456789!@#$%^&*,./<>?;:|"
5 OK Completed
6 Auth Set Passmin 8
6 OK Completed
```

6

The Autoreply Command

The `Autoreply` command configures a Mirapoint system to send an automatic reply whenever mail arrives for a particular user. As an administrator, you can determine whether autoreplies will be sent upon receipt of spam.

Users can set up different autoreplies (different subject lines and content) for internal and external users, and they can determine the autoreply start and stop times.

For LDAP autoreply, the `mirapointMailUser` object class should allow the entry `miDeliveryOption`. The lines below enable LDAP autoreplies:

```
miDeliveryOption: autoreply
miDeliveryOption: mailbox
```

In the second line, `mailbox` means deliver the message as intended after autoreply. Replacing `mailbox` with `forward` says to forward the message instead of delivering it. In release 3.4 and later, the second line is optional, indicating autoreply without delivery or forward. (Earlier releases tried to prevent message loss.)

Current LDAP attributes for autoreply and what they specify:

- `miAutoreplySubject`—Autoreply subject. If a subject is not configured for external autoreplies, this string will be used as the subject line for replies to internal and external recipients.
- `miAutoreplyText`—Message body. If a message body is not configured for external autoreplies, this text will be used for replies sent to internal and external recipients.
- `miAutoreplyStart`—Start date.
- `miAutoreplyInterval`—Minimum time interval (`M` minutes or `D` days) between autoreplies to the same sender. Like `Autoreply Setinterval`, this defaults to 7 days, and reverts to the default 7 days if you specify an interval less than 5 minutes or greater than 365 days.
- `miAutoreplyExpiration`—Expiration date.
- `miAutoreplyExternalSubject`—Subject line for external autoreply.
- `miAutoreplyExternalText`—Message body for external autoreply.

```
miAutoreplySubject: On Vacation
miAutoreplyText: Contact my manager if you have any billing-related questions.
    I will be back in the office starting 16 February.
miAutoreplyStart 20080210080000Z
miAutoreplyInterval: 2d
miAutoreplyExpiration: 20080216160000Z
```

```
miAutoreplyExternalSubject: Out of Office
miAutoreplyExternalText: I will return 16 February.
```

Subcommands

Clear

Turns off automatic reply for the specified user, or the authenticated user if no user is specified. A user who doesn't have administration privileges is allowed to clear automatic reply for his own user account.

Note that there is a delay of several minutes between the time you issue this command and the time automatic reply actually stops. Messages received during this interval generate replies.

Syntax

```
tag Autoreply Clear username
```

where *username* (optional) identifies the user whose automatic reply configuration you want to see.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own user account)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Get

Responds with the current automatic reply configuration for the specified user, or the authenticated user if no user is specified.

A user who doesn't have administration privileges is allowed to get the automatic reply status of his own user account.

Syntax

```
tag Autoreply Get username
```

where *username*(optional) can be:

- The unique username of the user whose automatic reply configuration you want to see.
- A string of the form "`(user=`*`user`*`)(scope=`*`scope`*`)`" where:
  - *`user`* is the unique username. The *`user`* parameter is required in all cases and is implied if there are no parentheses.
  - *`scope`* is either `local` or `nonlocal`. The *`scope`* parameter defaults to `local` if a value is not provided. `local` is identical to the definition used by the `Smtp Set Recipientcheck` feature. `nonlocal` is defined as anything not local.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- User (can access only his or her own user account)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Response

`Get` returns the body of the user's automatic reply message as a literal string (see Literal Strings in Responses on page 49). If the response specifies two empty strings (`"" ""`), this means that automatic reply is disabled for the specified user account.

Example

```
1 Autoreply Get demo
* 1 "Nobody Home" {107}
Hi,

I'm out of the office for a few days. I'll respond to your mail
when I return.

Thanks,

Demo

1 OK Completed
```

Getexpiration

Responds with the autoreply expiration date for a specified user.

Syntax

tag Autoreply Getexpiration *username*

where *username* identifies the user whose automatic reply is configured with an expiration date.:

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own user account)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
2 Autoreply Getexpiration u1
* 2 20081015000000
2 OK Completed
```

Getinterval

Responds with the current automatic reply interval for the specified user, with one of the following suffixes to indicate units:

- d——days
- m——minutes

If the user does not have automatic reply enabled, the command generates a NO response. A user may get the automatic interval of his own user account.

Syntax

tag Autoreply Getinterval *username*

where *username* (optional) identifies the user whose automatic reply interval you want to see.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

- User (can access only his or her own user account)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
3 Autoreply Getinterval demo
* 3 7D
3 OK Completed
```

Getmode

See if autoreply is responding to additional (forwarded) or spam messages.

Syntax

tag `Autoreply Getmode` *mode*

where *mode* is:

- `Autoreplytoall`—Checks whether autoreplies are sent for forwarded messages.
- `Replyjunk`—Checks whether autoreplies are sent for spam emails.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
4 Autoreply Getmode Autoreplytoall
* 4 NO
4 OK Completed
```

Getstart

Responds with the start date of any autoreply message for a specified user.

Syntax

tag `Autoreply Getstart` *username*

where *username* identifies the user whose automatic reply is configured with a start date.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own user account)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
5 Autoreply Getstart u1
* 5 20081221030005
5 OK Completed
```

Set

Sets the automatic reply configuration for the specified user, or the authenticated user if no user is specified. The command prompts you to enter a subject line and body text for the reply message——you must terminate the body text with a single dot '.' character on a line by itself. Autoreply works in conjunction with forward.

Users who lack administration privileges are allowed to set automatic reply only for their own account. To avoid excessive automatic replies, the system only replies to messages addressed directly to that user (not by means of a distribution list). The system also keeps track of email senders to whom it has sent automatic replies, and sends only one automatic reply per week to a given address.

As of system software release 2.9.2, autoreply works with LDAP aliases. Before, autoreply would work only if mail was addressed to a user's published name. It now works even if mail is addressed to the user's alternate address.

LDAP user entries may have one additional attribute (besides the published name) listing alternate mail addresses for the user. Mail addressed to any of the addresses in this list of values will generate an automatic reply if autoreply is enabled for that user. Commonly used email address attributes include `mail`, `mailAlternateaddr`, and `mailLocaladdress`.

To turn off automatic reply, specify the empty string for both the *subject* and *body* parameters.

Syntax

```
tag Autoreply Set username subject body
```

where:

- *username*(optional) is one of the following:
  - Unique username of the user whose automatic reply configuration is to be set.
  - A string of the form "*(*`user=`*user)(*`scope=`*scope)*" where:
    - *user* is the unique username. The *user* parameter is required in all cases and is implied if there are no parentheses.
    - *scope* is either `local` or `nonlocal`. The *scope* parameter defaults to `local` if a value is not provided. `local` is identical to the definition used by the Smtp Set Recipientcheck feature. `nonlocal` is defined as anything not local.
- *subject* is the text for "Subject:" header of the reply message.
- *body* is the body text for the reply message. Ordinarily, you specify this as a literal string (see Literal Strings on page 48).

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own user account)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example 1

```
6 Autoreply Set demo "Nobody Home" {107+}
Hi,

I'm out of the office for a few days. I'll respond to your mail
when I return.

Thanks,

Demo

6 OK Completed
```

Example 2

```
7 Autoreply Set "(user=sophia) (scope=nonlocal)" "Out of Office" "I will return
    31 May"

7 OK Completed
```

Setexpiration

For the specified user, sets an expiration date for any autoreply message.

Syntax

```
tag Autoreply Setexpiration username datestring
```

where:

- *username* identifies the user whose automatic reply is configured with an expiration date.
- *datestring* can be:
  - A value of `0` (default). `0` disables expiration of the autoreply; that is, autoreply is enabled indefinitely.
  - A specified date and time. The value is of the form:
  - *yyyymmddhhmmss*

    where:

    – *yyyy* is the four digit year, such as `1999`.
    – *mm* is the two-digit month, such as `06`.
    – *dd* is the two-digit day of the month, such as `09`.
    – *hh* is the two-digit hour using 24-hour clock, such as `02` or `16`.
    – *mm* is the two-digit minute, such as `08`.
    – *ss* is the two-digit second, such as `05`.

    A value less than the current date allows configuration of a non-functional autoreply message that can be reactivated by setting a later date (or `0`). The time entered is relative to the user's time zone (if set) or the server's global time zone.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own user account)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
8 Autoreply Setexpiration bill 20081221030005
8 OK Completed
```

Setinterval

For the specified user, sets a minimum autoreply interval, which defaults to 7 days. If a user does not have autoreply enabled, this command generates a `NO` response. Users may set the automatic interval of their own user accounts.

This setting is lost when automatic reply is disabled for the user. If automatic reply is subsequently re-enabled for the user, reply interval reverts to the default 7 days.

Syntax

tag `Autoreply Setinterval` *username interval*

where:

- *username* identifies the user whose automatic reply interval you want to set.
- *interval* is the time interval that you want to specify for the user with one of the following suffixes to indicate units:
    - `d`——days
    - `m`——minutes

  If you specify an *interval* less than 5 minutes or greater than 365 days, the setting reverts to the default 7 days.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
9 Autoreply Setinterval demo 20m
9 OK Completed
```

Setmode

Have autoreply respond to spam and forwarded messages.

Syntax

tag `Autoreply Setmode` *mode state*

where

- *mode* is one of:
  - Autoreplytoall——If *state* is No (default), an autoreply is sent only if the recipient's email address is found in the To or Cc lines of the message. If *state* is Yes, an autoreply is sent even if the recipient's email address is not found in the To or Cc lines of a message. The system still attempts to avoid autoreplying to distribution lists by checking if Precedence is Junk or Bulk, if a List header exists, or if Auto-submitted has a value other than No.
  - Replyjunk—If *state* is On, autoreplies are sent for messages that are tagged as spam. If state is Off (default), autoreplies are not generated for spam emails. Spam is defined as any message with a UCE value greater than the administrator-configured server threshold (which is set with Uce Setoption Threshold command).
- State—Setting for *mode*.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
10 Autoreply Setmode Autoreplytoall Yes
10 OK Completed
```

Setstart

For the specified user, sets a start date for any autoreply message.

Syntax

tag Autoreply Setstart *username datestring*

where:

- *username* identifies the user whose automatic reply is configured with a start date.
- *datestring* can be:
  - A value of 0 (default), which means begin immediately.
  - A specified date and time. The value is of the form:
  - *yyyymmddhhmmss*

  where:

– `yyyy` is the four digit year, such as `1999`.
– `mm` is the two-digit month, such as `06`.
– `dd` is the two-digit day of the month, such as `09`.
– `hh` is the two-digit hour using 24-hour clock, such as `02` or `16`.
– `mm` is the two-digit minute, such as `08`.
– `ss` is the two-digit second, such as `05`.

A value less than the current date means begin immediately. The time entered is relative to the user's time zone (if set) or the server's global time zone.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own user account)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
11 Autoreply Setstart vhugo 0
11 OK Completed
```

The Backup Command

The `Backup` command saves important system and user data on a Mirapoint system to a local tape device attached to that system, or to a storage device on a remote Solaris server using the Remote Magnetic Tape protocol. The Solaris command that implements this protocol is `rmt(1M)`.

Before using remote `Backup` and `Restore`, you must configure your Solaris system so the Mirapoint system has permission to access the remote storage device. Do this by adding a user account on Solaris, "mira" by default, and creating a `.rhosts` file to grant this user remote access from the Mirapoint system. For more information, refer to the Backup chapter in the *Administrator's Guide*.

Backup Levels

The `Backup` command lets you perform the following levels of backup:

- Full—Backs up system configuration data and all mailboxes and email messages. This level is also known as Level 0.
- Incremental—Backs up system configuration data and those email messages received since the most recent full backup. This level is also known as Level 1.
- Levels 2 through 9—Backs up system configuration data and those email messages received since the most recent lower-level backup. For example, a Level 3 backup backs up all messages received since the most recent Level 2, Level 1, or Level 0 backup.
- Selective—Backs up only the specified mailboxes.

Each `Backup` subcommand starts a backup operation and returns immediately; you may log out and log back in again any number of times without interfering with a backup. To monitor the progress of a backup, use `Backup Status`.

Backup starts by listing all directories in the filesystem. New messages that arrive after the start of a backup are ignored. Deleted and modified message might or might not appear in the backup, depending on their location and timing. Of course message changes after backup completion are never saved.

Subcommands

Abort

Aborts the backup operation identified by the specified job ID. To determine job ID, run `Backup Status`.

Syntax

tag `Backup Abort` *jobid*

where *jobid* is the job ID of the backup operation you want to abort. This *jobid* is case-sensitive, so use capitals as returned by `Backup Status`.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
1 Backup Abort Backup-5928
1 OK Completed
```

Full

Backs up system configuration data and all mailboxes and email messages to the specified storage device. This command starts the backup operation and immediately responds with a unique job ID identifying the operation. For the format of the job ID, see `Backup Status`.

Syntax

tag `Backup Full` *device blocksize*

where:

- *device* has one of the following formats:

  host-name:*device-name*

  user-name@*host-name*:*device-name*

  `Tape`

  where:

  - *host-name* is the host name of the remote system to which the backup device is attached, for RMT.
  - *device-name* is the full remote system path of the special file referring to the backup device.

    - *user-name* is the name of the user identity to assume *on the remote system* when doing the backup. This **backup user** must have write access to *device-name*. If you don't specify a backup user, the command uses the default name `mira`.
    - `Tape` is a special keyword identifying the tape drive attached to your Mirapoint system.
- *blocksize* is the block size in bytes to use in writing to the backup device. The null string ("") uses the default block size, 10240 bytes (10 KB) for RMT and 61440 bytes (60 KB) for tape. Tape block size is unalterable. This parameter may also contain a directive to back up just the system, user, mailbox, and DL information (not all messages) using the following syntax:

  `"(BlockSize=bytes)(nomessages=)"`

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
2 Backup Full sunw:/dev/rmt/0c 10240
* 2 Backup-5928
2 OK Completed
```

```
2 Backup Full sunw:/dev/rmt/0c "(BlockSize=10240)(nomessages=)"
* 2 Backup-5928
2 OK Completed
```

Incremental

Backs up system configuration data and those email messages received (and not yet deleted) since the last full backup. The backup data is stored on the specified storage device. This command starts the backup operation and immediately responds with a unique job ID identifying the operation. For the format of the job ID, see `Backup Status`.

Syntax

tag `Backup Incremental` *device blocksize*

where:

- *device* identifies the backup device (see `Backup Full`).
- *blocksize* is the block size in bytes to use in writing to the backup device. The null string ("") uses the default block size, 10240 bytes (10 KB) for RMT and 61440 bytes (60 KB) for tape. Tape block size is unalterable. This parameter may also contain a directive to back up just the system, user, mailbox, and DL information (not all messages). See `Backup Full` for details.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
3 Backup Incremental allansun:/dev/rmt/0c 10240
* 3 Backup-5954
3 OK Completed
```

LevelX

The `Backup Level0` command is equivalent to saying `Backup Full`. The commands `Backup Level1` through `Backup Level9` back up system configuration data and those email messages received (but not yet deleted) since the most recent backup of a lower level. For example, `Backup Level3` backs up all messages received since the most recent `Level2`, `Level1`, or `Level0` backup.

The backup data is stored on the specified storage device. These commands start a backup operation and immediately respond with a unique job ID identifying the operation. For the format of the job ID, see `Backup Status`.

Syntax

tag `Backup LevelX` *device blocksize*

where:

- *X* is the backup level, which must be a digit `0` through `9`.
- *device* identifies the backup device (see `Backup Full`).
- *blocksize* is the block size in bytes to use in writing to the backup device. The null string (`""`) uses the default block size, 10240 bytes (10 KB) for RMT and 61440 bytes (60 KB) for tape. Tape block size is unalterable. This parameter may also contain a directive to back up just the system, user, mailbox, and DL information (not all messages). See `Backup Full` for details.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
2 Backup Level1 allansun:/dev/rmt/0c ""
* 2 Backup-24417
2 OK Completed
```

Media

This command can:

- Report whether the specified backup operation is waiting for you to change the media
- Inform the backup system that you have changed the media to allow the specified backup operation to resume

For the format of the job ID, see `Backup Status`.

Syntax

tag `Backup Media` *keyword jobid*

where:

- *keyword* is one of:
  - `Changed`—Informs the backup system that you have changed the media and instructs the specified backup operation to resume
  - `Wanted`—Reports whether the backup operation is suspended waiting for you to change the media
- *jobid* is the job ID of the backup operation for which the media is being used. This *jobid* is case-sensitive, so use capitals as returned by `Backup Status`.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
4 Backup Media Wanted Backup-5928
* 4 Waiting for volume #2 for allansun:/dev/rmt/0c.
4 OK Completed
5 Backup Media Changed Backup-5928
5 OK Completed
```

Selective

Backs up only the specified mailboxes and subfolders to the specified device. This command starts backup operation and immediately responds with a unique job ID identifying the operation. For the format of the job ID, see `Backup Status`.

Syntax

```
tag Backup Selective device blocksize mailboxes
```

where:

- *`device`* identifies the backup device (see `Backup Full`).
- *`blocksize`* is the block size in bytes to use in writing to the backup device. The null string (`""`) uses the default block size, 10240 bytes (10 KB) for RMT and 61440 bytes (60 KB) for tape. Tape block size is unalterable.
- *`mailboxes`* is a quoted, space-separated list of mailbox names relative to the root of the mailbox hierarchy. User mailboxes are specified as `user.`*`username`*. Selective backup is recursive, so specifying `user.joe` also backs up all of Joe's subfolders. If a mailbox name contains a space, you must quote it again with a pair of \” backslash-quotes inside the double quotes. If this argument is given as `(type=brand)` then only contents of the branding directory `/usr/store/www` are backed up.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
6 Backup Selective allansun:/dev/rmt/0c "" "user.al user.sal"
* 6 Backup-5978
6 OK Completed

7 Backup Selective allansun:/dev/rmt/0c "" "user.al \"user.big dog\" user.sal"
* 7 Backup-5979
7 OK Completed
```

Status

Responds with the current status of all outstanding `Backup` operations. A completed operation is reported only once.

Syntax

```
tag Backup Status
```

Response

Each response line has one of these two forms:

`* tag Backup-num processed total messages`

or:

`* tag Backup-num status`

where:

- `Backup-num` is a unique job identifier (job ID) for the backup operation.
- *total* is the number of files that have been backed up so far.
- *status* is the operation status, which may be:
    - `started`—the system is preparing for backup
    - A message requesting that the backup media be changed (see `Backup Media`)
    - `done`—the operation completed successfully
    - An error message

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
8 Backup Status
* 8 Backup-5928 70001 of 71414
8 OK Completed
```

8

The Brand Command

The `Brand` command allows publication of one or more brands, associates brands with domains, and enables the sharing of brands between domains. Publication of a brand allows you to change appearance of a system and its domains.

To publish a brand, you add it and assign a domain to it. You can also list, count, and delete brands. A brand is shared if more than one domain is assigned to it.

Subcommands

Add

Publishes a brand of the name given. A brand of that name must not already exist; if the brand name exists, use `Brand Change` instead. Brand names can contain alphabetic and numeric characters. Case is unimportant, but if upper and lowercase characters are mixed, brands are listed in lowercase only. The brand name “system” is reserved and refers to the top-level system brand.

Brand characteristics are specified in a ZIP encoded file passed as an `ftp://` URL to the `Brand Add` or `Change` command. The file must be retrievable by FTP. Mirapoint suggests placing your branded graphic images in the `/extras` directory. However do not place pre-existing (shared or repeated) images there, because that causes brand publication to fail.

Privilege Levels

Administrator

Domain Sensitivity

None

Syntax

```
tag Brand Add brandname URL
```

where:

- ◆ *brandname* is the brand name (see above for naming rules).
- ◆ *URL* is the uniform resource locator of the brand’s ZIP file on an FTP server.

Example

```
2 Brand Add BrandX ftp://brand.example.com/brandX.zip
2 OK Completed

3 Brand Add BrandX ftp://juser:password@brand.example.com/brandX2.zip
3 NO brand already exists
```

Assign

Tell the Mirapoint system to use a named brand for a given domain. The "system" brand is used by the top-level domain. Other domains also use the `system` brand until you assign them to another brand. The top-level domain cannot be assigned to a named brand because it uses only the `system` brand.

Virtual domains (deprecated in Mirapoint release 3.0) always use the system brand. Only delegated domains can be associated with a brand.

To disassociate a brand from a domain, reassign that domain to the `system` brand.

Privilege Levels

Administrator.

Domain Sensitivity

None

Syntax

```
tag Brand Assign domain brandname
```

where:

- *`domain`* is the name of a delegated domain previously created with `Domain Add`.
- *`brandname`* is the name of a brand previously created with `Brand Add`.

Example

```
4 Brand Assign soap.com brandX
4 OK Completed

5 Brand Assign suds.com brandX
5 OK Completed
```

Assigned

Retrieves the named brand to which a particular domain is assigned. If a domain was reassigned to the `system` brand, returns "NO... not assigned" instead.

Privilege Levels

Administrator.

Domain Sensitivity

None

Syntax

```
tag Brand Assigned domain
```

where *domain* is the name of a delegated domain with an associated brand.

Example

```
6 Brand Assigned soap.com
* 6 brandX
6 OK Completed
```

Change

Replaces the brand of a given name with the appearance and content specified in the given file. See `Brand Add` for rules on naming a brand.

Brand characteristics are specified in a ZIP encoded file passed as an `ftp://` URL to the `Brand Change` or `Add` command. The file must be retrievable by FTP. Mirapoint suggests placing your branded graphic images in the `/extras` directory. However do not place pre-existing (shared or repeated) images there, because that causes brand publication to fail.

Privilege Levels

Administrator

Domain Sensitivity

None

Syntax

```
tag Brand Change brandname URL
```

where:

- *brandname* is the brand name (see above for naming rules).
- *URL* is the uniform resource locator of the brand's ZIP file on an FTP server.

Example

```
6 Brand Change BrandX ftp://juser:password@brand.example.com/brandX2.zip
6 OK Completed

7 Brand Change BrandX ftp://brand.example.com/brandX2.zip
7 OK Completed

8 Brand Change BrandZ ftp://brand.example.com/brandZ.zip
8 NO brand does not exist
9 Brand Add BrandZ ftp://brand.example.com/brandZ.zip
```

```
9 OK Completed
```

Count

Displays the number of brands on the system. The “system” brand counts as a one.

Privilege Levels

Administrator

Domain Sensitivity

None

Syntax

tag `Brand Count` *pattern*

where *pattern* is a string to match brands. The asterisk (*) is a matching wildcard, and the empty string (" ") matches everything.

Example

```
10 Brand Count ""
* 10 3
10 OK Completed
```

Countdomain

Displays the number of domains sharing a particular brand.

Privilege Levels

Administrator.

Domain Sensitivity

None

Syntax

tag `Brand Countdomain` *pattern*

where *pattern* matches a brand name.

Example

```
11 Brand Countdomain brandX
* 11 2
11 OK Completed
```

Delete

Removes the brand of a given name. If a brand is in use by a domain, it cannot be removed. To show domains currently using a brand, use the `Brand Listdomain` command. If you remove the “system” brand, all domains using that brand revert to factory defaults.

Privilege Levels

Administrator

Domain Sensitivity

None

Syntax

```
tag Brand Delete brandname
```

where *`brandname`* is the name of a brand previously created with `Brand Add`.

Example

In this example, a different brand must be assigned to `soap.com` and `suds.com` before the originally assigned `brandX` can be deleted.

```
15 Brand Delete brandX
15 NO brand is used by a domain
16 Brand Assign soap.com brandZ
16 OK Completed
17 Brand Assign suds.com brandZ
17 OK Completed
18 Brand Delete brandX
18 OK Completed
```

Get

Get the ZIP file of a named brand by copying brand data from the appliance to the specified `ftp://` URL. FTP must have write permission on the incoming directory.

Privilege Levels

Administrator

Domain Sensitivity

None

Syntax

```
tag Brand Get brandname URL
```

where:

- *brandname* is the name of a brand previously created with `Brand Add`.
- *URL* is the uniform resource locator of the brand's ZIP file on an FTP server. Some type of FTP login access is required.

Example

```
12 Brand Get brandZ ftp://brand.example.com/incoming/brandZ2.zip
12 OK Completed
```

List

Display all the named brands matching the specified pattern, in alphabetic order. The system brand is always shown.

Privilege Levels

Administrator

Domain Sensitivity

None

Syntax

tag `Brand List` *pattern start count*

where:

- *pattern* is a string to match brands. The asterisk (*) is a matching wildcard, and the empty string (`""`) matches everything.
- *start* is the number of the first to list. The empty string (`""`) indicates zero (`0`).
- *count* is the number of brands to list. The empty string (`""`) again indicates all. If *count* is greater than the number of brands, returns as many as possible.

Example

```
13 Brand List "" "" ""
* 13 brandX
* 13 brandZ
* 13 system
13 OK Completed
```

Listdomain

Display all the domains assigned to a brand. Output is sorted by brand name and then by domain name.

Privilege Levels

Administrator.

Domain Sensitivity

None

Syntax

`tag` `Brand Listdomain` *`pattern start count`*

where:

- *`pattern`* matches a brand name.
- *`start`* is the number of the first domain to list. The empty string (`""`) means `0`.
- *`count`* is the number of domains to list. The empty string (`""`) implicitly means all. If *`count`* is greater than the number of brands, returns as many as possible.

Example

```
14 Brand Listdomain brandX "" ""
* 14 brandX soap.com
* 14 brandX suds.com
14 OK Completed
```

The Calendar Command

The `Calendar` command is used by higher-level interface software for manipulating the scheduling of Mirapoint users. This facility allows for migration or backup of user calendars and server state, and tuning of server parameters.

Personal WebCal requires a license. Group Calendar requires an additional license, and multi-tier Group Calendar also requires a Mail Routing license. WebCal and WebMail are separately licensed; one does not depend on the other.

Mirapoint Calendar supports dates between midnight January 1st 1970 (GMT) and December 31st 2035 only.

Calendars

The relationship between a “calendar” and a messaging user account is like that between a mailbox and a user account. However, a user can have only one calendar. Like user logins, calendar UIDs should be *user@domain* for delegated domain users.

The `Get` and `Set` commands retrieve and change server-wide calendaring variables, such as quotas and default calendar configurations. The `Exportcal` and `Importcal` commands retrieve and change per-calendar parameters such as user preferences. The `Upsync` and `Downsync` commands synchronize calendaring events using the vCalendar protocol, and are the basis for synchronization with other calendaring agents and related services such as backup and restore.

In this chapter, the term “calendar” refers to a database containing:

- User information, such as email reminder address.
- User preference settings.
- Calendar state, including the time last modified.
- All event and alarm information for the calendar.

For group calendaring to work in multi-tier environments, you must use LDAP authentication, not `plaintext:local`. Group calendar relies on LDAP mailgroups (distribution lists) to determine group membership.

Calendars should be backed up, so they can be restored in case of disaster.

Subcommands

Addsubscribed

Adds the named calendar to the subscribed list for all users in a domain.

Syntax

```
tag Calendar Addsubscribed calname title type
```

where:

- *calname* is the name of a user who owns this calendar with subscribed bit set, or an external calendar. External calendars are not verified when added.
- *title* is a descriptive name for display to the user. UTF8 strings are allowed. May not be longer than 256 characters.
- *type* is one of the following:
  - `Required`—Users are automatically subscribed, and cannot unsubscribe.
  - `Suggested`—Users are automatically subscribed, but can unsubscribe.
  - `Optional`—Display to users as a calendar to which they might subscribe.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, this command applies to the system's primary domain.

Example

```
9 Calendar Addsubscribed http://expo.org/sched.ical "Trade Shows" Optional
9 OK Completed

10 Calendar Addsubscribed holidays "Company Holidays" Suggested
10 OK Completed
```

Countsubscribed

Counts the calendars added to the subscribed list for this domain.

Syntax

```
tag Calendar Countsubscribed pattern
```

where *pattern* must be * or the null string (“ “) to match all subscriptions.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, this command applies to the system's primary domain.

Example

```
11 Calendar Countsubscribed ""
* 11 2
11 OK Completed
```

Deletesubscribed

Removes the named calendar from the subscribed list for users in a domain.

Syntax

```
tag Calendar Deletesubscribed calname
```

where *calname* is the user name of the calendar owner, or an external calendar, either previously added with `Calendar Addsubscribed`.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, this command applies to the system's primary domain.

Example

```
13 Calendar Deletesubscribed http://expo.org/sched.ical
13 OK Completed
```

Downsync

Get event, to-do, and alarm data from the WebCal server. This command applies to individual calendars and retrieves all scheduling data from the calendar database.

Syntax

tag Calendar Downsync *calname protocol*

where:

- *calname* is the name of the calendar to be downloaded from the server, and is the same as the name of the user who owns the calendar.
- *protocol* is the protocol used to represent the calendaring information. Currently, the only accepted protocol is `vcal`, which communicates using the vCalendar protocol.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- Users (can access only their own calendar)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, this command applies to the system's primary domain.

Example

```
2 Calendar Downsync juser vcal
* 2 {292}
BEGIN:VCALENDAR
PRODID:-//Mirapoint Calendar
VERSION:1.0
BEGIN:VEVENT
UID:20010426T0000Z-0@when.mirapoint.com
DTSTART:20010426T070000Z
DTEND:20010426T080000Z
DCREATED:20010426T220700Z
LAST-MODIFIED:20010426T220700Z
PRIORITY:3
SUMMARY:test
CLASS:PRIVATE
END:VEVENT
END:VCALENDAR

2 OK Completed
```

Exportcal

Gets parameters of a calendar. This command applies to individual calendars and is used to retrieve parameters for import into another calendar.

Syntax

```
tag Calendar Exportcal calname
```

where *calname* is the name of the calendar to be retrieved, and is the same as the name of the user who owns the calendar.

The output of `Exportcal` is a string-literal containing a CRLF-separated list of parameter/value pairs. The output of this command should be treated as an opaque object that can be passed to the `Importcal` command (see below).

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- Users (can access only their own calendar)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, this command applies to the system's primary domain.

Example

```
3 Calendar Exportcal juser
* 3 {367}
flags:
uiflags:DONECONFIG
uidfltview:w
timezone:America/Los_Angeles
reminderarr:
reminderarrpager:
daystart:01:00
dayend:13:00
weekstart:0
reminderdiff:0
pagerreminderdiff:0
summarytimeemail:-1
summarytimepager:-1
version:v_nojs
calrdar:AT_PUBLIC:
calwrar:AT_ONLYME:
dfltrdar:AT_ONLYME:
dfltwrar:AT_ONLYME:
frbsrdar:AT_ONLYME:
schdwrar:AT_ONLYME:

3 OK Completed
```

Get

Obtains the current value of a server-wide calendar parameter.

Syntax

```
tag Calendar Get parameter
```

where *parameter* is one of the available settings. For details see `Calendar Set`.

Privilege Levels

- ◆ Administrator
- ◆ Helpdesk administrator
- ◆ Domain administrator
- ◆ Backup operator
- ◆ Users (can access only their own settings)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, this command applies to the system's primary domain.

Example

```
4 Calendar Get maxnumevents
* 4 2000
4 OK Completed
```

Importcal

Sets parameters of a calendar. This command applies to individual calendars and is used to import parameters into a calendar.

Syntax

```
tag Calendar Importcal calname value
```

where:

- ◆ *calname* is the name of the calendar to import, and is the same as the name of the user who owns the calendar.
- ◆ *value* should be a string-literal containing a CRLF-separated list of parameter/value pairs in the same format as output by the `Exportcal` command.

Privilege Levels

- ◆ Administrator
- ◆ Helpdesk administrator

- Domain administrator
- Users (can access only their own calendar)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, this command applies to the system's primary domain.

Example

```
5 Calendar Importcal juser {367+}
flags:
uiflags:DONECONFIG
uidfltview:w
timezone:America/Los_Angeles
reminderarr:
reminderarrpager:
daystart:01:00
dayend:13:00
weekstart:0
reminderdiff:0
pagerreminderdiff:0
summarytimeemail:-1
summarytimepager:-1
version:v_nojs
calrdar:AT_PUBLIC:
calwrar:AT_ONLYME:
dfltrdar:AT_ONLYME:
dfltwrar:AT_ONLYME:
frbsrdar:AT_ONLYME:
schdwrar:AT_ONLYME:
```

```
5 OK Completed
```

Listsubscribed

Display all calendars added to the subscribed list for this domain.

Syntax

tag `Calendar Listsubscribed` *pattern start count*

where *pattern* must be * or the null string (" ") to match all subscriptions, *start* is the beginning position, and *count* is the number to display.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, this command applies to the system's primary domain.

Example

```
12 Calendar Listsubscribed * "" ""
* 12 http://expo.org/sched.ical "Trade Shows" Optional
* 12 holidays "Company Holidays" Suggested
12 OK Completed
```

Set

Set changes the value of a server-wide or domain-wide parameter.

Syntax

`tag Calendar Set parameter value`

where:

- *parameter* is one of the following items to set:
  - Attachenabled—Whether email attachments are allowed. Yes (default) indicates attachments are enabled; No indicates attachments are disabled.
  - Attachquota—Specifies the maximum attachment size in kilobytes. Default quota is 10240 KB.
  - Daygranularity—Breaks the day into specific time intervals. Allowed values are 15, 30, 60. Other values are invalid. Default value is 60 minutes.
  - Dfltcalflags—Default calendar flags. This is a list of flag names separated by commas. The presence of a flag name implies value 1 (true), while its absence means 0 (false). White space is not allowed. If more than one flag name is present, the entire list must be enclosed in double quotes. Flag names are case-sensitive, and are as follows:
    - DAILYREM—send daily reminders.
    - EMAILREM—send per-event email reminders (the default).
    - MONTHLYREM—send monthly reminders similar to DAILYREM.
    - WEEKLYREM—send weekly reminders similar to DAILYREM.
  - Dfltdayend—Time at which the user's day finishes, specified as *hour*:*minute* where hour ranges from 0 to 24 and minute from 0 to 60. Default is 18:00.
  - Dfltdaystart—Time at which the user's day starts, specified as *hour*:*minute* where hour ranges from 0 to 24 and minute from 0 to 60. Default is 8:00.
  - Dfltuiflags—Default user interface flags for a new account. This is a list of flag names separated by commas. The presence of a flag name means it takes a value 1; its absence means 0. White space is not allowed. If more than one flag name is present, the flag list must be enclosed in double quotes. Flag names are case-sensitive, and are described below.

  - WEEKDAYSEP—Show separators between weekdays in the weekly view.
  - EVENTTEXT—Show event text within calendar.
  - TODOWITHCAL—Show to-do list on the main calendar page.
  - SUMMARYNXTDAYEMAIL—Set if the user wants next day's events in regular email format, unset for present day's events. Default is unset.
  - SUMMARYNXTDAYPAGER—Set if the user wants next day's events in pager format, unset for present day's events. Default is unset.
- Dfltview—Default calendar view. View values may be d, w, m, or h (daily, weekly, monthly, or horizontal-weekly). The default is weekly.
- Externalmaxnum—Sets the maximum number of external calendars to which each user can subscribe. Default is 10.
- Externalmaxsize—Sets the maximum size of each external calendar to which users can subscribe, in KB. Default is 256.
- Groupcalmode—Controls whether calendar uses LDAP, the local system, or both to define groups and find users. Possible modes for *value* are ldap, local, and all. The default mode is all for maximum compatibility. Site administrators should set mode to ldap or local to improve performance, when possible. If mode is set to all, LDAP takes precedence over local. Local DLs or LDAP mailgroups define calendar groups and resources; see Resourcegroup below. Users are located with Url Add Groupcalendar for LDAP or locally by user and full name search.
- Ldapmaxcount—Maximum number of records returned. Default is 100.
- Ldaptimeout—Asynchronous per-user LDAP search timeout in seconds. The default value is 15 seconds.
- Maxnumevents—Maximum number of events allowed per calendar. Zero (0) means to accept the default of 2000. See also Domain Set.
- Monthlysummaryday—Sets the day for monthly summaries. Values 1 or 2 mean the first or second day of the month, and so on. Values -1 or -2 mean the last or second-to-last day of the month, and so on. Default value is 1.
- Monthlysummarytime—Sets the time for monthly summaries. The value and default are the same as for Summarytimeemail.
- Pagerreminderdiff—Number of minutes before an event that a pager reminder should be generated. Allowed values: 5, 10, 15, 20, 25, 30. Other values are invalide. Default 5. Use Dfltcalflags to enable.
- Reminderdiff—Number of minutes before an event that an email reminder should be generated. Allowed values: 5, 10, 15, 20, 25, 30. Other values are invalid. Default 5. Use Dfltcalflags to enable.
- Removeafter—Number of days after which to remove expired events. Zero (0) means no limit and no removal. See also Domain Set.
- Resourcegroup—Specifies the name of a mailgroup stored on an LDAP server, or a distribution list stored locally, either of which may be used to obtain a list of calendar resources (meeting rooms, telephones, projectors, and so forth). Default value is the null string. For LDAP and local, *value* can be *resource*, packaged as a DL or mail group of resources. Prefixes ldap: or localdl: may be given to force one or the other.
- Schedulemode—Performs all calendar updates using SMTP email messages. Allowed values: email, server. Use email to enable. Use server to disable.
- Summarytimeemail—Time when summary (regular format) is sent. The value is given in 24 hour hh:mm format. Allowed values are 00:00-23:00. Currently only hour increments are supported, so the value should be 0:00,

1:00, up to 23:00. Other values are considered invalid. Default is the same as `dfltdaystart`, currently 8:00.
- `Summarytimepager`—Time when summary (compact format) is sent. Values are the same as for `Summarytimeemail`.
- `Timeout`—Calendar timeout in minutes, used by the graphical user interface. Default value is 360. See also `Domain Set`.
- `Weeklysummaryday`—Sets the day for weekly summaries. Value 0 means Sunday, 1 means Monday, and so forth to 6 meaning Saturday. Default is 0, which on Sunday produces a summary for the current week.
- `Weeklysummarytime`—Sets the time for monthly summaries. The value and default are the same as for `Summarytimeemail`.

- *`value`* is the setting. For possible values, see parameter descriptions above.

Privilege Levels

- Administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, this command applies to the system's primary domain.

Example

```
6 Calendar Set dfltview d
6 OK Completed

7 Calendar Set dfltcalflags "DAILYREM"
7 OK Completed
```

Upsync

Synchronizes event information for a particular user's calendar with the data input. This is the counterpart of the `Downsync` command. Data get imported in vCalendar format, a multi-line string. The synchronization may be done in two different ways: either by always overwriting existing data and removing any data in the database not present in uploaded data, or by rewriting data only if older than the uploaded data and retaining items not present in the uploaded data.

When fields are upsynched in vCalendar format, they do not necessarily have the same `ENCODING` properties as when they were downsynced. The calendar server chooses an appropriate `ENCODING` parameter, for instance quoted-printable.

Syntax

```
tag Calendar Upsync calname protocol options value
```

where:

- *calname* is the name of the calendar to be uploaded into the server, and is the same as the name of the user who owns the calendar.
- *protocol* is the protocol used to represent the calendaring information. Currently, the only accepted protocol is `vcal`, which communicates using the vCalendar protocol.
- *options* controls the way synching occurs. Possible values are `overwrite`, which causes all existing events in the database to be deleted and replaced with uploaded data, and `mostrecent`, which lets existing events remain untouched if they were more recently modified than their uploaded counterpart.
- *value* is a string literal, which contains vCalendar information to update the database. This is in the same format as output of the `Downsync` command. Input terminates with a CRLF.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (can access only their own calendar)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, this command applies to the system's primary domain.

Example

```
8 Calendar Upsync u0 vcal overwrite {292+}
BEGIN:VCALENDAR
PRODID:-//Mirapoint Calendar
VERSION:1.0
BEGIN:VEVENT
UID:20010426T0000Z-0@when.mirapoint.com
DTSTART:20010426T070000Z
DTEND:20010426T080000Z
DCREATED:20010426T220700Z
LAST-MODIFIED:20010426T220700Z
PRIORITY:3
SUMMARY:test
CLASS:PRIVATE
END:VEVENT
END:VCALENDAR
8 OK Completed
```

10

The Cluster Command

The `Cluster` command configures the active and standby units for N+1 failover.

The older `Failover` command offers redundancy with one standby unit and one active unit, called 1+1 failover. The `Cluster` command now provides N+1 failover, where one standby unit provides redundancy for multiple active units (first to fail). SAN storage is required for N+1 failover. From the standpoint of performance, 4+1 or 5+1 are the recommended configurations, although 9+1 is the software limit.

Traditional failover, supported only by 3.x releases, requires a license. N+1 failover, also supported by 4.x releases, requires a different license.

Configuring an N+1 Failover Cluster

If already using Failover, unconfigure the existing active 1+1 failover head, revoke its license, reboot, and wait for the standby head to become active.

```
> Storage Configsan "" Detach
OK Completed
> License Revoke 8-license
OK Completed
> Reboot
```

Network the (detached) standby head and configure it for N+1 failover:

```
> Storage Listsan
1.0.1.1 SANTYPE 20000MB masked clear "" ...
OK Completed
> Storage Configsan 1.0.1.1 Init
OK Completed
> License Apply F-license
OK Completed
> Cluster Add Cluster1
OK Completed
```

Go to each active head, one by one, adding it to the N+1 failover cluster:

```
> License Apply F-license
OK Completed
> Cluster Attach Cluster1
OK Completed
```

To unconfigure an N+1 failover cluster, run `Cluster Detach` on each active head, then run `Cluster Delete` on the standby head.

It is not necessary to backup N+1 settings because the head unit must be reattached to the cluster's LUN before the restore can take place. Backup for N+1 SAN systems is handled by the SAN. For system recovery, first restore the SAN LUN using the

SAN vendor's backup software, then disaster recover the Mirapoint system using the System Recovery disk, and finally reattach the SAN LUN and SAN cluster.

Powering Down for Maintenance

When moving a cluster or performing maintenance that requires a shutdown, halt the standby first and then halt the active nodes, in any order. When powering up, boot the active nodes first, then the standby.

Subcommands

Add

Converts a SAN head unit into a N+1 standby unit. Reduces the unit's functionality so it can be used only for system or cluster monitoring and administrative mail.

Syntax

`tag Cluster Add clustername`

where *clustername* is an arbitrary 1-16 character unique name for a given cluster. Characters are limited to upper and lower case letters and the numbers 0 to 9. This command fails if the cluster name is already in use by a different cluster. It also fails if the head unit shows signs of being used to process email, such as the presence of user mailboxes and licenses unrelated to standby.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Cluster Add Cluster2
2 OK Completed
```

Attach

Associates an active SAN head unit with the failover cluster of a given cluster name. Should be run on active units, not on the standby unit.

Syntax

`tag Cluster Attach clustername`

where *clustername* is a name previously established by the `Cluster Add` command. Attach fails under the following circumstances: if the standby unit is not currently running, if the cluster already contains more than nine (9) active units, if a cluster of the given name does not exist, or has the same name as another cluster.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Cluster Attach Cluster2
4 OK Completed
```

Count

Displays the number of clusters visible on the network.

Syntax

tag `Cluster Count` *pattern*

where *pattern* may be either * or null string ("") to count everything.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Cluster Count ""
* 6 2
6 OK Completed
```

Countmember

Displays the number of of nodes in the cluster.

Syntax

tag `Cluster Countmember` *clustername pattern*

where *clustername* is a name previously established by the `Cluster Add` command, where *pattern* may be either * or null string ("") to count everything.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Cluster Countmember Cluster2 ""
* 8 3
8 OK Completed
```

Delete

After all active units are detached, removes the cluster and reverts the standby back to a normal active head. Can be run only on the standby unit.

Syntax

```
tag Cluster Delete
```

The deleted cluster name is the one previously established by `Cluster Add`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
12 Cluster Delete
12 OK Completed
```

Detach

Disassociates an active head unit from the failover service of a given cluster name. Should be run on active units, not on the standby unit.

Syntax

```
tag Cluster Detach
```

The detached cluster name is the one previously given to `Cluster Attach`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
11 Cluster Detach
11 OK Completed
```

Get

Retrieves the name, role, or status for an N+1 failover cluster.

Syntax

`tag Cluster Get parameter value`

where *parameter* is one of the following:

- `Name`—Prints the failover cluster name; *value* must be null string (“ “).
- `Role`—Prints the node’s role in the cluster (active or standby); *value* can specify a hostname, or can be null string (“ “) to indicate the current host.
- `Status`—Prints the cluster’s status (optimal or degraded); *value* can specify a cluster name, or can be null string (“ “) to indicate the current cluster. Optimal means the standby unit is available. Degraded means it has failed-over so the standby is gone.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Cluster Get Status Cluster2
* 3 optimal
3 OK Completed
```

List

Displays the cluster name of each N+1 failover group visible on the network.

Syntax

`tag Cluster List pattern start count`

where *pattern* may be either `*` or null string (`""`), *start* specifies the first cluster to list, and *count* specifies the extent of the listing.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
7 Cluster List "" "" ""
* 7 Cluster1
* 7 Cluster2
7 OK Completed
```

Listmember

Displays the hostname of each node in the N+1 failover cluster.

Syntax

`tag Cluster Listmember clustername pattern start count`

where *clustername* is a name previously established by the `Cluster Add` command; *pattern* may be either * or null string (""), *start* specifies the first hostname to list, and *count* specifies the extent of the listing.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
9 Cluster Listmember Cluster2 "" "" ""
* 9 ms1c2.example.com
* 9 ms2c2.example.com
* 9 ms3c2.example.com
9 OK Completed
```

Scan

Causes the standby to scan the network for new active units attached to its cluster name. Can be run only on the standby unit.

Syntax

`tag Cluster Scan`

After scanning is complete, you can display all active units in this cluster with the `Cluster Listmembers` command.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Cluster Scan
5 OK Completed
```

11

The Conf Command

The `Conf` command enables and disables certain system features; see `Conf Enable` for a list of configurable features.

Subcommands

Count

Responds with the number of features that can be enabled with `Conf Enable` and disabled with `Conf Disable`.

Syntax

```
tag Conf Count pattern
```

where *pattern* must be * or " " (null string).

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
2 Conf Count ""
* 2 12
2 OK Completed
```

Disable

Disables the specified feature.

Syntax

```
tag Conf Disable feature
```

where *feature* is one of those described under `Conf Enable`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Conf Disable Filtering
3 OK Completed
```

Enable

Enables a system feature. Configurable features are detailed below; however, the availability of certain features depends on whether you are licensed for its use.

Syntax

tag `Conf Enable` *feature*

where *feature* is one of:

- `Antispam`—Turns on the Antispam service, if licensed. For more information, see Help About Uce. (Chapter 66, The Uce Command)Default is enabled.
- `Antispam-all`—Turns on Antispam checking for delivery to non-local addresses, which includes outbound messages. Default is disabled. Otherwise same as `Antispam` above.
- `Antivirus-Fsav`——Turns on the Antivirus service using F-Secure software, if licensed. For details, see Help Antivirus. (Chapter 4, The Antivirus Command) Default is enabled when licensed.
- `Antivirus-Rapid`—Turns on Antivirus service using Rapid software, if licensed. For details, see (Chapter 4, The Antivirus Command) Default is enabled when licensed.
- `Antivirus-Sophos`—Turns on the Antivirus service using Sophos software, if licensed. For details, see (Chapter 4, The Antivirus Command) Default is enabled when licensed.
- `Consolelogin`—When enabled for security, the Mirapoint system console prompts for a login name and password before providing administrators with access to the command-line interface. Disabled by default.
- `Customercare`—Sends service Reporting; reports system health and status information to Mirapoint Customer Services. This helps Customer Support detect and respond quickly to problems with your system. Default is enabled.
- `DerivedomainIP`—Allows the system to derive the domain name of directly connected IMAP, POP, and WebMail users based on the incoming IP address. The `Netif Setdomain` command associates an IP address with a domain name.

- `Derivedomainurl`—Appends users' delegated domain name (typed at login) to identify them using fully qualified addresses. Subdomain levels are discarded until a matching domain is found, with fallback to the top-level domain. This is used by XML, WebMail, WebCal, and Administration Suite (old style and new Miradmin). Default is disabled.
- `EnterpriseUi`—Enables Corporate Edition WebMail and WebCal, which provides a uniform user interface combining mail, calendar, and address book. Default is disabled. Mandates a supported browser with Javascript. This option requires a license, as do WebMail and WebCal.
- `Filtering`—Enables per-user and domain message filtering (seeChapter 22, The Filter Command and the *Administrator's Guide*). Default is enabled.
- `Getmail`—Enables fetching of user email from remote systems (seeChapter 24, The Getmail Command and the online help for WebMail). Default is enabled.
- `Httpproxy`—Uses a proxy server for HTTP client operations; see `Conf Set`. Disabled by default.
- `Imaplistonlyinbox`—Configures the system so that IMAP clients only display users' own mailboxes instead of all the mailboxes available system-wide. Although this setting can significantly improve WebMail and IMAP performance, enabling this feature will prevent users from configuring additional shared folders. Users will still be able to access shared folders set up prior to the enabling of this setting.
- `Junkmailmanager`—Configure this system to act as a Junk Mail Manager, primarily by delivering spam messages to `QTNBOX` instead of `INBOX`. Usually used in conjunction with `Keeptomailhost`. Requires a JMM license.
- `Keeptomailhost`—If a user does not exist on this system, but has a remote mailhost, forward messages to that mailhost. How it works: when the `Filter` command's `Keep` action is performed, the user's mailhost is looked up in LDAP. If the user is non-local, messages are queued for that user and delivered without further MX processing. The Administrator mailbox is exempt from this action. Requires a Mail Routing license.
- `Ldapall`—Enables all LDAP configurations below. Disabled by default.
- `Ldapautoreply`—When enabled, automatic replies are controlled from LDAP instead of by the local system. This feature is intended for message servers only and should not be enabled on inbound message routers (IMRs). The LDAP attribute `miDeliveryOption` specifies autoreply. Attributes `miAutoreplyText`, `miAutoreplySubject`, and `miAutoreplyInterval` contain the body of the autoreply message, the subject, and the time interval between autoreplies to the same sender (as with `Autoreply Setinterval`). For more information, see (The Autoreply Command on page 73).
- `Ldapexception`—When enabled, allows administrators to set exceptions in an LDAP user entry. Denial of login is currently the only supported exception. Denied users get an "authentication failure" error, which is intentionally vague to thwart further security attacks. To deny login to a user, add an `miException` attribute to the user's LDAP entry, set the attribute value to `denylogin`, then

run the `Conf Enable Ldapexception` command. This presupposes use of LDAP, not local, authentication.

```
miException: denylogin
```

- `Ldapforward`—When enabled, forwards are set and retrieved from LDAP instead of by the local system. This feature is intended for message servers only and should not be enabled on inbound message routers (IMRs). The LDAP attributes `miDeliveryOption` and `miForwardingAddress` control delivery (forward with or without copying local mailbox) and destination address. For more information, see The Fwd Command on page 259.
- `Ldapgui`—If enabled, employs the LDAP directory server to maintain user and domain information. After version, the string "+LDAP" appears near the bottom of web pages in the Administration Suite, but otherwise the GUI looks similar, except for provisioning support. After account creation, users employ `Ldappassupdate` and so forth to change passwords and other information.
- `Ldapgui-jmm`—If enabled, extends `Ldapgui` (above) for Junk Mail Manager maintenance. This is the only LDAP configuration not enabled by `Ldapall`.
- `Ldappassupdate`—When enabled, permits password changes in the LDAP database. This is done by the `User Set Pass` command when `Plaintext:Ldap` authentication is active. Users modify their own passwords using LDAP Bind (80 byte limit). Administrators can modify user passwords using the access list defined by the `Ldap Addaccess` command. The specified password replaces the `UserPassword` attribute value in the user LDAP record. With `Plaintext:Ldap` authentication active, the **Change Password** link appears in WebMail when `Ldappassupdate` is enabled, but disappears when `Ldappassupdate` is disabled (link status determined by the `User Get Rights` command).
- `Ldapuuid`—If LDAP is enabled, the system writes user ID information into the `miUUID` field of user LDAP records. This behavior can be disabled.
- `Ldapwmprefs`—When enabled, stores all user WebMail preferences in LDAP, where they can be accessed by related applications. For more information about non-LDAP `User Set Wmprefs`, see The User Command on page 657.
- `Reputation`—Enables all functionality associated with IP reputations. See The Reputation Command on page 495.
- `Verifyreboot`—Enables reboot verification prompting. This also affects patch installation. Default enabled. If enabled, the CLI does extra prompting when reboots are necessary. If disabled, typing reboot actually causes reboot.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Conf Enable Filtering
```

```
4 OK Completed
```

Enabled

Responds with `YES` if the specified feature is enabled and `NO` otherwise.

Syntax

`*tag* Conf Enabled *feature*`

where *`feature`* is one of those described under `Conf Enable`.

Response

The response line has the format:

`* *tag* *yes-or-no*`

where *`yes-or-no`* is one of:

- NO—the feature is disabled
- YES—the feature is enabled

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

None

Example

```
5 Conf Enabled Filtering
* 5 YES
5 OK Completed
```

Export

Exports some system configuration data as a string literal, or to the given URL, which may designate either an HTTP or FTP server. System configuration is an opaque (undocumented) object, formatted so it can be pasted into `Conf Import`. A checksum is included for import verification. `Conf Export` and then `Import` create a partial mirror image of the original configuration.

`Conf Export` uses message-based backup for data export, transferring system configuration, but it stops short of copying user mailboxes. Items not exported include: Mirapoint licenses, the SMTP queue, mail logs, system logs, security logs, SSH keys, and network parameters normally set at installation time (IP address, netmask, gateway, DNS server, hostname, and domain name). Refer to the backup chapter in the *Administrator's Guide* for more information.

Syntax

```
tag Conf Export nothing confdata
```

where:

- `nothing` is reserved for future use and must currently be the null string (“ “).
- `confdata` may be a valid URL to specify an HTTP or FTP server to receive the system configuration data, or null string (“ “) to show system configuration as a string literal. The URL can have parameters specified appropriately to provide authentication information, if necessary.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Conf Export "" "ftp://login:passwd@sys.example.com:6080/archive/mira01.cnf"
* 8 INFO Connecting to server
8 OK Completed
```

Get

Gets configurable attributes, currently for the client HTTP proxy.

Syntax

```
tag Conf Get attribute
```

where `attribute` currently must be `Httpproxy`, the proxy server used for HTTP client operations.

Privilege Levels

Administrator

Domain Sensitivity

Returns an error in a delegated domain.

Example

```
8 Conf Get Httpproxy
* 8 myproxy.example.com:8000
8 OK Completed
```

Import

Fetches and applies system configuration either from a string literal or from URL, which may designate either an HTTP or FTP server. System configuration is an opaque object, formatted for pasting into the command-line interface. Tampering with configuration data may cause unpredictable behavior or system destruction. Applying multiple configurations has no effect; only the last configuration applied affects the system. `Conf Export` and then `Import` create a partial mirror image of the original configuration.

After decoding the opaque object, `Conf Import` uses message-based restore for data import. See Export on page 125 for a list of items not included in the imported system configuration.

Syntax

`tag` `Conf Import` *`nothing confdata`*

where:

- *`nothing`* is reserved for future use, perhaps limiting the extent of configuration, and must currently be the null string (" ").
- *`confdata`* may be a valid URL to specify an HTTP or FTP server to provide the system configuration data, or null string (" ") to import system configuration as a string literal. The URL can have parameters specified appropriately to provide authentication information, if necessary.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
9 Conf Import "" "ftp://login:passwd@sys.example.com:6080/archive/mira01.cnf"
* 9 INFO Fetching config
* 9 INFO Applying config
9 OK Completed
```

List

Responds with a list of features that can be enabled or disabled using `Conf Enable` and `Conf Disable`.

Syntax

`tag` `Conf List` *`pattern start count`*

- *pattern* is currently ignored.
- *start* is the number of the first feature keyword you want to see. The empty string ("") implicitly means 0.
- *count* is the number of feature keywords you want to see. The empty string ("") implicitly means all keywords. If *count* is greater than the total number of feature keywords, `list` returns as many keywords as possible.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
6 Conf List "" "" ""
* 6 antispam
* 6 antivirus-sophos
* 6 consolelogin
* 6 customercare
* 6 filtering
* 6 getmail
* 6 httpproxy
* 6 ldapautoreply
* 6 ldapexception
* 6 ldapforward
* 6 ldappassupdate
* 6 ldapwmprefs
* 6 verifyreboot
6 OK Completed
```

Set

Sets configurable attributes, currently for the client HTTP proxy.

Syntax

tag `Conf Set` *attribute value*

where:

- *attribute* currently must be `Httpproxy`, which is the proxy server used (by the Antivirus and Update subsystems for instance) with all HTTP client operations. If *value* is set to the null string ("") no HTTP proxying is done, which is the default. If *value* is set to a host name with optional modifiers, that hostname performs proxying. The form of *value* is as follows, where *user* and *password* optionally represent an authenticated login, *hostname* is mandatory, and *port* optionally gives the connection port:

  [*user*[:*passwd*]@]*hostname*[:*port*]

- *value* is the attribute setting.

Privilege Levels

Administrator

Domain Sensitivity

Returns an error in a delegated domain.

Example

```
7 Conf Set Httpproxy myproxy.example.com:8000
7 OK Completed
```

12

The Cos Command

The Cos command controls class of service (COS), which controls the services (such as IMAP or WebMail) that users may access, and also their mail quotas.

If COS checking is enabled for a service, an LDAP lookup is done before permitting users to access the service. If COS checking is not yet enabled, or has been disabled, all users on the system are granted access without restriction or LDAP lookup.

COS checking can be enabled for at least the following services: POP, IMAP, WebMail, calendaring, SSL login, message undelete, quotas, getmail, filtering, forwarding, autoreply, antispam, and antivirus.

User and class information are stored in the LDAP database. User records can have an attribute that points to a COS entry. If no attribute is present, COS uses the default entry for the user's domain, or @* if no domain entry is present.

```
dn: uid=@*,dc=example,dc=com
objectclass: mirapointMailUser
objectclass: inetOrgPerson
uid: @*
mail: @*
cn: Global COS default
miService: pop
```

Disabling a Service

Technically "DISABLED" is not a valid miService value, however it works **because** it is not a valid value.

If a user has no miService definition (even in domain or box wide scope) then COS defaults to giving that user all services. If a user has an miService definition, COS denies all services **except** the ones listed. So something like "miService: Suspended" in a user's LDAP attributes causes denial of all COS controlled features. However, if a service is not COS controlled, they can still use that service. For example, if COS checking for POP is disabled, users can still access POP service. To lock a user out of POP but still allow email to be delivered, even ignoring the state of COS enabled, run this command, and put "miException: denylogin" in the user's LDAP record.

```
Conf Enable Ldapexception
```

This allows email delivery but prevents the user from logging into a Mirapoint box that has Ldapexception enabled.

LDAP Extensions

COS uses the following LDAP attributes, which must be allowed by the schema:

- `miCosDN`—An attribute that holds the DN (distinguished name) for a COS entry. It can be placed in a user record to indicate that user's class of service.
- `miDefaultJunkmailFilter`—Specifies a new filter rule to replace the default System Junk Mail Rule. If the filter rule contains multiple lines, end it with two successive newlines (one blank line), base64-encode this text, and separate the encoded filter value from its attribute name by two colons and a space.
- `miMailExpirePolicy`—Attribute that controls message expiration values.
- `miMailQuota`—The `Ldap Setquery user:Quota` command gets the disk quota for a given class of service, using `miMailQuota` or `mailQuota` attribute, or on a Junk Mail Manager, `miQuarantineQuota`.
- `miMailUndeleteQuota`—Attribute that controls mailbox undelete behavior.
- `miQuarantineHost`—Specifies Junk Mail Manager host for a user or domain.
- `miService`—The given value adds a service for which the user is authorized. Possible values are the same as feature names under `Cos Enable`, except quota, which is controlled by an LDAP attribute such as miMailQuota and the `Ldap` command's `user:Quota` query filter.

Subcommands

Disable

Turns off access verification for the specified feature. All users are granted access.

Syntax

```
tag Cos Disable feature
```

where *feature* is a valid feature name (see `Cos Enable`).

Privilege Levels

Administrator

Domain Sensitivity

COS works in delegated domains, but this command runs only at the top level.

Example

```
3 Cos Disable Webmail
3 OK Completed
```

Enable

Turns on access verification for the specified feature. Only users that have LDAP configured to grant access to a particular feature are allowed to use that feature. Users who are denied access receive a “not authorized” error message.

The COS user lookup, like many others in a multi-tier environment, occurs after routing (or proxying). LDAP user queries must be defined so that lookups on the pre-routing and post-routing addresses return the same result.

To deny all COS services to a banned user, that user’s LDAP entry should contain a single value for the `miService` attribute: `Disabled` (as in the example below). Note: Disabled is not a `Cos Enable` keyword; this works due to lack of any valid feature.

```
miService: DISABLED
```

Syntax

```
tag Cos Enable feature
```

where *`feature`* is one of:

- `Antispam`—The user’s incoming messages are scanned to detect and filter out junk mail. COS entry should be based on the user’s destination address. LDAP attribute `miDefaultJunkmailFilter` specifies an alternate System Junk Mail rule. See System Junk Mail Rule on page 243 for details.
- `Antivirus`—The user’s incoming messages and attachments are scanned for viruses. COS entry should be based on the user’s destination address.
- `Autoreply`—The user can set up automatic replies.
- `Calendar`—User has access to schedule management and email notification.
- `EnterpriseUi`—User has access to Corporate Edition WebMail and WebCal.
- `Filter`—User has access to filtering capabilities. Filters are ignored if the user does not have access to this service.
- `Forward`—The user can set up automatic message forwarding.
- `Getmail`—The user can fetch POP email from remote systems.
- `Groupcalendar`—The user has access to group calendaring and notification.
- `Imap`—User has access to IMAP services.
- `Imodemail`—Deprecated in release 3.8.0.
- `Msgexpiration`—Enables deletion of messages older than a given date, based on the `miMailExpirePolicy` attribute, as in the LDAP entry shown below. The administrator must first schedule message expiration. An expired *`mailbox`* can be `Inbox`, `Inbox.`*`folder`*, or a shared mailbox. If a mailbox name contains a space, it must be quoted on the expire policy line (“Junk Mail” for example). Wildcards with `*` are allowed, and multiple lines in the LDAP database can apply to different subfolders. The *`days`* specify the number of 24-hour periods until expiration. The letter *`how`* can be either `I` (inserted) to delete messages that arrived that number of days ago, or `U` (updated) to delete messages that have

been read (`\Seen` flag changed) and arrived that number of days ago. The `U` flag does not apply to shared folders.

`miMailExpirePolicy:` *`mailbox days how`*

- `Msgundelete`—User is allowed to retrieve messages from the deletedmessages hierarchy into any mail folder. See also `Mailbox Undelete`. For configuration information, refer to "Setting Up Message Undelete for Users" in the *Administrator's Guide*. The LDAP attribute `miMailUndeleteQuota` specifies the undelete quota for users and shared mailboxes. This is a number indicating the size in bytes of the quota, similar to mail quota. The value -1 (unlimited) is not allowed, and quota cannot exceed disk size.

  `miMailUndeleteQuota:` *`bytes`*

- `Pop`—User has access to POP server.
- `Quota`—On overquota condition, system rechecks LDAP quota for possible change. See Quota Set on page 470. Mailbox quota can be set using LDAP attribute `miMailQuota` or `mailQuota` to specify the maximum size in bytes of a user's mailboxes. This value cannot exceed disk size.

  `miMailQuota:` *`bytes`*

- `Sender_as`—The user's outgoing messages and attachments are scanned to filter out junk mail. Antispam must also be enabled; see `antispam`.
- `Sender_av`—The user's outgoing messages and attachments are scanned to detect viruses. Antivirus must also be enabled; see `antivirus`.
- `Ssl`—User has permission to login via SSL.
- `Wapcalendar`—Deprecated in release 3.8.0.
- `Wapmail`—Deprecated in release 3.8.0.
- `Webmail`—User has access to WebMail Direct.

Privilege Levels

Administrator

Domain Sensitivity

COS works in delegated domains, but this command runs only at the top level.

Example

```
4 Cos Enable Filter
4 OK Completed
```

Enabled

Responds with whether a feature is enabled or disabled for access verification.

Syntax

tag `Cos Enabled` *feature*

where *feature* is a valid feature name (see `Cos Enable`).

Response

The response line has the format:

```
* tag yes-or-no
```

where *yes-or-no* is one of:

- NO—the feature is disabled
- YES—the feature is enabled

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

COS works in delegated domains, but this command runs only at the top level.

Example

```
5 Cos Enabled Filter
* 5 YES
5 OK Completed
```

Getstatus

Displays whether a user is allowed access to the service(s) or not, and also displays the LDAP configuration that determines the user's access rights.

If the retrieved LDAP entries total more than 1024 bytes, `Cos Getstatus` returns "System Error (Too many values)" to indicate overflow. In particular, creating too many `miMailExpirePolicy` attributes can cause overflow.

Syntax

```
tag Cos Getstatus username pattern start count
```

where:

- *username* is the ID of a user in LDAP (either login or published name).
- *pattern* is a valid feature name, or either `""` or `*` to indicate all features.
- *start* is the beginning feature number; the empty string (`""`) means `0`.
- *count* is the number of features to show; the empty string (`""`) displays all.

Response

This command returns two JMM-related values that cannot be COS-enabled, nor do they appear in `Cos List`: `defaultJunkMailFilter` and `junkMailManager`. Response lines have this format:

```
* tag feature yes-no-etc status
```

where:

- *`feature`* is a valid feature name (see `Cos Enable`).
- *`yes-no-etc`* is one of:
  - NO—the user is denied access.
  - YES—the user is allowed access.
  - `I` or `U`—The message expiration policy, based on Insert or Update-seen.
  - *`Quotasize`*—For Quota, the mail quota for this user (as obtained from LDAP). If quota is disabled, or an error occurred, then `unknown` is returned.
  - `Sender`—If Antivirus and Sender_av are licensed and enabled, and user *`username`* is not in LDAP, the keyword `sender` may appear.
- *`status`* is a descriptive string showing the feature's state as follows:
  - `Disabled`—COS checking not enabled for this feature.
  - `Direct`—User's entry in LDAP has COS values.
  - `Indirect`—User's entry in LDAP is COSDN indirect.
  - `Default-Direct`—Default entry for domain has COS values.
  - `Default-Indirect`—Default entry for domain is COSDN indirect.
  - `Equated-Direct`—User's entry in equated domain has COS values.
  - `Equated-Indirect`—User's equated domain entry is COSDN indirect.
  - `Equated-Default-Direct`—Default equated domain has COS values.
  - `Equated-Default-Indirect`—Default equated domain COSDN indirect.
  - `Global-Default-Direct`—Default global domain has COS values.
  - `Global-Default-Indirect`—Default global domain is COSDN indirect.
  - *`Error message`*—An error of this type occurred during lookup, preventing the COS check from succeeding.

Privilege Levels

- Administrator
- Helpdesk administrator

Domain Sensitivity

COS works in delegated domains, but this command runs only at the top level.

Example

```
2 Cos Getstatus juser@example.com webmail "" ""
* 2 webmail yes direct
2 OK Completed
```

List

Displays features that can be controlled by COS for user access.

Syntax

tag Cos List *pattern start count*

- *pattern* is either " " or * to indicate all features.
- *start* is the number of the first feature keyword you want to see. The empty string ("") implicitly means 0.
- *count* is the number of feature keywords you want to see. The empty string ("") implicitly means all keywords. If *count* is greater than the total number of feature keywords, List returns as many keywords as possible.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

COS works in delegated domains, but this command runs only at the top level.

Example

```
6 Cos List "" "" ""
* 6 antispam
* 6 antivirus
* 6 autoreply
* 6 calendar
* 6 filter
* 6 forward
* 6 getmail
* 6 groupcalendar
* 6 imap
* 6 msgexpiration
* 6 msgundelete
* 6 pop
* 6 quota
* 6 sender_as
* 6 sender_av
* 6 ssl
* 6 webmail
6 OK Completed
```

The Customercare Command

The `Customercare` command lets you specify contact information for your system to allow Mirapoint Customer Services to contact you if Service Reporting reports any problem with your system. For more information, see the `Conf` command.

Subcommands

Get

Responds with the value of the specified parameter.

Syntax

`tag` `Customercare Get` *`parameter`*

where *`parameter`* must currently be `Contact`, meaning the information that Customer Services can use to contact you if there is a problem with your system.

Privilege Levels

- ◆ Administrator
- ◆ Backup operator

Domain Sensitivity

None

Example

```
3 Customercare Get Contact
* 3 {64}
Jane Administrator
999-999-9999
Anytown, USA
jadmin@example.com

3 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

`tag Customercare Set parameter value`

where:

- *`parameter`* must currently be `Contact`, meaning the information that Mirapoint Customer Services can use to contact you if Service Reporting reports any problem with your system.
- *`value`* is the string (less than 1024 bytes long) that you assign to *`parameter`*.

  The value is a literal string of the following suggested format:

  ```
  Name: contact-name
  Phone: number
  Address: mailing-address
  E-mail: email-address
  ```

  where:

  - *`contact-name`* is the administrator's name.
  - *`number`* is the telephone number for the system administrator.
  - *`mailing-address`* is the administrator's mailing address, on one line.
  - *`email-address`* is the administrator's email address.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Customercare Set Contact
* 2 {64}
Jane Administrator
999-999-9999
Anytown, USA
jadmin@example.com
2 OK Completed
```

The Diag Command

The `Diag` command runs diagnostic programs on the Mirapoint system.

Subcommands

Abort

Stops the currently running diagnostic program immediately.

Syntax

```
tag Diag Abort
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Diag Abort
2 OK Completed
```

Changer

Returns the results of a SCSI `Inquiry` command on an attached autochanger device.

Syntax

```
tag Diag Changer Inquiry "" "" ""
```

Three pairs of double quotes are needed.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
9 Diag Changer Inquiry "" "" ""
* 9 "EXABYTE " "Exabyte 17D      " "E113" "THX1138"
9 OK Completed
```

Clear

Clears all accumulated tape values, thus resetting the compression ratio.

Syntax

tag Diag Clear *parameter*

where *parameter* currently must be TapeDeviceLog.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
10 Diag Clear TapeDeviceLog
10 OK Completed
```

Disk

Displays information about a specific disk device.

Syntax

tag Diag Disk Inquiry *logicalID*

where *logicalID* is a quadruple (A.B.C.D) describing the Lun.

Parameter and value pairs may include: vendor string, model name, revision level, serial number, number of raw sectors, sector size in bytes, access type device, unique ID of the system that created this Lun, and host name that last mounted this Lun.

In the output, a SAN storage device that had a valid mount in the past will display its last modify time, and drive serial number data will show asterisks (*) in place of unprintable characters.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
17 Diag Disk Inquiry 1.6.0.0
* 17 vendor HP
* 17 model OPEN-8*3
* 17 revision 1550
* 17 serial_num 092342034LKDJFLJSD
* 17 sectors 53687091200
* 17 sector_size 512
* 17 access_type "Fixed Direct Access SCSI-2"
* 17 config_uid 00a0c9adfe6b3c4e7131
* 17 hostname mail.example.com
17 OK Completed
```

Get

Retrieves the value of a given diagnostic or tape parameter.

Syntax

tag `Diag Get` *parameter*

where *parameter* is one of:

- `ChangerAddress`—The autochanger device's controller, target, and logical unit number.
- `Dataxferelementaddr`—Returns the physical element addresses of all data transfer elements (tape drives) contained in the changer attached to a Mirapoint system. If the changer is not attached to the Mirapoint system, you must obtain element addresses in some other manner. However, data transfer elements need not be attached to anything valid in order for these addresses to be useful.
- `TapeAddress`—The tape device's controller, target, logical unit number, and actual device name.
- `TapeCompression`—Returns the current state of the drive's compression.
- `TapeCompressionRatio`—Returns the read and write compression ratios. Depending on the amount of data read or written so far, this compression ratio may be invalid. How much data must be transferred before the ratio becomes valid is device-dependent; no attempt is made to filter out invalid ratios.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
11 Diag Get TapeCompression
* 11 ON
11 OK Completed

12 Diag Get TapeCompressionRatio
* 12 Read 2:1 Write 1.87:1
12 OK Completed

13 Diag Get TapeAddress
* 13 Controller 1, Target 2, LUN 0, Device /dev/nrsa0
13 OK Completed

14 Diag Get ChangerAddress
* 14 Controller 1 Target 1 LUN 0
14 OK Completed

15 Diag Get Dataxferelementaddr
* 15 0 480 "HP Ultrium 2-SCSI HUL3H04210"
* 15 1 481 "HP Ultrium 2-SCSI HUL3J02140"
15 OK Completed
```

Netstat

Shows network status, including network addresses and the state of sockets. Output is similar to the Unix command `netstat -anf inet`. A similar command exists on WindowsNT. The `Netstat` subcommand displays the current state of all sockets, with numeric network addresses, for the AF_INET protocol family.

Syntax

tag `Diag Netstat` *pattern*

where *pattern* matches data lines and acts as a filter to limit the amount of output. For example, "`*.25*`" shows all SMTP-related network information on port 25, while "`*10.0.0.128*`" shows information related to IP address 10.0.0.128 only.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Diag Netstat *
* 3 tcp 0 0 10.0.3.29.10143 192.168.0.36.42634 ESTABLISHED
* 3 tcp 0 0 10.0.3.29.10144 192.168.0.84.3376 ESTABLISHED
* 3 tcp 0 0 *.7937 *.* LISTEN
* 3 tcp 0 0 *.7938 *.* LISTEN
* 3 tcp 0 0 *.111 *.* LISTEN
* 3 tcp 0 0 *.25 *.* LISTEN
* 3 tcp 0 0 *.513 *.* LISTEN
* 3 tcp 0 0 *.514 *.* LISTEN
* 3 tcp 0 0 *.21 *.* LISTEN
* 3 tcp 0 0 127.0.0.1.53 *.* LISTEN
* 3 tcp 0 0 *.80 *.* LISTEN
* 3 tcp 0 0 *.2424 *.* LISTEN
* 3 tcp 0 0 *.10144 *.* LISTEN
* 3 tcp 0 0 *.10143 *.* LISTEN
* 3 tcp 0 0 *.113 *.* LISTEN
* 3 tcp 0 0 *.23 *.* LISTEN
* 3 udp 0 0 *.7938 *.* UDP
* 3 udp 0 0 *.111 *.* UDP
* 3 udp 0 0 *.* *.* UDP
* 3 udp 0 0 127.0.0.1.123 *.* UDP
* 3 udp 0 0 10.0.3.29.123 *.* UDP
* 3 udp 0 0 *.123 *.* UDP
* 3 udp 0 0 *.53 *.* UDP
* 3 udp 0 0 127.0.0.1.53 *.* UDP
* 3 udp 0 0 10.0.3.29.53 *.* UDP
* 3 udp 0 0 *.* *.* UDP
3 OK Completed
```

Ping

Sends the specified number of network packets to the specified host and reports the results of transmission. This command lets you determine whether specific hosts are reachable on the TCP/IP network.

Syntax

tag Diag Ping *host packetcount*

where:

- *host* is the host name or IP address of the host you want to ping.
- *packetcount* is the number of ECHO_RESPONSE packets to send to *host*. This must be a number in the range from 1 to 60. A value of "" (empty string) implicitly means 5, the default number of packets.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Diag Ping doc1 1
* 4 "PING 10.0.3.29: 56 data bytes"
* 4 "Received 64 bytes from 10.0.3.29: seq=0 ttl=255 time=0.15 ms"
* 4 "Statistics for 10.0.3.29:"
* 4 "Packets:  1 transmitted,  1 received,  0 (0%) lost"
* 4 "Round trip time: min=0.15 ms,  max= 0.15 ms,  ave= 0.15 ms"
4 OK Completed
```

Rescan

Requests rescanning of SCSI busses that might have a tape or autochanger device attached, in order to detect addition or removal of those devices.

This command is deprecated. You must reboot the system to add a new SCSI device. Please refer to the hardware manual for your system for detailed instructions.

It might take several minutes to run `Rescan`. Failure to find any added or removed devices is not an error. An incorrectly added device may cause the `Rescan` command to hang. Some reported errors may be fixed only by a reboot, depending on the error. It is unknown what all the errors may be.

Syntax

tag `Diag Rescan`

SCSI bus rescans are determined by the system and may not be selected.

Privilege Levels

Administrator

Domain Sensitivity

Only available from the primary domain.

Example

```
16 Diag Rescan
* 16 rescan of bus 3 succeeded
* 16 rescan of bus 4 succeeded
* 16 rescan of bus 5 succeeded
16 OK Completed
```

Set

Changes the value of a given diagnostic or tape parameter. The tape device must be online for a tape `Set` command to succeed (it could still fail for other reasons).

Syntax

tag `Diag Set` *parameter value*

where:

- *parameter* currently must be `TapeCompression`.
- *value* is either `ON` or `OFF` to indicate whether tape hardware compression is enabled or not. The default value of tape compression is `ON`. The manufacturers of all tape devices currently supported by Mirapoint recommend leaving tape compression `ON`. Turning tape compression off negatively impacts backup and restore performance.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
15 Diag Set TapeCompression OFF
15 OK Completed
```

Smtpconnect

Performs a diagnostic check of the connection between the SMTP service on the current system and SMTP service on some remote system. `Diag Smtpconnect` sends a predefined message where you specify the From address, the To address, and the remote SMTP server's hostname or IP address, with optional port.

Syntax

`tag Diag Smtpconnect fromAddress toAddress remoteSmtp[:port] options`

where:

- *fromAddress* is the envelope From address appearing in the test message.
- *toAddress* is the envelope To address of the intended message recipient.
- *remoteSmtp* is the fully qualified DNS hostname (or numeric IP address) of the remote SMTP server.
- *port* is the optional port number for SMTP service, which defaults to 25 if not present, and may be any value from 1 to 65535.
- *options* can be either null string (" ") or the keyword `timeout` in this format:
  - `timeout=`*numtries*[,*interval*]
    where *numtries* is the number of connection attempts `Diag Smtpconnect` will make, and *interval* [optional] is the seconds of wait time between attempts. The default value of both *numtries* and *interval* is 1 (one).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
18 Diag Smtpconnect joe@testing.net juser@example.com mail.example.com ""
18 "-Date: Wed, 18 Oct 2006 19:55:23 +0000"
```

```
18 "-trying mail.example.com (192.168.0.19)"
18 "-connected to mail.example.com"
18 "<220 mail.example.com ESMTP Mirapoint Messaging Server MOS nnn Queueing"
18 ">EHLO qa196.mirapoint.com"
18 "<250-alpo.mirapoint.com Hello qa196.mirapoint.com"
18 "<250-8BITMIME"
18 "<250-SIZE 50000000"
18 "<250-ETRN"
18 "<250-DSN"
18 "<250-STARTTLS"
18 "<250 PIPELINING"
18 ">MAIL FROM:"
18 "<250 Sender OK"
18 ">RCPT TO:"
18 "<250 Recipient OK"
18 ">DATA"
18 "<354 Enter your message, followed by a dot on a line by itself"
18 ">Date: Wed, 18 Oct 2006 19:55:23 +0000"
18 ">Subject: MOS diag smtpconnect from qa.testing.net"
18 ">To: joe@testing.net"
18 ">From: juser@example.com"
18 ">
18 MOS diag smtpconnect test message from:
18 qa.testing.net
18 Sent at Wed, 18 Oct 2006 19:55:23 +0000"
18 ">."
18 "<250 ACB36404 Message accepted for delivery"
18 ">QUIT"
18 OK Complete
```

Status

Responds with the output of the most recent diagnostic command. Currently this covers only tape-related operations.

Syntax

tag Diag Status

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Diag Status
* 5 ReadWriteTest: test begin
* 5 ReadWriteTest: test ok
* 5 REPORT SUMMARY:
* 5 Read Write Test: passed
5 OK Completed
```

Tape

Runs the specified diagnostic program on the system's attached tape drive.

Syntax

tag `Diag Tape` *program numrecords times blocksize*

where:

- *program* is one of:
  - `Inquiry`—Responds with the results of a SCSI `Inquiry` command on the tape drive.
  - `Read`—Reads the specified number of records from the tape drive. You must first use `Write` to write the same number of records to the tape. To view the output of this diagnostic, use `Diag Status`.
  - `Write`—Writes the specified number of records to the tape drive in preparation for using `Read`. To view the output of this diagnostic, use `Diag Status`.
  - `Readwrite`—Equivalent to performing `Read` followed by `Write` for the specified number of records. To view the output of this diagnostic, use `Diag Status`.
  - `All`—Runs all the above diagnostics.
- *numrecords* is the number of records to read or write. For `Inquiry`, you must specify the empty string (`""`) for this parameter because it is required but ignored. For all other diagnostics, if you specify the empty string, the program uses the default number of 64 records.
- *times* is the number times to perform the specified `Inquiry`, `Read`, `Write`, or `Readwrite` diagnostic. If you specify the empty string (`""`), the specified diagnostic runs once.
- *blocksize* is the block size in bytes to use in reading or writing the tape. If you specify the empty string (`""`), the program uses the default block size of 65536. The maximum value is 131072 for release 3.0 and later. If the number ends in a `b`, `k`, `m`, or `g` it is multiplied by 1 (byte), 1024 (kilobyte), 1048576 (megabyte) or 1073741824 (gigabyte). If the number is out-of-range, it reverts to *blocksize*.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Diag Tape Inquiry "" "" ""
* 6 "HP      " "Ultrium 1-SCSI  " "E12V" "THX1138"
6 OK Completed
7 Diag Tape All "" "" ""
```

```
7 OK Completed
```

Tcptraceroute

Sends TCP packets to the specified host and reports the route used to reach the target host. This command lets you determine how network packets are routed on the IP network from your Mirapoint system to specific hosts.

With widespread use of firewalls on the Internet, packets sent by `Diag Traceroute` often get filtered, making it impossible to trace the network path to a destination. Many firewalls permit inbound TCP packets on specific ports, so by sending TCP SYN packets (instead of UDP or ICMP ECHO packets) `Diag Tcptraceroute` is able to bypass the most common firewall filters.

Syntax

tag `Diag Tcpraceroute` *host portnum*

where *host* is the host name or IP address of the host to which you want to trace the route, and *portnum* is the port number.

Response

An "!" after timings indicates TTL (time to live) is <= 1, with these error codes:

- `!A`—Access denied (by network firewall or filter).
- `!C`—Host prohibited access.
- `!F`—Fragment needed, cannot finish assembling packet.
- `!H`—Host is unreachable.
- `!N`—Network is unreachable.
- `!P`—Protocol is unreachable.
- `!p`—Port is unreachable.
- `!S`—Source route failed, unreachable.
- `!U`—Unknown host or network.
- `!T`—TOS (type of service) unreachable.
- `!`*n* —Numeric code for other ICMP error.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
9 Diag Tcptraceroute mail.example.com
```

```
* 9 "Tracing path to mail.example.com (10.1.1.9) on TCP port 80, 30 hops max"
* 9 "1 10.1.1.1 171.576 ms 189.566 ms 219.128 ms"
* 9 “2 10.1.1.9 319.904 ms 263.843 ms 142.382 ms”
9 OK Completed
```

Traceroute

Sends network packets to the specified host and reports the route used to reach the target host. This command lets you determine how network packets are routed on the TCP/IP network from your Mirapoint system to specific hosts.

This command uses 40-byte packets and does not trace routes requiring more than 30 hops.

Syntax

```
tag Diag Traceroute host
```

where *host* is the host name or IP address of the host to which you want to trace the route.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Diag Traceroute doc1
* 8 "traceroute to 10.0.3.29,  30 hops max,  40 byte packets"
* 8 " 1  10.0.3.29  0.12 ms  0.06 ms  0.06 ms"
8 OK Completed
```

15

The Dir Command

`Dir` commands provide an way to manage LDAP databases on a Directory Server.

You maintain LDAP data on a different system, transferring it to the Mirapoint Directory Server using the `ImportLdif` command. As delivered, the Directory Server includes a database named `default`, but you can create others.

Directory Server requires a license, which may have been preinstalled at the factory. Licensing automatically enables and starts the service. Here are the basic commands needed to allow the Directory Server to service LDAP requests:

```
Service Enable Dir  # maybe
Service Start Dir   # maybe
Dir AddDb db1 ""
Dir AddDbsuffix db1 dc=example,dc=com
Dir ImportLdif dc=example,dc=com "" ftp://mysys/db1.ldif
```

In the final line of this example, `db1.ldif` is an LDIF (LDAP Data Interchange Format) file maintained on an FTP server called `mysys`. This LDIF file might be automatically produced by scripts from employee or customer databases elsewhere within your organization.

One good way to back up a Directory Server is to replicate data on another server. Another way is to `ExportLdif` from one server and `ImportLdif` to another. To save an entire LDAP directory server onto tape, Mirapoint recommends image backup with NDMP. For details, refer to various Knowledge Base articles about NDMP on the http://support.mirapoint.com website.

Directory Command Overview

The `ImportConfig` and `ExportConfig` subcommands allow semi-automatic configuration of a directory server from or to another server. See page 154.

The `Set` and `Get` subcommands control directory server global options; `Setlogging` and `Getlogging` control protocol logging. See page 156.

The `AddDb`, `CountDb`, `DeleteDb`, and `ListDb` subcommands control the layout of directory server data. A variety of options, and database suffixes to shorten object names, can be associated with each database. See page 159.

The `ImportLdif` and `ExportLdif` subcommands allow import and export of data to and from the directory server. The `ModifyLdif` subcommand allows selective modification of LDAP entries. See page 167.

The AddSchema, CountSchema, DeleteSchema, GetSchema, ListSchema, and SetSchema subcommands enable modification of your LDAP schema. The AddIndex, CountIndex, DeleteIndex, GetIndex, ListIndex, and SetIndex subcommands permit indexing of selected LDAP fields. See page 171. You can see the default schema by typing Dir ListSchema then Dir GetSchema *Name*.

The Replicate subcommand performs replication of the LDAP database server. The AddRepHost, CountRepHost, DeleteRepHost, and ListRepHost subcommands manage a slave system in a replication agreement. The AddReplica, CountReplica, DeleteReplica, and ListReplica subcommands manage replication agreements between a master and one or more slaves. The CountRepOption, GetRepOption, ListRepOption, and SetRepOption subcommands manage options to those replication agreements. See page 180.

The AddACL, CountACL, DeleteACL, GetACL, ListACL, and SetACL subcommands create access control lists that specify access to an item (such as everything, self, domains, DNs, DN attributes, and so forth). The AddACLentry, CountACLentry, DeleteACLentry, GetACLentry, ListACLentry, and SetACLentry subcommands add entries to an existing access control list and specify access types (such as none, compare, search, read, or write). See page 190.

Database names, schema names, replica names, and ACL names must start with an alphabetic character or underscore, followed by alphabetics, digits, or underscores, up to a maximum length of 32 characters.

For information about configuring LDAP, refer to the *Administrator's Guide*.

Configuration Subcommands

These subcommands allow semi-automatic configuration of a directory server from information stored on another system. Database configuration includes information about indexes, database options, suffixes, custom schemas, ACLs, and ACL entries. The full configuration may be exported from one system and imported to another, although information might need editing before importation.

ExportConfig

The ExportConfig command produces a copy of the directory server configuration in a format that can later be imported into the same or a similar system with the ImportConfig subcommand. The data format is release-specific, allowing simple initialization of a network of systems.

Syntax

```
tag Dir ExportConfig arg
```

where *arg* is currently ignored and must be "" (the empty string).

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
2 Dir ExportConfig ""
* 2 {547}
indexes = (4:maillocaladdress, 4:mail, 4:cn, 4:sn, 4:uid, 4:objectclass);
7:read:all = (*, read, "");
4:maillocaladdress = (eq);
suffixes = (0default:);
acls = (3:passwd, 3:read);
aclentries = (7:passwd:self, 7:passwd:others, 7:read:all);
7:passwd:self = (self, write, "");
9default = ("");
3:passwd = ("(attr=userpassword)");
4:cn = ("eq,pres");
7:passwd:others = (*, compare, "");
4:uid = ("eq,pres");
3:read = (*);
4:mail = ("eq,pres");
4:objectclass = ("eq,pres");
4:sn = ("eq,pres");
databases = (9default);
0default: = ();

2 OK Completed
```

ImportConfig

The `ImportConfig` command downloads all information about a system that was generated by an `ExportConfig` command. This is expected to be done only upon initial configuration of a system.

Syntax

`tag Dir ImportConfig arg value`

where:

- *arg* is currently ignored and must be " " (the empty string).
- *value* represents is either a literal string as output by `ExportConfig` or a URL representing a file on an FTP server.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Dir ImportConfig "" {547+}
...
```

```
* 3 INFO "verifying data"
3 OK Completed
```

```
4 Dir ImportConfig "" ftp://server/config1
* 4 INFO "Fetching data: ftp://server/config1"
* 4 INFO "verifying data"
4 OK Completed
```

Option Subcommands

These subcommands control directory server options including security and user name translation.

Get

This is the inverse of the `Set` command. The `Get` command returns a default value if the option is unset. For details see the `Dir Set` command.

Syntax

tag `Dir Get` *option*

where *option* may be `password-hash` (encryption scheme), `security` (cleartext and SSL), or `userIDtoDN` (login to distinguished name mapping).

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
5 Dir Get Security
* 5 "cleartext ssl"
5 OK Completed
```

```
6 Dir Get UserIDtoDN
* 6 "uid=$(mbox),dc=example,dc=com"
6 OK Completed
```

GetLogging

The `GetLogging` command returns the status of directory server protocol logging.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Syntax

tag Dir GetLogging *logtype*

where *logtype* is one of the types specified by Dir SetLogging, or * (or " ") to get all directory logging types.

Example

```
5 Dir GetLogging ""
* 5 protocol "on"
5 OK Completed
```

Set

The Set command controls global options. These options are always preset to a default value that you can change, or reset with the null string (" ").

Syntax

tag Dir Set *option value*

where *option* and *value* may be one of the following:

- Password-hash *scheme*

  Sets the default used by the set-password extended operation, and checks that changes to attributes using userPassword syntax 1.3.6.1.4.1.1466.115.121.1.40 data contains a scheme. Supported schemes are: CLEARTEXT, CRYPT (same as UNIX), MD5, MIRA, NS-MTA-MD5, SMD5, SHA, SSHA. The default is unset, and issuing the command with an empty string resets the default. When unset, the SSHA hash mechanism is used for the set-password extended operation, and userPassword updates are not required to be hashed. MIRA is Mirapoint-specific and may be used with APOP:LDAP authentication.

- Security *security-list*

  The Security option sets the order of access methods for the directory service. Choices are cleartext (the default) and ssl (secure socket layer).

- UserIDtoDN *template*

  The userIDtoDN option specifies a mapping of SASL (Simple Authentication and Security Layer) ID name to a distinguished name. The default template uid=$(mbox),$(dcmap) is used during SASL binding when the administrator wishes to use Kerberos GSSAPI authentication. Possible templates are:

  - $(mbox)—name of the user
  - $(domain)—the user's full DNS domain, such as example.com
  - $(login)—the user's full login name, including the DNS domain, equivalent to $(mbox)@$(domain)
  - $(dcmap)—a comma-separated list of DNS domain components, such as dc=example,dc=com

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Dir Set Security "cleartext ssl"
6 OK Completed
```

```
7 Dir Set UserIDtoDN "uid=$(mbox),dc=example,dc=com"
7 OK Completed
```

SetLogging

The `SetLogging` command changes the status of directory server protocol logging.

Syntax

tag `Dir SetLogging` *logtype level*

where *logtype* is one of the following:

- `authentication`—If `On`, log both successful logins to the server and failed authentication attempts. Default is `failures` to log failed authentication only. If *level* is `Off`, neither successful nor failed attempts get logged.
- `index`—If *level* is `Unindexed`, log search attributes for which no index is present. If `Errors`, log attributes for which no matching rules are defined. The default is `On` to log both. If `Off`, all index logging is disabled.
- `protocol`—If `On`, log LDAP protocol requests and responses from the server. If *level* is `Brief`, attributes and values are not logged. Default is `Off`.
- `replication`—If `On`, log replication events (see `Dir Replicate`). This logging impacts performance, but logs are useful when debugging replication problems. If *level* is `Off`, replication events are not logged. Default is `Off`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Dir SetLogging protocol on
6 OK Completed
```

Database Subcommands

These subcommands control database layout on the server. A database name is a unique identifier that constitutes a grouping of directory entries. A database has single valued options that define behavior for access to data in the database, an access list, some number of indexed attributes, and a list of base DNs that represent the top of directory information trees stored in the database.

AddDb

The `AddDb` command creates a new directory database instance. This database represents a grouping of related information. A database called "`default`" is preinstalled on new systems at the factory.

One or more directory information trees may be associated with a database using the `AddDbSuffix` command below. Options may be set on the database to control behavior using the `SetDbOption` command. One or more attributes within the database may be indexed by using the `AddIndex` command. Access control lists may be added to a database with the `AddAcl` command. Data may be imported using the `ImportLdif` command. When a database is deleted, all suffixes, options, indexes, ACLs, and data associated with the database are also removed.

Syntax

`tag` `Dir AddDb` *`dbname args`*

where:

- *`dbname`* is the directory database instance name.
- *`args`* is reserved for future use and must currently be the null string (" ").

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Dir AddDb database1 ""
8 OK Completed
```

CountDb

The `CountDb` command returns the number of databases that match a given pattern.

Syntax

`tag` `Dir CountDb` *`pattern`*

where *pattern* is a string, possibly containing wildcards, to match database names, or the null string to match all databases.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
9 Dir CountDb "database*"
* 9 2
9 OK Completed
```

DeleteDb

The `DeleteDb` command removes the given database, along with any suffixes, options, indexes, ACLs, and data associated with it.

Syntax

```
tag Dir DeleteDb dbname
```

where *dbname* is the directory database instance name.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
10 Dir DeleteDb "database2"
10 OK Completed
```

ListDb

The `ListDb` command lists any databases supported by the server that matches a given pattern in the range specified by start and count.

Syntax

```
tag Dir ListDb pattern start count
```

where `pattern` is a string, possibly containing wildcards, to match database names, or null string to match all databases; `start` and `count` enumerate begin and end.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
11 Dir ListDb "database*" "" ""
* 11 database1 ""
* 11 database2 ""
11 OK Completed
```

AddDbSuffix

The `AddDbSuffix` command adds a new directory information tree to the named database. You specify the DN of the base node in this tree. The database must already have been created using the `AddDb` command. Data may be added using the `ImportLdif` or `ModifyLdif` commands, or through the LDAP protocol.

You may not create suffixes that already exist in another database. You also may not add database suffixes if they are children of existing suffixes in that database.

Syntax

```
tag Dir AddDbSuffix dbname DN
```

where:

- *dbname* is the directory database instance name.
- *DN* is the distinguished name for the root of this directory information tree.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
12 Dir AddDbSuffix database1 dc=example,dc=com
12 OK Completed
```

```
12e Dir AddDbSuffix database1 dc=child,dc=example,dc=com
12e NO child to existing suffix in database
```

```
12f Dir AddDbSuffix anotherDB dc=child,dc=example,dc=com
12f OK Completed
```

CountDbSuffix

The `CountDbSuffix` command returns the number of items in the suffix list that match the given pattern.

Syntax

`tag Dir CountDbSuffix dbname pattern`

where:

- *dbname* is the directory database instance name.
- *pattern* is a string, possibly containing wildcards, to match the DN of this directory information tree.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
13 Dir CountDbSuffix database1 ""
* 13 1
13 OK Completed
```

DeleteDbSuffix

The `DeleteDbSuffix` command removes support for a given directory information tree (DIT), and removes data associated with that DIT from the database.

Syntax

`tag Dir DeleteDbSuffix dbname DN`

where:

- *dbname* is the directory database instance name.
- *DN* is the distinguished name for the root of this directory information tree.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
14 Dir DeleteDbSuffix database1 dc=example,dc=com
14 OK Completed
```

ListDbSuffix

The `ListDbSuffix` command shows a number of items within the suffix list that match the given pattern in the range specified by start and count.

Syntax

```
tag Dir ListDbSuffix dbname pattern start count
```

where:

- *dbname* is the directory database instance name.
- *pattern* is a string, possibly containing wildcards, to match the DN of database suffixes, or the null string to match all suffixes for this database; *start* and *count* give the beginning of the matching list and the number of elements.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
15 Dir ListDbSuffix database1 "" "" ""
* 15 database1 dc=mirapoint,dc=com
15 OK Completed
```

CountDbOption

The `CountDbOption` command returns the number of items in the database option list that match the given pattern.

Syntax

```
tag Dir CountDbOption dbname pattern
```

where:

- *dbname* is the directory database instance name. An empty *dbname* (“ “) counts global options.
- *pattern* is a string, possibly containing wildcards, to match database options.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
16 Dir CountDbOption database1 "*timelimit"
* 16 1
16 OK Completed
```

GetDbOption

The `GetDbOption` command prints one requested entry from the option list. This command is the same as `ListDbOption` with a single name.

Syntax

```
tag Dir GetDbOption dbname option
```

where:

- *dbname* is the directory database instance name. An empty *dbname* (“ “) gets the global value.
- *option* is one of the keywords listed under the `Dir SetDbOption` command.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
17 Dir GetDbOption database1 timelimit
* 17 360
17 OK Completed
```

ListDbOption

The `ListDbOption` command returns a number of options that match the given pattern in the range specified by start and count.

Syntax

tag `Dir ListDbOption` *dbname pattern start count*

where:

- *dbname* is the directory database instance name. An empty *dbname* (“ ”) lists global option values.
- *pattern* is a string, possibly containing wildcards, to match database option, or null string to match all options; *start* and *count* enumerate begin and end.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
18 Dir ListDbOption database1 "" "" ""
* 18 idletimeout 360
* 18 readonly true
* 18 referral ldap://ldap.mirapoint.com:389/
* 18 refonly off
* 18 rootdn uid=root,dc=mirapoint,dc=com
* 18 rootpw {CRYPT}k1llk44cooOf00xf
* 18 schemacheck TRUE
* 18 sizelimit 600
* 18 timelimit 60
18 OK Completed
```

SetDbOption

The `SetDbOption` command sets a single-valued option for the given database. These options may be specific to a database, or global across all databases by using an empty string as the database argument. If unset the option takes a default value, as given below. Options may be reset to default by setting values to “ “ (null string).

Syntax

tag `Dir SetDbOption` *dbname option value*

where:

- *dbname* is the directory database instance name. An empty *dbname* (" ") sets the option globally across all databases to the default value. If the option value is set for a specific database, then that value overrides the global default value.
- *option* is one of the following, and *value* is the option setting:
  - `Chaining on|off`
    Enables or disables LDAP chaining for each database. The default is off.
  - `IdleTimeout` *seconds*
    Maximum amount of time that a client may remain connected to the server without sending a request before the connection is terminated. Default 60.
  - `Proxyauth on|off`
    Enables or disables proxy authentication for each database. Default is off.
  - `RdnIntegrity` *setting*
    RFC 2251 states that attributes present in the relative domain name (RDN) of an entry may not be modified by the LDAP add and modify operations. Vendors implement this restriction differently; this option provides settings for compatibility with different databases (the default is `Off`):
    - `On`—Require any naming attributes to be present at all times. Add without RDN attributes yields "constraint violation" error.
    - `Off`—Do not enforce any restrictions on naming attributes.
    - `Nomodify`—Do not allow modification of any naming attributes, but do not require their presence. Modify yields "not allowed" error.
    - `Implicit`—Infer additional attributes for the entry from its RDN.
  - `ReadOnly on|off`
    Specifies whether the server allows updates (on) or does not allow updates (off). The default is off.
  - `Referral` *URL*
    An LDAP URL to return when running with the `RefOnly` flag set, or when an item is requested from a directory information tree that is unsupported by this server. If the `RefOnly` flag is set, but no referral is defined, then an error is returned to the client. The default is unset.
  - `RefOnly on|off`
    Specifies whether server returns referrals only (on, required for chaining) or serves data requests from its local database (off). The default is off.
  - `RootDN` *DN*
    A distinguished name for the LDAP superuser. Any client able to bind as this user is able to bypass all access rights to change the database at will. The DN must be within the directory information tree in the given `dbname`, and a global default `rootdn` cannot be set. The default is unset.
  - `RootPW` *bindPW*
    Specifies the password that `rootDN` must use to bind. It may be given with a scheme defining the password type. The default is unset. The password is of the format `{`*scheme*`}`*passwdString* where *scheme* is one of `CLEARTEXT`, `CRYPT` (same as `UNIX`), `MD5`, `NS-MTA-MD5`, `SMD5`, `SHA`, or `SSHA`. The format of *passwdString* is dependent on scheme. If *scheme* is absent, *passwdString* is assumed to be cleartext.
  - `SchemaCheck on|off`
    Controls whether records are verified to match the schema upon database modifications (on) or to ignore schema verification (off). The default is to

check records, but this may be turned off to improve update performance.

- `SizeLimit` *`count`*
  Maximum number of entries to return from search operations. Default 100.
- `TimeLimit` *`seconds`*
  Maximum number of seconds that the server devotes to a search request before returning an error (LDAP query timeout). Default 10.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
16 Dir SetDbOption database1 Chaining on
16 OK Completed

17 Dir SetDbOption database1 IdleTimeout 360
17 OK Completed

18 Dir SetDbOption database1 Proxyauth on
18 OK Completed

19 Dir SetDbOption database1 RdnIntegrity implicit
19 OK Completed

20 Dir SetDbOption database1 ReadOnly on
20 OK Completed

21 Dir SetDbOption database1 RefOnly on
21 OK Completed

22 Dir SetDbOption database1 Referral ldap://ldap.mirapoint.com:389/
22 OK Completed

23 Dir SetDbOption database1 RootDN uid=root,dc=exapmle,dc=com
23 OK Completed

24 Dir SetDbOption database1 RootPW {CRYPT}k1llk44coo0f00xf
24 OK Completed

25 Dir SetDbOption database1 SchemaCheck off
25 OK Completed

26 Dir SetDbOption database1 SizeLimit 600
26 OK Completed

27 Dir SetDbOption database1 TimeLimit 60
27 OK Completed
```

Bulk-Data Subcommands

These subcommands allow bulk export, modification, and bulk import of database content using LDIF (LDAP Data Interchange Format). `ImportLdif` is a bulk add facility that automatically takes the database off-line to guarantee consistency. `ModifyLdif` executes changes against a live database.

ExportLdif

The `ExportLdif` command retrieves data from the LDAP server. Records matching a specified suffix are returned as one long length-encoded string. If no suffix is given (an empty string is given) then the database contents are returned. Data format is LDIF, as specified in RFC 2849.

Syntax

`tag` `Dir ExportLDIF` *`suffix flag`*

where:

- *`suffix`* is the ending portion of a distinguished name to match database entries.
- *`flag`* may be null or "`c`" to indicate continue (instead of exit) on error, or "`o`" to export operational attributes. Operational attributes were exported by default before Release 3.6, but since they cannot be imported by `Dir ImportLdif`, they are no longer exported, except when requested for backup by the "`o`" flag.

Privilege Levels

Administrator

Domain Sensitivity

- Administrator
- Backup operator

Example

```
28 Dir ExportLdif dc=hb,dc=com ""
* 28 {460}
dn: dc=hb,dc=com
objectClass: top

dn: cn=Fred Flintstone,dc=hb,dc=com
objectClass: top
objectClass: MirapointUser
objectClass: person
cn: Fred Flintstone
sn: Flintstone

dn: cn=Barney Rubble,dc=hb,dc=com
objectClass: top
objectClass: MirapointUser
objectClass: person
cn: Barney Rubble
sn: Rubble

28 OK Completed
```

ImportLdif

The `ImportLdif` command uploads data to the directory server. Records are added incrementally so any failure may cause termination with the server database only partially updated. Adds are supported, but not modifications, deletes, or modDN. Data format is LDIF, as specified in RFC 2849.

Replication agreements do not copy data set by `ImportLdif` (they do copy changes by `ModifyLdif`). After `ImportLdif`, Mirapoint advises you either to run `Replicate` manually, or `ImportLdif` on the replica server too.

Syntax

tag `Dir ImportLDIF` *suffix|database flag value*

where:

- *database* is the name of a database returned by `Dir Listdb`. Only records that would be inserted into the specified database are imported; other records are silently discarded.
- *suffix* is the ending portion of a distinguished name. Only records whose distinguished name attribute ends with this suffix are imported; other records are silently discarded.
- *flag* may be:
  - the null string (the default) means exit in case of error.
  - "`c`" indicates continue importing data after an error.
  - "`l`" (letter L) indicates live replacement of data. Information gets loaded into a temporary database, while the directory server continues running. After import completes successfully, the server is stopped and affected databases are replaced. If used in conjunction with the "`c`" flag, non-critical errors (such as schema violations) do not cause the import to fail. However using "`c`" and "`l`" together may result in an empty database. Limited to hosts with no replication agreements. With the "`l`" flag, *suffix* must name a database, because live import cannot replace just a subtree.
- *value* is the bulk data to import, in literal string format, or a URL pointing to the data.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
29 Dir ImportLdif dc=hb,dc=com c {214+}
dn: cn=Wilma Flintstone,dc=hb,dc=com
objectclass: top
objectclass: MirapointUser
objectclass: person
cn: Wilma Flintstone
sn: Flintstone
* 29 INFO "adding data"
29 OK Completed
```

```
30 Dir ImportLdif dc=hb,dc=com c ftp://server/file
* 30 INFO "fetching data ftp://server/file"
* 30 INFO "adding data"
30 OK Completed
```

ModifyLdif

The `ModifyLdif` command is similar to the `ImportLdif` command, but accepts modification items in the data, and does not require the server to be stopped. The `ModifyLdif` command might be more convenient, but the `ImportLdif` command performs initial loads more quickly.

Directory entries can be modified by inserting one of the following `changetype` attributes into LDIF data entries.

- `add`—inserts an entire new entry where none existed before
- `delete`—removes the entire entry if it exists in the directory information tree
- `modify`—changes the entry attributes line by line, possibly adding or deleting
- `modrDN`—changes the relative distinguished name of an entry (moves it)
- `modDN`—changes the distinguished name of an entry (moves it)

Modifications can have entries without change types, in which case they are controlled by the `flag` (see below).

Syntax

`tag` `Dir ModifyLDIF` *`suffix flag value`*

where:

- *`suffix`* is the ending portion of a distinguished name, where information starts in a lookup.
- *`flag`* may be null or one of the following:
  - `c` to indicate continue (instead of exit) on error.
  - `a` meaning add when modification type is missing in records.
  - `r` meaning replace when modification type is missing in records.
- *`value`* is the bulk data to change, in literal string format, or a URL pointing to the data.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

The first CLI example changes User1's mail quota. The second changes an entry's relative distinguished name. The third moves all of the entry's children to a new location in the directory tree (LDAPv3 only).

```
Dir ModifyLdif o=miratop c
Enter LDIF directory data, finish with '.' on a line by itself:
dn: miLoginid=user1,miDomainName=primary,ou=domains,o=miratop
changetype: modify
replace: mimailquota
mimailquota: 10340000
.
INFO "modifying data"
INFO "1 of 1 successful"
OK Completed
```

```
Dir ModifyLdif dc=example,dc=com c
Enter LDIF directory data, finish with '.' on a line by itself:
dn: cn=Paul Jensen, ou=Product Development, dc=example, dc=com
changetype: modrdn
newrdn: cn=Paula Jensen
deleteoldrdn: 1
.
```

```
Dir ModifyLdif dc=example,dc=com c
Enter LDIF directory data, finish with '.' on a line by itself:
dn: ou=PD Accountants, ou=Product Development, dc=example, dc=com
changetype: modrdn
newrdn: ou=Product Development Accountants
deleteoldrdn: 0
newsuperior: ou=Accounting, dc=example, dc=com
.
```

```
31 Dir ModifyLdif dc=yyy,dc=zzz "c" ftp://server/file
* 31 INFO "Fetching ftp://server/file"
* 31 INFO "Verifying ldif"
* 31 INFO "Modifying ldif data"
31 OK Completed
```

Schema and Index Subcommands

Using `AddSchema` and related commands, custom schema information may be added in the form of files that get included into the Mirapoint configuration file. Schema files define attributes and object classes used in LDAP entries. Schemas `core`, `cosine`, `inetorgperson`, `misc`, and `mirapoint` are preinstalled with the product and may not be removed.

Using `AddIndex` and related commands, different individual attributes can be indexed for fast searching of each database.

AddSchema

The `AddSchema` command extends the standard schema shipped with Mirapoint directory server. The schema format is specified in RFC 2252. Schema ordering is important, so new ones must be added before deleting old ones, keeping the system always complete. Verification of the overall schema is done as part of the add.

To match for `EQUALITY`, two values must be identical, employ the same attribute syntax, and conform to the data type of the attribute syntax. Schemas often specify CaseIgnoreMatch for case-insensitive equality, but (at least) the following are also supported: BooleanMatch, CaseExactMatch, IntegerMatch, NumericStringMatch, ObjectIdentifierMatch, TelephoneNumberMatch, and UniqueMemberMatch.

To match for `ORDERING`, syntax must be open to comparisons of less than, equal to, and greater than. For example, 50 is less than 100, and N is greater than B.

To match `SUBSTRING`, a syntax must be open to search and comparison patterns that include the asterisk (*) wildcard. For example, in a syntax using substring matching, N*V*L would match naval, navel, or novel.

In the schema attribute definitions, the `SYNTAX` keyword specifies OID (object ID), and takes an optional suffix {*number*} inside curly braces, indicating the maximum length of this field in bytes.

Attribute definitions can be ended by the `SINGLE-VALUE` option to indicate that this is not a list but must take one value. Unfortunately format requirements are rigid: you cannot have a close parenthesis on a line by itself.

Schema are not copied by replication agreements. Before setting up a replica server, add any schema that you created on the master server.

Syntax

`tag` `Dir AddSchema` *`schemaname value`*

where:

- *`schemaname`* is your name for the new schema, such as `custom`, which must not conflict with a pre-existing schema name.
- *`value`* is the contents of the new schema in literal string format, or a URL.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
32 Dir AddSchema local {204+}
attributetype ( 1.3.6.1.4.1.3246.249.6.1 NAME 'newschema'
        EQUALITY caseIgnoreMatch
        ORDERING caseIgnoreOrderingMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
        SINGLE-VALUE )
* 32 INFO "verifying data"
32 OK Completed
```

```
33 Dir AddSchema remote ftp://server/file
* 33 INFO "fetching ftp://server/file"
* 33 INFO "verifying schema"
```

```
33 OK Completed
```

CountSchema

The `CountSchema` command returns the number of current schema instances that match the given pattern.

Syntax

`*tag* Dir CountSchema *pattern*`

where *`pattern`* is a string, possibly containing wildcards, to match schema names. The null string means match all schema names.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
34 Dir CountSchema ""
* 34 7
34 OK Completed
```

DeleteSchema

The `DeleteSchema` command deletes the named schema from the schema list. The standard (factory-supplied) schemas may not be removed. Schemas that contain data used by other schemas may not be removed either.

Syntax

`*tag* Dir DeleteSchema *schemaname*`

where *`schemaname`* is the name of an existing schema to remove.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
35 Dir DeleteSchema local
35 OK Completed
```

GetSchema

The `GetSchema` command returns the full named schema file as a length-encoded string. The schema format is specified in RFC 2252.

Syntax

tag `Dir GetSchema` *schemaname*

where *schemaname* is the name of an existing schema to retrieve.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
36 Dir GetSchema local
* 36 {205}
attributetype ( 1.3.6.1.4.1.3246.249.6.1 NAME 'newschema'
        EQUALITY caseIgnoreMatch
        ORDERING caseIgnoreOrderingMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
        SINGLE-VALUE )
36 OK Completed
```

ListSchema

The `ListSchema` command returns the names of schemas matching a given pattern possibly within the range specified. To show schema values, use the `GetSchema` command. The ordering of schemas is important so names are returned in the order that they were added.

Syntax

tag `Dir ListSchema` *pattern start count*

where *pattern* is string, possibly containing wildcards, to match schema names. The null string matches all names; *start* and *count* enumerate begin and end.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
37 Dir ListSchema "" "" ""
* 37 core
* 37 cosine
* 37 inetorgperson
* 37 misc
* 37 mgrp
* 37 mirapoint
* 37 local
37 OK Completed
```

SetSchema

The `SetSchema` command changes the value of a schema. The data format is as specified in RFC 2252. Setting schema contents to an empty value is equivalent to deleting the schema.

Schema are not copied by replication agreements. Before setting up a replica server, set any schema that you set on the master server.

Syntax

tag `Dir SetSchema` *schemaname value*

where:

- *schemaname* is the name of an existing schema to replace.
- *value* is the new schema contents in literal string format, or a URL.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
38 Dir SetSchema local {342+}
attributetype ( 1.3.6.1.4.1.3246.249.6.1 NAME 'newattr'
        EQUALITY caseIgnoreMatch
        ORDERING caseIgnoreOrderingMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
        SINGLE-VALUE )
objectclass ( 1.3.6.1.4.1.3246.249.6.10 NAME 'newobjclass' SUP top
        MUST ( objectClass $ newattr )
        MAY ( description ) )
* 38 INFO "verifying data"
38 OK Completed
```

```
39 Dir SetSchema remote ftp://server/file
* 39 INFO "fetching data: ftp://server/file"
* 39 INFO "verifying data"
39 OK Completed
```

AddIndex

The AddIndex command creates an index for the specified attribute and type. If the database name is an empty string then the rule is a default applied to all databases. Defaults for all databases are preconfigured on new installations as follows:

```
maillocaladdress  eq
mail              eq,pres
cn                eq,pres
sn                eq,pres
uid               eq,pres
objectclass       eq,pres
```

Indexes are not copied by replication agreements. After configuring a replica server, add any indexes that might be required for LDAP searches.

Syntax

tag Dir AddIndex *dbname attribute types*

where:

- *dbname* is the directory database instance name.
- *attribute* is an attribute name in the LDAP record.
- *types* is a (possibly comma-separated) list of attributes types as follows:
  - approx—approximates (sounds like); if an approximate index was not generated for an attribute, then approx searches default to equality searches
  - autolang—index language extensions of attribute instance, for example cn;lang-es for Español in addition to cn for common name (same as lang)
  - autosubtypes—index all subtypes of attribute (same as subtypes)
  - eq—equals (equality search)
  - none—no indexes for this attribute
  - pres—present in
  - sub—substring (same as subany or substr)
  - subfinal—word ends with
  - subinitial—word begins with

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
40 Dir AddIndex "" uid eq,approx
* 40 INFO "reindexing: database1"
* 40 INFO "reindexing: database2"
40 OK Completed
```

CountIndex

The `CountIndex` command returns the number of indexed attributes that match the given pattern.

Syntax

tag `Dir CountIndex` *dbname pattern*

where:

- *dbname* is the directory database instance name. If *dbname* is blank (“ “), the count is the total number of globally applied indexes.
- *pattern* is string, possibly containing wildcards, to match indexed attributes. The null string matches everything.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
41 Dir CountIndex database1 "mail*"
* 41 1
41 OK Completed
```

DeleteIndex

The `DeleteIndex` command stops an attribute from being indexed, and removes any currently existing index data for that attribute.

Syntax

tag `Dir DeleteIndex` *dbname attribute*

where:

- *dbname* is the directory database instance name. If *dbname* is blank (“ “), this command deletes the global default index for the named attribute.
- *attribute* is the LDAP attribute name for which to remove indexing.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
42 Dir DeleteIndex database1 uid
42 OK Completed
```

GetIndex

The `GetIndex` command returns a single item from the index list. It is the same as `ListIndex` with an absolute name in the pattern, but is included for completeness.

Syntax

```
tag Dir GetIndex dbname attribute
```

where:

- *dbname* is the directory database instance name.
- *attribute* is the LDAP attribute name for which to remove indexing.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
43 Dir GetIndex database1 uid
* 43 database1 uid eq,approx
43 OK Completed
```

ListIndex

The `ListIndex` command returns all matching attributes within the range specified by start and count.

Syntax

tag Dir ListIndex *dbname pattern start count*

where:

- *dbname* is the directory database instance name. If *dbname* is blank (“ “), this command lists the global default indexes for that attribute pattern.
- *pattern* is string, possibly containing wildcards, to match LDAP attributes. The null string matches all names; *start* and *count* enumerate begin and end.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
44 Dir ListIndex database1 "" "" ""
* 44 database1 mailLocalAddress eq
* 44 database1 uid eq,approx
44 OK Completed
```

SetIndex

The SetIndex command replaces an item in the index list. This has the same effect as calling DeleteIndex then AddIndex. Setting the index to an empty string is the same as deleting the index. Setting the index type to none removes all indexing for the database.

Indexes are not copied by replication agreements. After configuring a replica server, reset any indexes that might be required for LDAP searches.

Syntax

tag Dir SetIndex *dbname attribute types*

where:

- *dbname* is the directory database instance name.
- *attribute* is an attribute name in the LDAP record.
- *types* is a (possibly comma-separated) list of attributes types as described under the Dir AddIndex subcommand.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
45 Dir SetIndex database1 uid eq,approx
* 45 INFO "reindexing: database1"
45 OK Completed
```

Replication Subcommands

These subcommands control the replication of databases between Mirapoint directory servers. A replication agreement is an arbitrary name that represents a grouping. The group contains a list of hosts, with a set of options that control which data are replicated, when, and to whom. Ordinarily replication occurs after any change in data.

The following items are not copied by a replication agreement: schemas, indexes, ACLs, and object attributes set by `ImportLdif`. Whereas object attributes changed by `ModifyLdif` are replicated automatically. After `ImportLdif`, Mirapoint advises you to run `Replicate` manually, or run `ImportLdif` on the replica server too. When setting up a replica server, add any schema or ACLs that you created on the master server. Indexes may be established as needed.

When upgrading replicated directory servers, upgrade the master first. Shut down the slave servers before starting directory service on the master. Upgrade and start the slave severs, then run `Dir Replicate ... Full`.

AddReplica

The `AddReplica` command adds a replication agreement to the server's list. The replication protocol type determines the meaning of replica options and behavior. By default, replication occurs immediately after any change in data.

LDAP replication is a good way to create a backup database in case the first fails. Here are commands for replicating a Mirapoint LDAP server. This presupposes a second LDAP server (ldap2) configured with Directory Server license. The name of the replication agreement (RA) is arbitrary. DN `o=miratop` is typical for Mirapoint.

```
> Dir Addreplica RA Mirapoint o=miratop o=miratop
> Dir Addrephost RA ldap://ldap2.example.com
> Dir Replicate RA ldap://ldap2.example.com Full
```

Syntax

tag `Dir AddReplica` *raname type masterbase slavebase*

where:

- *raname* is the replication agreement name.
- *type* is the replication type:

- `mirapoint`—either server or slave
- `iplanet`—slave only, not guaranteed to work
- `netscape`—slave only, not guaranteed to work
- `openldap`—does not work with newer versions of OpenLDAP

◆ *`masterbase`* is a distinguished name that specifies the top of the subtree on the master server where replication begins. If *`masterbase`* is not given, the default is to replicate the entire local tree.

◆ *`slavebase`* is a distinguished name that specifies the base of the remote tree that corresponds to the master. In each entry, the master base is stripped from the DN and replaced with the slave base. If *`slavebase`* is not provided, the default is to use the master base subtree.

To change a *`masterbase`* and *`slavebase`* pair, the replica must be deleted, and a new replica created with new master base and slave base values.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
48 Dir AddReplica replica1 "mirapoint" dc=corp,dc=com dc=newco,dc=com
48 OK Completed

49 Dir AddReplica replica2 "iplanet" "" o=baseball,c=us
49 OK Completed
```

CountReplica

The `CountReplica` command returns the number of replication agreement entries that match the given pattern.

Syntax

`tag` `Dir CountReplica` *`pattern`*

where *`pattern`* is string, possibly containing wildcards, to match replication agreement names. The null string matches all names.

Privilege Levels

◆ Administrator

◆ Backup operator

Domain Sensitivity

None

Example

```
51 Dir CountReplica ""
* 51 2
51 OK Completed
```

DeleteReplica

The `DeleteReplica` command removes the given replication agreement from the system list, removes the master base and slave base association for this replica, and removes any RepHost or RepOption information associated with this replica.

Syntax

```
tag Dir DeleteReplica raname
```

where *raname* is a replication agreement name previously added with `AddReplica`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
52 Dir DeleteReplica replica1
52 OK Completed
```

ListReplica

The `ListReplica` command displays matching replication agreements within the range specified by start and count. The output of the list shows the replica name, replica type, replica master base, and replica slave base.

Syntax

```
tag Dir ListReplica pattern start count
```

where *pattern* is string, possibly containing wildcards, to match replication names. The null string matches all names; *start* and *count* enumerate begin and end.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
53 Dir ListReplica "" "" ""
* 53 replica1 "mirapoint" "dc=corp,dc=com"  "dc=newco,dc=com"
* 53 replica2 "iplanet" "" "o=baseball,c=us"
53 OK Completed
```

AddRepHost

The `AddRepHost` command adds a slave system to a replication agreement, which must have been previously created with the `AddReplica` command.

Syntax

tag `Dir AddRepHost` *raname URL*

where:

- *raname* is the replication agreement name.
- *URL* is the LDAP uniform resource locator, `ldap://slave` for example.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
54 Dir AddRepHost replica1 ldap://slave1
54 OK Completed
```

```
55 Dir AddRepHost replica2 ldaps://slave2:2222
55 OK Completed
```

CountRepHost

The `CountRepHost` command returns the number of entries in the replication host list that match the supplied pattern.

Syntax

tag `Dir CountRepHost` *raname pattern*

where:

- *raname* is the replication agreement name.
- *pattern* is string, possibly containing wildcards, to match replication hosts. The null string matches all names.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
56 Dir CountRepHost replica1 ""
* 56 2
56 OK Completed
```

DeleteRepHost

The `DeleteRepHost` command removes a slave from the replication agreement.

Syntax

tag `Dir DeleteRepHost` *raname URL*

where:

- *raname* is the replication agreement name.
- *URL* is the LDAP uniform resource locator, `ldap://slave` for example.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
57 Dir DeleteRepHost replica1 ldap://slave1
* 57 INFO "removing replication log: ldap://slave1
57 OK Completed
```

ListRepHost

The `ListRepHost` command returns items from the replication host list that match the given pattern within the range specified by start and count.

Syntax

tag `Dir ListRepHost` *raname pattern start count*

where:

- *raname* is the replication agreement name.
- *pattern* is string, possibly containing wildcards, to match replication hosts. The null string matches all names; *start* and *count* enumerate begin and end.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
58 Dir ListRepHost replica1 "" "" ""
* 58 replica1 ldap://slave1
* 58 replica1 ldaps://slave2:2222
58 OK Completed
```

CountRepOption

The `CountRepOption` command returns the number of replication options that match the given pattern.

Syntax

tag `Dir CountRepOption` *raname pattern*

where:

- *raname* is the replication agreement name. If *raname* is blank (“ “), counts all replication options.
- *pattern* is string, possibly containing wildcards, to match replication options. The null string matches all names.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
59 Dir CountRepOption replica1 ““
* 59 7
59 OK Completed
```

GetRepOption

The `GetRepOption` command returns one named option from the option list. It is the same as `ListRepOption` with single pattern, but is included for completeness.

Syntax

```
tag Dir GetRepOption raname option
```

where:

- *raname* is the replication agreement name. If *raname* is blank (“ “), shows the default value for this replication option.
- *option* is a replication option as specified by the `SetRepOption` subcommand.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
60 Dir GetRepOption replica1 trigger
* 60 replica1 trigger userPassword
60 OK Completed
```

ListRepOption

The `ListRepOption` command returns replication option settings matching the given pattern within the range specified by start and count.

Syntax

```
tag Dir ListRepOption raname pattern start count
```

where:

- *raname* is the replication agreement name. If *raname* is blank (“ “), lists the default options for all replication options.
- *pattern* is string, possibly containing wildcards, to match replication options. The null string matches all names; *start* and *count* enumerate begin and end.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
61 Dir ListRepOption replica1 "" "" ""
* 61 master ldap://ldap.mirapoint.com:389
* 61 schedule "0,15,30,45 0-16 * * 1-5"
* 61 trigger userPassword
* 61 binddn uid=root,dc=mirapoint,dc=com
* 61 bindpw secret
61 OK Completed
```

SetRepOption

The `SetRepOption` command adds options to control the behavior of a replication agreement. If the replica name is omitted then the option is used as the default for all replication agreements. Each option takes a single value that is option-specific. Setting an option to the empty string removes the option.

Syntax

tag `Dir SetRepOption` *raname option value*

where:

- *raname* is the replication agreement name.
- *option* is one of the following replication options with related *value*:
  - `Master` *URL*
    The master is the authoritative server for this tree. A referral to the listed URL is returned for all updates sent to a nonmaster server. The master must always be specified.
  - `BindDN` *DN*
    This is the distinguished name used for binding to this host for replication. This option must be specified.
  - `BindPW` *password*
    This is the password to use during the bind operation. This option must be specified.
  - `Exclude` *attribute-list*
    If an exclude list is specified, then the listed attributes are explicitly removed for objects before a slave is updated. This allows stripping of sensitive data from a set of slave servers. The default is to send all attributes.
  - `Schedule` *time*
    The time option sets the schedule when items are sent to this replica server. If no time list is set, then updates are sent immediately. The *time* is specified in `crontab` format: numbers indicate minute, hour, day, month, or weekday, and can be * to indicate the entire range. For example (working backwards) this means Monday through Friday, all year, any day of the month, from 8:15 AM to 6:45 PM, every half hour.

`15,45 8-18 * * 1-5`

- `Trigger` *`attribute-list`*
  If one of the trigger attributes changes, then the entry is sent to the replica immediately. This is useful for immediate propagation of password changes, among other uses. The default is to send changes immediately.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
62 Dir SetRepOption replica1 Master ldap://ldap.mirapoint.com:389
62 OK Completed

63 Dir SetRepOption replica1 BindDN uid=root,dc=example,dc=com
63 OK Completed

64 Dir SetRepOption replica1 BindPW secret
64 OK Completed

65 Dir SetRepOption replica1 Exclude userPassword
65 OK Completed

66 Dir SetRepOption replica1 Schedule "0,15,30,45 0-16 * * 1-5"
66 OK Completed

67 Dir SetRepOption replica1 Trigger userPassword
67 OK Completed
```

Replicate

The `Replicate` command requests an immediate update of the specified agreement. On a slave system the server is contacted to request an update, and on the server one or more client updates are scheduled. This command schedules a replication, but does not wait for the replication to complete.

Syntax

`tag` `Dir Replicate` *`raname url protocol`*

where:

- *`raname`* is the replication agreement name. The null string (“ “) indicates all agreements that include this replication host.
- *`url`* is the URL of the replication host. The null string (“ “) indicates all hosts.
- *`protocol`* is the replication protocol, either `full` or `incremental`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
46 Dir Replicate replica1 "" full
46 OK Completed

47 Dir Replicate replica1 ldap://slave incremental
47 OK Completed
```

Promote

Promotes replication slave to the master. First verifies that a replication agreement already exists, and checks whether the given host is a current slave server for the agreement. If so, it exchanges the current master with the given slave host. If this is the local host, after server restart the database becomes writable and changes are logged for replication. If the given slave host is nonlocal, changes the host contacted for replication updates, or from which updates are allowed. If the local host was the master, then it becomes a slave and its replication logs are removed.

Syntax

```
tag Dir Promote raname url
```

where:

- `raname` is the replication agreement name. The null string (“ “) indicates all agreements that include this replication host.
- `url` is the URL of the replication host. The null string (“ “) indicates all hosts.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
88 Dir Promote replica1 ldap://slave1
88 OK Completed

89 Dir Promote replica9 ldap://slave1
89 NO replication agreement does not exist
```

Repstat

Displays status of the replication agreement. This information includes the last CSN (change sequence number) transmitted, the last time synchronization occurred, and the last error message. On a slave only local information displays; on the master information about each host can be displayed. The system with the highest CSN is the most up-to-date system and should be selected for promotion.

Syntax

tag `Dir Repstat` *raname url*

where:

- *raname* is the replication agreement name. The null string (“ “) indicates all agreements that include this replication host.
- *url* is the URL of the replication host. The null string (“ “) indicates all hosts.

Output of `Repstat` includes the host URL, the CSN (defined by ISO 8824-1:1995 and including year *yyyy* month *mm* day *dd* hour *hh* : minute *mm* : second *ss* Z# *counter # replicaID # namespace*), timestamp of message, and message string.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
90 Dir Repstat replica1 ldap://slave1
* 90 ldap://slave1 2001040521:22:10Z#0x0007#0#0000 14:26:10 "Full replication completed"
* 90 ldap://slave2 2001040521:22:10Z#0x0001#0#0000 14:26:10 "Failed to bind to slave"
90 OK Completed
```

Access Control Subcommands

These subcommands manage access control lists (ACLs) on the server. ACLs control who has access to LDAP data.

Directory ACLs are complicated. Here are some recommended settings to allow anonymous binds from the network, without granting read access to any objects:

```
Dir Addacl "" default *
Dir Setaclentry "" default example peername=ip=208\.77\.188\.166 read break
Dir Setaclentry "" default localhost peername=ip=127\.0\.0\.1 read break
Dir Setaclentry "" default users users read stop
Dir Setaclentry "" default unbound anonymous auth stop
Dir Setaclentry "" default internet * none stop

Dir Addacl "" password attr=userPassword
Dir Setaclentry "" password owner self write stop
Dir Setaclentry "" password others * compare stop

Dir Setaclentry "" userdata self self write stop
Dir Setaclentry "" userdata others * read stop

Dir Setaclentry "" read all * read stop
Dir Deleteacl ““ passwd
```

AddACL

The `AddACL` command extends the server's access control list. If an ACL contains a database name, the ACL is relevant only for that database, otherwise it is global. Every request is passed through the ACL to verify that the client has permission.

The ACL is split into two pieces: a pattern to match the object, and a list of patterns to match users added by the `AddAclEntry` command. The first pattern that matches an object determines access privileges unless overridden by the control field of an ACL entry. The following default global ACL is preconfigured as follows:

- `passwd`—`"(attr=userpassword)"`
- `userdata`—a multitude of WebMail preferences; Autoreply text, interval, and subject; Forwarding address and delivery options; and Mirapoint unique UID: `"(attr=miWmpref*,miAutoreply*,miForwardingAddress,miUUID)"`
- `read`—`"*"`

ACLs are not copied by replication agreements. Before setting up a replica server, add any ACLs that you created on the master server.

Syntax

tag `Dir AddACL` *dbname aclname what*

where:

- *dbname* is the directory database instance name. If *dbname* is blank (" "), this ACL applies to all databases.
- *aclname* is the name of the access control list.
- *what* is the specific setting for that access control list. Possible arguments include one or a combination of the following elements:

  ```
  *
  (attr=attributeList)
  (dn[.style]=regularExpression)
  (filter=LDAPfilter)
  ```

  where *style*=`regex|base|one|subtree|children`

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
68 Dir AddACL database1 example "(dn.subtree=dc=example,dc=com')"
68 OK Completed
69 Dir AddACL database1 self "(attr=member,entry)"
69 OK Completed
```

```
70 Dir AddACL database1 uofm "(dn='.*,o=U of M,c=US')(attr=homePhone)"
70 OK Completed
```

CountACL

The `CountACL` command returns the number of ACL entries that match a pattern.

Syntax

tag `Dir CountACL` *dbname pattern*

where:

- *dbname* is the directory database instance name. If *dbname* is blank (“ “), counts global ACLs across all databases.
- *pattern* is string, possibly containing wildcards, to match ACL names. The null string (“ “) matches all names.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
71 Dir CountACL database1 ““
* 71 3
71 OK Completed
```

DeleteACL

The `DeleteACL` command removes an item from the specified access control list.

Syntax

tag `Dir DeleteACL` *dbname aclname*

where:

- *dbname* is the directory database instance name. If *dbname* is blank (“ “), deletes this ACL from the global list of ACLs.
- *aclname* is the name of the access control list.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
**72 Dir DeleteACL database1 self**
72 OK Completed
```

GetACL

The `GetACL` command prints one requested item from the access control list. It is the same as `ListACL` with single pattern, but is included for completeness. For details about ACL format, see AddACL on page 191.

Syntax

tag `Dir GetACL` *dbname aclname*

where:

- *dbname* is the directory database instance name. If *dbname* is blank (“ “), gets this ACL from the global list of ACLs.
- *aclname* is the name of the access control list.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
73 Dir GetACL database1 ““
* 73 database1 example “(dn=.*,dc=example,dc=com)”
73 OK Completed
```

ListACL

The `ListACL` command returns the access control list entries that match the given pattern within the range specified by start and count. For details about ACL format, see AddACL on page 191.

Syntax

tag `Dir ListACL` *dbname pattern start count*

where:

- *dbname* is the directory database instance name. If *dbname* is blank (" "), shows ACLs from the list of global ACLs that match the pattern.
- *pattern* is string, possibly containing wildcards, to match replication options. The null string matches all names; *start* and *count* enumerate begin and end.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
74 Dir ListACL database1 "" "" ""
* 74 database1 example "(dn='.*,dc=example,dc=com')"
* 74 database1 self "(attr=member,entry)"
* 74 database1 uofm "(dn='.*,o=U of M,c=US')(attr=homePhone)"
74 OK Completed
```

SetACL

The SetACL command adds or replaces an existing entry in the access control list. The ordering of items in the access control list is critical. Setting an item to the empty string results in deleting the entry. For information about ACL format, see AddACL on page 191.

ACLs are not copied by replication agreements. Before setting up a replica server, set any ACLs that you set on the master server.

Syntax

```
tag Dir SetACL dbname aclname what
```

where:

- *dbname* is the directory database instance name. If *dbname* is blank (" "), sets this ACL as a global ACL for all databases.
- *aclname* is the name of the access control list.
- *what* is the specific setting for that access control list. Possible arguments include one or a combination of the following elements:

```
*
(dn=regularExpression)
(filter=LDAPfilter)
(attr=attributeList)
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
75 Dir SetACL "" password "(attr=userPassword)"
75 OK Completed
```

AddAclEntry

The `AddAclEntry` command adds an access list entry to the named access list. The pattern in the access list chooses objects that are covered by this list, and entries on the entry set access rights for specific groups of users.

Adding an ACL defines what will be accessed. Adding an ACL entry defines who gets what kind of access to the added ACL. For example, the passwords of a subtree defined by its distinguished name are defined as what to access:

```
dir addacl db2 newacl "(dn=ou=group1,dc=example,dc=com)(attr=passwd)"
```

Adding an ACL entry defines the person (LDAP Member object) or group of people (groupOfNames) who have specific kinds of access rights. You could create an ACL entry that allows each user belonging to a subtree to edit their own password only (and nothing else) as follows:

```
dir addAclEntry db2 newacl ownpasswds self write ""
dir addAclEntry db2 newacl nooneelse * none ""
```

Syntax

tag `Dir AddAclEntry` *dbname aclname entryname who level control*

where:

- *dbname* is the directory database instance name. If *dbname* is blank (""), this adds an entry for a global ACL.
- *aclname* is the name of the access control list.
- *entryname* is the name of the item to add to the ACL.
- *who* specifies the users who are allowed to do what, in the following syntax,

  ```
  (* | anonymous | users | self | [(dn[.style]=regex)] [(dnattr=attrname)]
  [(group[/objectclass[/attrname]][.substyle]=regex)]
  [(domain[.substyle]=regex)] [(sockurl[.substyle]=regex)]
  [(peername[.substyle]=regex)] [(sockname[.substyle]=regex)] )
  ```

  where square brackets [...] indicate optional, style is one of the following, and substyle is only one of the first two:

  ```
  regex | exact | base | one | subtree | children
  ```

- *level* specifies what users are allowed to do. Permissions are listed below in hierarchical order: `none` means no access, `auth` allows authentication only, `compare` adds database compare, `search` adds the ability to locate items but not return data, `read` adds return of attribute value data, `selfwrite` permits a user to change data owned by the user's bound DN (used for groupOfNames), and `write` adds the ability to change any item in the database.

```
none | auth | compare | search | read | selfwrite | write
```

- *control* governs program flow and may be either " " (the null string), `stop`, `continue`, or `break`.

On new installations a default access list is added containing the following entries:

```
aclname   entryname   who    level    control
passwd    self        self   write    ""
passwd    others      *      compare  ""
read      all         *      read     ""
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
76 Dir AddAclEntry db1 example ownerentry "self" write ""
76 OK Completed

77 Dir AddAclEntry db1 example groupentry "dn.subtree=dc=eg,dc=com" search ""
77 OK Completed

78 Dir AddAclEntry db1 otheracl memberentry "dnattr=member" selfwrite ""
78 OK Completed

95 Dir AddAcl "" userdata "(attr=miWmpref*,miAutoreply*,miForwardingAddress,miUUID)"
95 OK Completed
96 Dir AddAclEntry "" userdata self self write ""
96 OK Completed
97 Dir AddAclEntry "" userdata others * read ""
97 OK Completed
```

In #95, the "*" represents multiple attribute names that must all be given explicitly.

CountAclEntry

The `CountAclEntry` command returns the number of access list entries that match a given pattern in the specified ACL.

Syntax

```
tag Dir CountAclEntry dbname aclname pattern
```

where:

- *dbname* is the directory database instance name. If *dbname* is blank (" "), counts ACL entries from the global list of ACLs.
- *aclname* is the name of the access control list.

- *pattern* is string, possibly containing wildcards, to match ACL entry names. The null string matches all names.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
79 Dir CountAclEntry database1 example ""
* 79 2
79 OK Completed '
```

DeleteAclEntry

The `DeleteAclEntry` command removes an item from the specified ACL.

Syntax

```
tag Dir DeleteAclEntry dbname aclname entryname
```

where:

- *dbname* is the directory database instance name. If *dbname* is blank (" "), deletes the named ACL entry from the global list of ACLs.
- *aclname* is the name of the access control list.
- *entryname* is the name of the item to delete from the ACL.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
80 Dir DeleteAclEntry database1 example groupentry
80 OK Completed
```

GetAclEntry

The `GetAclEntry` command prints a single requested item from the ACL. It is the same as `ListAclEntry` with single pattern, but is included for completeness. For details about ACL entry format, see AddAclEntry on page 195.

Syntax

tag `Dir GetAclEntry` *dbname aclname entryname*

where:

- *dbname* is the directory database instance name. If *dbname* is blank (" "), gets the named ACL entry from the global list of ACLs.
- *aclname* is the name of the access control list.
- *entryname* is the name of the item to get from the ACL.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
81 Dir GetAclEntry db1 example groupentry
* 81 db1 example groupentry "dn=\*,dc=example,dc=com" search ""
81 OK Completed
```

ListAclEntry

The `ListAclEntry` command returns the ACL entries that match the given pattern within the range specified by start and count parameters in the named list. For details about ACL entry format, see AddAclEntry on page 195.

Syntax

tag `Dir ListAclEntry` *dbname aclname pattern start count*

where:

- *dbname* is the directory database instance name. If *dbname* is blank (" "), lists ACL entries in the global list of ACLs that match the pattern.
- *aclname* is the name of the access control list.
- *pattern* is string, possibly containing wildcards, to match ACL entries. The null string matches all names; *start* and *count* enumerate begin and end.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
82 Dir ListAclEntry db1 example "" "" ""
* 82 db1 example ownerentry self write ""
* 82 db1 example groupentry "dn=\*,dc=example,dc=com" search ""
82 OK Completed
```

SetAclEntry

The `SetAclEntry` command adds an entry to the ACL, or replaces an existing entry. The ordering of items in the access control list is critical. Setting an item to the empty string results in deleting the entry. For information about ACL entry format, see AddAclEntry on page 195.

Syntax

tag `Dir SetAclEntry` *dbname aclname entryname who level control*

where:

- *dbname* is the directory database instance name. If *dbname* is blank (" "), sets the specified ACL entry in all databases.
- *aclname* is the name of the access control list.
- *entryname* is the name of the item to add to the ACL.
- *who* gives the users who are allowed to do what.
- *level* specifies what those users are allowed to do.
- *control* governs program flow and is usually " " (the null string).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
83 Dir SetAclEntry "" password admin
   "(domain=.*\.=eg\.com)(group.base=cn=admin,dc=eg,dc=.com)" write continue
83 OK Completed

84 Dir SetAclEntry "" password local "(domain=.*\.=eg\.com)(self)" write ""
84 OK Completed

85 Dir SetAclEntry "" password self "(self)" read ""
85 OK Completed

86 Dir SetAclEntry "" password others "(users)" compare ""
86 OK Completed
```

```
87 Dir SetAclEntry "" password anonymous "(anonymous)" auth ""
87 OK Completed
```

16

The DI Command

The `DI` command lets you create, delete, and view distribution lists and do other list queries. A distribution list is a collection of email addresses, which may be local user names, remote addresses, or other local distribution lists.

Mail addressed to a distribution list is sent to all members of the list. If an email address matches both a distribution list and a user name, then mail will not be sent to the user unless the user is included in the distribution list expansion.

Subcommands

Add

Creates a new, empty distribution list (DL)

The number of DLs, and the number of entries in a DL, are stored in a database and limited primarily by system resources, especially disk storage.

Syntax

```
tag DI Add Distlist
```

where *`Distlist`* is the name of the distribution list you want to create. `Distlist` follows the same naming rules as for SMTP addresses (see Chapter 56, The Smtp Command) and must not be longer than 64 characters.

Privilege Levels

- ◆ Administrator
- ◆ Helpdesk administrator
- ◆ Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
2 DI Add junk
```

```
2 OK Completed
```

Count

Responds with the number of distribution lists on the system.

Syntax

```
tag DI Count pattern
```

where *pattern* is currently ignored.

Privilege Levels

- ◆ Administrator
- ◆ Helpdesk administrator
- ◆ Domain administrator
- ◆ Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
3 DI Count ""
* 3 6
3 OK Completed
```

Delete

Deletes the specified distribution list.

Syntax

```
tag DI Delete Distlist
```

where *Distlist* is the distribution list you want to delete.

Privilege Levels

- ◆ Administrator
- ◆ Helpdesk administrator
- ◆ Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
8 DI Delete junk
8 OK Completed
```

Importsendmail

Imports a sendmail-style list of email aliases (the format commonly used in the `/etc/aliases` file on UNIX systems) as distribution lists.

This command replaces *all* existing distribution lists except the system distribution lists documented in the *Administrator's Guide*.

`Importsendmail` does *not* support, and silently ignores, the following in the sendmail-style input:

- Program execution (pipe '`|`' syntax)
- Delivery to files
- '`include`' references
- Malformed email addresses

Syntax

```
tag DI Importsendmail aliases
```

where *`aliases`* is a literal string (see Literal Strings on page 48) consisting of a sendmail-style list of email aliases that you want to import as distribution lists.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
2 DI Importsendmail {212+}
a1:
    fred,
    a2,
    a3,
    u1,
    u10,
    u1001
a3:
    a1,
    a2,
    a3,
    a4,
    fred,
    u1,
    u10,
    u1001
refuse:
    Administrator
betty:
    u1,
    u10,
    u100
fred:
    u100,
    u1000,
    u101,
    u102,
    u103,
    u104

2 OK Completed
```

List

Responds with a list of distribution lists. Each line in the response contains the code letter 'L' indicating that the following name identifies a list.

Syntax

`tag DI List pattern start count`

where:

- *pattern* is currently ignored.
- *start* indicates the first distribution list you want to see. The empty string ("") implicitly means 0.
- *count* indicates the number of distribution lists you want to see. See the Example below. The empty string ("") implicitly means all distribution lists. If *count* is greater than the total number of distribution lists, *list* returns as many distribution lists as possible.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain. Delegated domains are so identified in the special MAILER-DAEMON@*deldom* address.

Example

```
5 Dl List "" "" ""
* 5 L abuse
* 5 L backup-alerts
* 5 L backup-status
* 5 L daily-reports
* 5 L mailer-daemon
* 5 L nobody
* 5 L operator
* 5 L postmaster
* 5 L system-alerts
* 5 L weekly-reports
5 OK Completed
```

Membership

Displays the distribution lists to which the current user, or specified user, belongs. Indirect distribution lists (supersets of DL membership) do not appear. To display members of a distribution list, run the `Dlentry List` command.

Syntax

tag `Dl Membership [`*username*`]`

To see DL membership for other users, Administrators, Helpdesk, and Backup may specify a *username*. Domain administrators must specify current domain users only. Regular users do not specify a *username*, but see only their own DL membership. In the protocol, regular users must specify null string (" ") for *username*.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator, in a domain
- Backup operator
- User, but only for that user

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
4 Dl Membership ""
* 4 D abuse
* 4 D daily-reports
* 4 D operator
* 4 D postmaster
* 4 D system-alerts
* 4 D weekly-reports
4 OK Completed
```

```
5 DI Membership postmaster
* 5 D mailer-daemon
5 OK Completed
```

17

The Dlentry Command

The `Dlentry` command lets you view the contents of distribution lists, add and delete entries from them, and do other distribution list queries.

Subcommands

Add

Adds a user or another distribution list to a distribution list.

Syntax

```
tag Dlentry Add distlist member
```

where:

- *`distlist`* is the name of the distribution list to which you want to add.
- *`member`* is the name of the user or distribution list you want to add to *`distlist`*.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
4 Dlentry Add junk stuff
4 OK Completed
```

Addbulk

Adds a list of users or distribution lists to a distribution list.

Syntax

tag Dlentry Addbulk *distlist members*

where:

- *distlist* is the name of the distribution list to which you want to add.
- *members* is a quoted space-separated list or a literal string (see Literal Strings on page 48) containing users or other distribution lists that you want to add to *distlist*. There is a limit on the size of an added *members* list, so if you have more than 2000 addresses, use a new Dlentry Addbulk command.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
3 Dlentry Addbulk junk {35+}
fred
ralph
julie
steph
xerxes

3 OK Completed
```

Count

Responds with the number of entries in the specified distribution list.

Syntax

tag Dlentry Count *distlist pattern*

where:

- *distlist* is the distribution list for which you want a member count.
- *pattern* is currently ignored.

Privilege Levels

- Administrator
- Helpdesk administrator

- Domain administrator
- Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
5 Dlentry Count junk ""
* 5 6
5 OK Completed
```

Delete

Deletes a member from a distribution list.

Syntax

tag `Dlentry Delete` *distlist member*

where:

- *distlist* is the distribution list from which you want to delete a member.
- *member* is the member you want to delete from *distlist*.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
6 Dlentry Delete junk stuff
6 OK Completed
```

List

Responds with a list of entries in a distribution list. Entries in the response that identify distribution lists are prefixed with the letter 'L'.

Syntax

tag `Dlentry List` *distlist pattern start count*

where:

- *distlist* is the name of the distribution list.
- *pattern* is currently ignored.
- *start* indicates the first list entry you want to see. The empty string ("") implicitly means 0.
- *count* indicates the number of entries you want to see. The empty string ("") implicitly means all entries. If *count* is greater than the total number of lists or members, List returns as many entries as possible.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
7 Dlentry List junk "" "" ""
* 7   fred
* 7   julie
* 7   ralph
* 7   steph
* 7 L stuff
* 7   xerxes
7 OK Completed
```

18

The Dns Command

The `Dns` command gets and sets the Mirapoint system's DNS domain name, adds hosts to and deletes them from the list of DNS name servers that the system checks, flushes the DNS cache, and helps diagnose DNS problems.

Here is an overview of DNS: http://en.wikipedia.org/wiki/Domain_name_system

Name Servers

A DNS (Domain Name Service) **name server** is a host that responds to DNS queries. Mirapoint systems maintain a list of name servers to which they issue DNS queries. If the first name server in the list fails to respond with the requested information, the system queries the next name server in the list. The system continues down the list until a name server responds with the requested information.

Follow these steps to change the primary name server (the first server in the list):

1. `Dns Delete` all name servers except the primary.
2. `Dns Add` the new primary name server, making it the second server in the list.
3. `Dns Delete` the old primary name server, promoting your new name server to be the primary name server.
4. `Dns Add` again any name servers that you deleted in step 1.

Subcommands

Add

Adds a DNS name server to the end of your Mirapoint system's list of DNS servers.

SMTP transactions that are pending when you issue this command are aborted and retried later. This is why you should add DNS servers only if email traffic is light.

Syntax

```
tag Dns Add nameserver
```

where *`nameserver`* is the IP address of the name server you want to add.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Dns Add 192.168.0.2
2 OK Completed
```

Count

Counts the number of DNS name servers that the Mirapoint system can query.

Syntax

```
tag Dns Count pattern
```

where *pattern* is currently ignored.

Privilege Levels

- ◆ Administrator
- ◆ Backup operator

Domain Sensitivity

None

Example

```
3 Dns Count ""
* 3 2
3 OK Completed
```

Delete

Removes the specified DNS name server from the list of name servers that the Mirapoint system can query.

SMTP transactions that are pending when you issue this command may be aborted and retried later. It is best to delete DNS servers only when email traffic is light.

Syntax

```
tag Dns Delete nameserver
```

where *nameserver* is the DNS domain server that you want to delete—you must specify the name server exactly as it appears in the list returned by Dns `List`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
7 Dns Delete 192.168.0.2
7 OK Completed
```

Flushcache

Stops and restarts the naming service, which flushes the DNS cache without reboot.

Syntax

tag `Dns Flushcache`

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Dns Flushcache
4 OK Completed
```

Get

Retrieves the DNS domain name where the Mirapoint system resides.

Syntax

tag `Dns Get` *parameter*

where *parameter* must currently be `Domain`, referring to the fully qualified domain name of the Mirapoint system, including host name.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
4 Dns Get Domain
* 4 host.example.com
```

List

Displays a list of DNS name servers that the Mirapoint system can query.

Syntax

```
tag Dns List pattern start count
```

where *pattern*, *start*, and *count* are currently ignored.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
5 Dns List "" "" ""
* 5 192.168.0.1
* 5 192.168.0.2
5 OK Completed
```

Lookup

Issues a DNS query. You can use this command to diagnose problems with your DNS server configuration. Queries time out if no response is received from the server within 15 seconds.

Syntax

```
tag Dns Lookup name options
```

where:

- *name* is the name for which you want to search in the DNS database
- *options* is a quoted, space-separated list of the following name-value pairs:
  - `type=`*type*

  where *type* is one of the following DNS record types:

– A
– ANY
– CNAME
– MX
– NS
– PTR
– SOA
– TXT

If you do not specify this option, its value defaults to ANY. For details about these record types, consult a DNS reference book, such as *DNS and BIND*, published by O'Reilly and Associates, Inc.

❖ server=*host*

where *host* is the name of the DNS server that you want to query. If you do not specify this option, the query defaults to your DNS server list (see Name Servers on page 211).

❖ recurse=*rec*

where *rec* is one of:

– true—query DNS records recursively (this is the default)
– false—do not query DNS records recursively

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Dns Lookup example.com ""
* 6 NS NS.ISI.EDU
* 6 NS VENERA.ISI.EDU
6 OK Completed

8 DNS Lookup hostname.example.com ""
8 NO Timed out
```

Set

Sets the DNS domain name where the Mirapoint system currently resides.

This command can also be used to (re)set the hostname.

Syntax

tag Dns Set *parameter value*

where:

- *parameter* must currently be `Domain`, referring to the fully qualified domain name of the Mirapoint system, including host name.
- *value* is the value you want to assign to *parameter*.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
9 Dns Set Domain host.example.com
9 OK Completed
```

The Domain Command

The `Domain` command lets you manage delegated domains on the Mirapoint system. If so licensed, a domain administrator can set catch-all and domain signature.

Delegated domains can be created and managed by the top-level administrator, but in-domain administration requires a Delegated Domain Administration license.

Delegated Domains

A **delegated domain** is a DNS mail domain hosted on the Mirapoint system that has its own namespaces for:

- Mailboxes
- Users accounts
- Distribution lists
- Administrators

In a delegated domain, the (plural) `administrators` ACL stands for all users with Administrator privilege.

Domain Sensitivity and the Current Domain

When you use the `Domain Setcurrent` command, specifying a valid delegated domain name, that domain becomes the **current domain**. Many administration protocol commands behave differently in this state than when no delegated domain is current. When you use `Domain Setcurrent` with an empty domain name (`""`), there is no current delegated domain, and all commands apply to the system's primary domain (the domain part of the output of `Dns Get Domain`).

For example, if you issue the command `Mailbox Add user.george` while the delegated domain `example.com` is current, you create a mailbox that is addressable as `george@example.com`. If, however, you issue the same command when no delegated domain is current, you create a mailbox in your system's primary DNS domain.

Subcommands

Add

Creates a delegated domain.

Syntax

```
tag Domain Add domain-name
```

where *domain-name* is the fully qualified name of the delegated domain you want to add, such as `example.com`. Delegated domains follow the same naming rules as DNS domains:

- Must have two or more (but fewer than 64) dot-separated segments
- Characters can be letters, numbers, underscore (_), or hyphen (-)
- May not start or end with underscore or hyphen
- One character must be non-numeric.

Privilege Levels

Administrator

Domain Sensitivity

This command is allowed only when no delegated domain is current.

Example

```
2 Domain Add example.com
2 OK Completed
```

Count

Responds with the current number of delegated domains.

Syntax

```
tag Domain Count pattern
```

where *pattern* is one of:

- "" (empty string)—equivalent to the wildcard character `*`, meaning all domain names.
- *value*—a pattern string, optionally containing wildcard characters, matching the domain names you want to count (see Using Patterns on page 49).

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

This command is allowed only when no delegated domain is current.

Example

```
2 Domain Count ""
* 2 2
2 OK Completed
```

Delete

Deletes the specified delegated domain. All mailboxes, email messages, user accounts, distribution lists, and configuration data belonging to the domain are destroyed.

Syntax

```
tag Domain Delete domain-name
```

where *domain-name* is the name of the domain you want to delete, such as `example.com`.

Privilege Levels

Administrator

Domain Sensitivity

This command is allowed only when no delegated domain is current.

Example

```
3 Domain Delete example.com
3 OK Completed
```

Get

Responds with the value of the specified parameter for the specified domain.

Syntax

```
tag Domain Get parameter domain-name
```

where:

- *parameter* is one of the domain settings documented under `Domain Set`.
- *domain-name* is the name of the domain to which *parameter* applies (or for the `Signature` parameter only, the special keyword `Primary`, meaning the system's top-level domain).

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator (`Domain Get Catchall` only)
- Backup operator

Domain Sensitivity

When no delegated domain is current, you may specify any delegated domain when issuing this command. When a delegated domain is current, you must specify the current domain.

Example

```
4 Domain Get Autoreplies example.com
* 4 OFF
4 OK Completed
```

Getcurrent

Responds with the name of current delegated domain (see Domain Sensitivity and the Current Domain on page 217).

Syntax

```
tag Domain Getcurrent
```

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

None

Example

```
5 Domain Getcurrent
* 5 example.com
5 OK Completed
```

List

Responds with a list of delegated domains.

Syntax

tag `Domain List` *pattern start count*

where:

- *pattern* is one of:
    - "" (empty string)—equivalent to the wildcard character `*`, meaning all domain names.
    - *value*—a pattern string, optionally containing wildcard character `*`, matching the domain names to list (see Using Patterns on page 49).
- *start* is the number of the first domain you want to see. The empty string (`""`) implicitly means `0`.
- *count* is the number of domains you want to see. The empty string (`""`) implicitly means all domains. If *count* is greater than the total number of domains, `List` returns as many domains as possible.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

This command is allowed only when no delegated domain is current.

Example

```
6 Domain List "" "" ""
* 6 beispiel.de
* 6 example.com
6 OK Completed
```

Set

Sets the value of the specified parameter for the specified domain.

Syntax

tag `Domain Set` *parameter domain-name value*

where:

- *parameter* is one of:
  - Autoreplies—specifies whether users in the domain can enable automatic replies for their mailboxes. The value is either ON or OFF. The default value is ON.
  - Calendar.maxnumevents—the maximum number of events allowed in any calendar. Zero (0) means to accept the default of 2000.
  - Calendar.removeafter—the number of days after which to remove expired events whose date has passed. Zero (0) means no limit, no removal. For repeating events, expiration occurs on the day of the last event instance. Events that repeat forever never expire.
  - Calendar.timeout—specifies the calendar timeout in minutes, used by WebCal, WAP, and XML interfaces.
  - Catchall—an email address to which undeliverable messages addressed to the domain should be sent. If you create a catchall, messages addressed to non-existent users in this domain get diverted to the address specified by *value*. Otherwise, these messages bounce. The default is the null string (“ ”) meaning no catchall address.
  - Diskquota—the maximum disk space in kilobytes that the domain may use, ranging from 0 (zero space allowed) on up. The default is no quota (quota root does not exist), implying unlimited disk space. To restore the default, set the disk quota to -1 (minus one). In the output of a Domain Get command, the kilobytes of disk space in use appears after a dot (domain’s parent mailbox) followed by the current disk quota.
  - Dls—specifies whether administrators of the domain can create and add users to distribution lists. The value is either ON (the default) or OFF.
  - Fwds—specifies whether domain users can enable automatic forwarding for their mailboxes. The value is either ON (the default) or OFF.
  - Signature—the “domain signature” content (given in 7-bit ASCII) that most mailers automatically append to the body of messages being sent from this domain. It follows any user-provided signature. Currently limited to 1024 bytes. The domain signature might appear to the email recipient as an attachment, depending on how the mailer handles multipart MIME messages. The value is a literal string of the following format, where *Content* is the verbiage that mail-user-agents should append to outgoing messages:

    ```
    Content-type: text/plain
    Content
    ```

  - Userquota—the maximum number of users allowed in the domain. The value must be a non-negative integer. Specifying a value of 0 allows the domain to contain an unlimited number of users. If you do not set this parameter, the number of domain users defaults to 20.
- *domain-name* is the name of the domain to which *parameter* applies (or for the Signature parameter only, the special keyword primary, meaning the system’s top-level domain).
- *value* is the value you want to assign to *parameter*.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator (`Catchall` and `Signature` only)

Domain Sensitivity

When no delegated domain is current, you may specify any delegated domain when issuing this command. When a delegated domain is current, you must specify the current domain.

Example

```
7 Domain Set Autoreplies example.com ON
7 OK Completed

8 Domain Set Calendar.maxnumevents example.com 2400
8 OK Completed
```

Setcurrent

Sets the current delegated domain. This affects the behavior of many commands, as described in Domain Sensitivity and the Current Domain on page 217. If you specify the empty string (`""`) for the domain name, the current domain is unset, leaving no domain current.

Syntax

tag `Domain Setcurrent` *domain-name*

where *domain-name* is the name of the delegated domain you want to make current. If you specify the empty string (`""`) for the domain name, the current domain is unset, leaving no domain current.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

None

Example

```
9 Domain Setcurrent ""
9 OK Completed
```

The Exception Command

The `Exception` command provides a way to alter functionality of various services, primarily security settings of inbound SMTP.

One common use for the `Exception` command is to prevent email spoofing. The following commands force SMTP connections to authenticate if they claim to be from example.com or any of its subdomains:

```
Exception Add smtp:in smtpauth (sender=*@example.com) required
Exception Add smtp:in smtpauth (sender=*@*.example.com) required
```

Combined with `Relay Add` for local IP addresses and trusted mail servers, this forces all incoming SMTP connections from the external Internet to authenticate. So if email falsely claims that its sender is @example.com, in this case it gets rejected with a "530 MAIL requires AUTH" error. Local users would be asked to authenticate (supply a password) if they have not already done so using POP or IMAP.

The `Exception` commands above are recommended for educational institutions and service providers that cannot trust every member of their online community. For organizations that trust their users, the authentication step (password for sending) can be avoided by disabling `Smtpauth` when the originating IP address is inside the local network, in this case 10.7.0.0 with netmask 255.255.0.0.

```
Exception Add smtp:in smtpauth (sender=*@example.com) required
Exception Add smtp:in smtpauth (sender=*@*.example.com) required
Exception Add smtp:in smtpauth (domain=10.7.0.0/16)(sender=*@example.com) off
Exception Add smtp:in smtpauth (domain=10.7.0.0/16)(sender=*@*.example.com) off
```

Subcommands

Add

Adds an exception to the system list. Currently supports only inbound SMTP.

Exceptions are limited to 1000 per protocol-attribute pair.

Syntax

`tag` `Exception Add` *`protocol attribute domain value`*

where:

- *protocol* is the name of a protocol on which to set an exception. This may include modes specific to the protocol, for example `smtp:in`.
- *attribute* is the protocol setting that is being changed from its default.
- *domain*, as used here, specifies a source of incoming email (`smtp:in`) to which the exception applies. *Domain* may be a hostname, an IP address, a DNS domain name, or a network and netmask pair. Or *domain* may be a parenthesized list "`(domain=`*domname*`)(constraint=`*type*`)`" where constraint is specific to a protocol-attribute pair, and may include conditions other than IP address or hostname. `Exception Add` normalizes the domain name. Only one constraint of a given type is allowed; if you need multiple constraints, enter them as separate exceptions. If no domain is specified for a constraint, then it is always applied, if there are no matching constraints of this type for the protocol-attribute pair. The order of lookup precedence is:
  - IP address—exact match (implicit or explicit `/32` netmask)
  - hostname—exact match (no wildcard allowed)
  - IP address—most specific netmask (netmask `/24` wins over `/16`)
  - *.domainname—longest name wins (wildcard "*" refers to a subdomain)
  - Bare constraint—a constraint is found without an associated domain, and no domain name or network matched with this constraint specified.
  - Rules with larger numbers of constraints have higher precedence, and rules that are identical except for constraints are sorted by constraint names.
- *value* is the exception setting of this protocol for the specified domain.

Only one domain can apply per protocol for a given domain name or IP address. Thus if there is an `smtp:in` security attribute for `*.example.com`, and an `smtp:in smtpauth` attribute for `desktop.example.com`, and a connection is made from `desktop.example.com`, then **only** the `smtpauth` attribute will match. To inherit a default subdomain attribute, you must make a copy of that attribute in the more specific domain entry.

The following table shows allowed protocols, attributes, values, and constraints. An *emailPattern* is an email address allowing * wildcard. Listener is as specified under `Smtp Addlistener`.

Table 2 Allowed Arguments for Exception Add

Protocol	Attribute	Value	Constraints
smtp:in	maxmsg	0 <= *Max* <= 128 M	listener=*IPaddr:port*
smtp:in	rbl	on, off	—

smtp:in	security	secureauthonly, ssl	sender=*emailPattern*
			listener=*IPaddr:port*

smtp:in	sendercheck	on, off	sender=*emailPattern*
smtp:in	senderisauth	on, off	sender=*emailPattern*
smtp:in	senderisvalidrecipient	on, off, reject	sender=*emailPattern*

smtp:in	smtpauth	on, off, norelays, required	sender=*emailPattern*
			listener=*IPaddr:port*

Privilege Levels

Administrator

Domain Sensitivity

Can be applied only in the top-level domain, but affects all domains.

Example

These commands say that all incoming SMTP connections from network 192.168 require SMTP authentication, and behave as if `Smtp Set Security Secureauthonly` had been run. The third and fourth are equivalent to the first and second. For more examples, see `Exception List`.

```
1 Exception Add smtp:in Smtpauth 192.168.0.0/16 Required
1 OK Completed
2 Exception Add smtp:in Security 192.168.0.0/16 Secureauthonly
2 OK Completed

3 Exception Add smtp:in Smtpauth (domain=192.168.0.0/16) Required
3 OK Completed
4 Exception Add smtp:in Security (domain=192.168.0.0/16) Secureauthonly
4 OK Completed
```

Count

Counts the number of exception that have been added to the system list.

Syntax

tag `Exception Count` *protocol attribute pattern*

where:

- *protocol* is the name of a protocol to which an exception was added.
- *attribute* is the protocol setting that was changed from its default.
- *pattern* may be null string (“ “) or asterisk (*) to match all exceptions.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

Can be applied only in the top-level domain, but affects all domains.

Example

```
3 Exception Count smtp:in security
* 3 1
3 OK Completed
```

Countattrs

Counts all supported attributes for a given protocol.

Syntax

tag `Exception Countattrs` *protocol pattern*

where:

- ◆ *protocol* is the name of a protocol with exception; see `Exception Add`.
- ◆ *pattern* may be the null string “ “ to match all protocol attributes.

Privilege Levels

- ◆ Administrator
- ◆ Backup operator

Domain Sensitivity

Can be applied only in the top-level domain, but affects all domains.

Example

```
7 Exception Countattrs smtp:in
* 7 6
7 OK Completed
```

Countprotocols

Counts supported protocols.

Syntax

tag `Exception Countprotocols` *pattern*

where *pattern* may be the null string “ “ to match all protocols.

Privilege Levels

- ◆ Administrator
- ◆ Backup operator

Domain Sensitivity

Can be applied only in the top-level domain, but affects all domains.

Example

```
8 Exception Countprotocols
* 8 1
8 OK Completed
```

Delete

Deletes an exception to the system list.

Syntax

```
tag Exception Delete protocol attribute domain
```

where:

- *protocol* is the name of a protocol from which to delete an exception.
- *attribute* is the protocol setting that was changed from its default.
- *domain* is an IP address, hostname, or (wildcard) domain; see `Exception Add`.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (for their remote mail)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
6 Exception Delete smtp:in security *.example.com
6 OK Completed
```

List

Lists exceptions that have been added to the system list. Output of this command makes valid `Exception Add` and `Delete` parameters.

Syntax

```
tag Exception List protocol attribute pattern start count
```

where:

- *protocol* is the name of a protocol to which an exception was added.
- *attribute* is the protocol setting that was changed from its default.
- *pattern* may be null string (`""`) or asterisk (`*`) to match all exceptions; *start* and *count* are numbers of the first and extent of the listings.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

Can be applied only in the top-level domain, but affects all domains.

Example

```
1 Exception Add smtp:in security (domain=10.0.0.0/24) ""
1 OK Completed
2 Exception Add smtp:in security (domain=*.example.com)(sender=alerts@*.example.com) ssl
2 OK Completed
3 Exception Add smtp:in security (domain=*.example.com) "ssl secureauthonly"
3 OK Completed
4 Exception Add smtp:in security (sender=alerts@*) "ssl secureauthonly"
4 OK Completed
5 Exception List smtp:in security "" "" ""
* 5 "(domain=10.0.0.0/24)" ""
* 5 "(domain=*.example.com)(sender=alerts@*.example.com)" ssl
* 5 "(domain=*.example.com)" "ssl secureauthonly"
* 5 “(sender=alerts@*)" "ssl secureauthonly"
5 OK Completed
```

Listattrs

Lists all supported attributes for a given protocol.

Syntax

tag `Exception Listattrs` *protocol pattern start count*

where:

- *protocol* is the name of a protocol with exception; see `Exception Add`.
- *pattern* may be the null string ("") to match all protocol attributes; *start* and *count* specify beginning and ending item.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

Can be applied only in the top-level domain, but affects all domains.

Example

```
5 Exception Listattrs smtp:in
* 5 rbl
* 5 security
* 5 sendercheck
* 5 senderisauth
* 5 senderisvalidrecipient
* 5 smtpauth
5 OK Completed
```

Listprotocols

Lists all supported protocols.

Syntax

tag `Exception Listprotocols` *pattern start count*

where *pattern* may be the null string (`""`) to match all protocols; *start* and *count* specify beginning and ending item.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

Can be applied only in the top-level domain, but affects all domains.

Example

```
6 Exception Listprotocols
* 6 smtp:in
6 OK Completed
```

Test

Tests system exceptions with optional constraint. Data returned from this command has two elements: an attribute and its value.

Syntax

tag `Exception Test` *domain IPaddress protocol attribute constraint*

where:

- *domain* is DNS-style domain name; if missing, IP address is used instead.
- *IPaddress* is a numeric Internet address; if missing, domain is used instead.
- *protocol* is the name of a protocol with exception; see `Exception Add`.
- *attribute* is the protocol setting that was changed from its default (optional). If empty (`""`) then all attributes matching the protocol are displayed.
- *constraint* currently may be (`sender=`*username*)`@`*domain*; see `Exception Add`. If empty (“ “) then this command tests for unconstrained exceptions.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

Can be applied only in the top-level domain, but affects all domains.

Example

```
9 Exception Test example.com 10.20.30.4 smtp:in "" sender=alerts@example.com
* 9 "ssl secureauthonly"
```

The Failover Command

The `Failover` command lets you shut down and reboot the active or standby unit of a failover-capable system. It also lets you set the monitored failover port.

Failover requires a license. Mirapoint supports failover for a single standby and one or more active units. On all units, failover software coordinates active and standby roles, monitors system heartbeats, and responds to requests from other units. Active and standby systems communicate using a disk partition where the active system has write permission while the standby system has only read permission.

Failover systems monitor port0 by default and sending an ARP request to the router when running network checks. The `Addport` command modifies this default by adding a port IP address for each network check. Ports are not monitored unless bound. Changing port configuration triggers a check that the monitoring address is still reachable. If it is not, an Alert is logged.

Subcommands

Addport

Adds a port to the list monitored by the failover system. If any of the monitored ports experience a failure on the active unit, but are working on the standby unit, the system will fail over.

Syntax

```
tag Failover Addport portnum monaddr
```

where:

- *`portnum`*—Ethernet port to be monitored for failover, as labeled on the system back panel. Possible values are `port0`, `port1`, `port2`, and so on. If the *`portnum`* already exists, this command sets its *`monaddr`* to the new value.
- *`monaddr`*—IP address to which the system should send an ARP request when checking for a network failure on the port. Specifying null string (“ “) sets value to the default router. This command fails if *`monaddr`* is unreachable.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

Adding Gigabit Ethernet port monitored on IP address 10.0.0.5 (private network):

```
4 Failover Addport port1 10.0.0.5
4 OK Completed
```

Countport

Displays the number of monitored ports.

Syntax

```
tag Failover Countport pattern
```

where *pattern* must be * or the null string ("") to match all ports.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Failover Countport *
* 5 2
5 OK Completed
```

Deleteport

Removes a port from the list monitored by the failover system.

Syntax

```
tag Failover Deleteport portnum
```

where *portnum* is a monitored Ethernet port, as in `Failover Addport`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
7 Failover Deleteport port0
7 OK Completed
```

```
8 Failover Listport * "" ""
* 8 port1 10.0.0.5
8 OK Completed
```

Get

Returns the current failover timeout value.

Syntax

tag `Failover Get` *parameter*

where *parameter* must be `Timeout`. The default is 30 seconds.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
9 Failover Get timeout
* 9 30
9 OK Completed
```

Listport

Displays the monitored failover ports and their address.

Syntax

tag `Failover Listport` *pattern start count*

where *pattern* must be * or the null string ("") to match all ports, *start* is the beginning position, and *count* is the number to display.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Failover Listport * "" ""
* 6 port0 ""
* 6 port1 10.0.0.5
6 OK Completed
```

Reboot

After a one-minute delay, allowing you to log out, this command reboots the specified failover unit. If you reboot the active unit, the standby unit detects this and becomes the active unit.

Syntax

tag `Failover Reboot` *unit*

where *unit* is one of:

- `Active`—the current active unit in the failover system.
- `Standby`—the current standby unit in the failover system.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Failover Reboot Standby
2 NO This model does not support failover
```

Set

Changes the failover timeout value. Gives an error if run on an active node. May be run on the standby node, or on a standalone system being prepared as the standby. This restriction guarantees that all nodes in a cluster see the new timeout value.

Active nodes in an N+1 failover cluster periodically write a heartbeat message to the shared disk area. The standby node reads each active node's heartbeat to determine if a node has died. Failover uses a system timer that forces takeover if an active node does not reset its heartbeat. When takeover occurs, the standby node reboots with a "Failover Subsystem Error - Missing Heartbeat" error.

N+1 failover gets confused if disk I/O takes a long time, as sometimes occurs during backup. A long delay waiting for I/O can cause unnecessary takeover. An adjustable value between 30 and 300 seconds allows you to increase the timeout on systems where I/O might be delayed. Note: increasing failover timeout increases the delay before the standby node takes over when a real failure occurs. Increase this value only on systems where I/O delays caused by SAN storage cause unwanted failover.

Syntax

```
tag Failover Set parameter value
```

where *parameter* must be `Timeout`, and *value* is a number from 30 to 300 seconds. Gives an error if *value* is outside this range. The default is 30 seconds.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
10 Failover Set timeout 60
10 OK Completed
```

Shutdown

After a one-minute delay, allowing you to log out, this command shuts down the specified failover unit. If you shut down the active unit, the standby unit detects this and becomes the active unit.

Syntax

```
tag Failover Shutdown unit
```

where *unit* is one of:

- `Active`—the current active unit in the failover system.
- `Standby`—the current standby unit in the failover system.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Failover Shutdown Standby
3 NO This model does not support failover
```

22

The Filter Command

The `Filter` command organizes incoming messages, specifying that certain actions, such as redirecting to a folder, be performed on messages that meet certain criteria.

Filters

A **filter** is a set of conditions, known as **rules**, and an associated **action** performed on messages directed to a particular domain or mailbox. Actions may be performed on messages for which any or all conditions are true. Each domain or mailbox can have multiple filters, known as a **filterset**, which are evaluated in order for each incoming message. Filtering can be turned off with `Conf Disable`.

Filtering occurs after distribution list (DL) expansion and address normalization have been done. Normalizations include dot-to-underscore mapping, alteration or removal of mail domains, and Unicode UTF interpretation.

You can set up filters either for a user inbox or for a domain. User filtering and domain-wide filtering are similar, so you can test a domain filter on one user first. Exceptions: user filters replace null addresses with `MAILER-DAEMON@`*`hostname`*, and header rewrites occur in user filters with LDAP and when appending `@`*`localhost`*.

Filters are executed in priority order: 100 before antivirus and antispam scanning, then 450 quarantine, and finally 500 for normal domain filters. However, all filters for a given priority are evaluated before any of them are acted on. Thus, the only way for a filter action to affect the evaluation of another domain filter is to assign each one a different priority. All user filters have the same priority, so the action of one user filter cannot affect the evaluation of another one.

Actions

The actions that a filter can take on a message are:

- `Discard`—Discards the message completely.
- `Disclaimer`—Adds a disclaimer to the message.
- `Fileinto`—Inserts the message into the specified mailbox (user filters only).
- `Fwdexcerpt`—Forwards an excerpt of a message to the specified address. For SMS, size of excerpt defaults to 3 lines or 160 bytes, whichever comes first.
- `Keep`—Message is retained until the end of filters at the same priority level. Equivalent to "`Fileinto INBOX`" action.

- `Modspamscore`—Raise or lower spam score in `X-Junkmail-Status` header.
- `Quarantine`—Sends substitute three-part message to the quarantine mailbox showing reason, RFC 2821 envelope, and RFC 822 message. May be used in domain filters to hold for approval by the Quarantine administrator, or in user filters to await approval by Junk Mail Manager (JMM) subscribers.
- `Redirect`—Sends the message to the specified email address or mailhost.
- `Reject` (for domains only)—Rejects the message with a permanent failure, generating a bounce message to the sender.
- `Wiretap` (for domains only)—Copies message to a specified email address, optionally changing the sender field, before delivering to recipient.

If no action results when a filterset is applied to a message, the message is placed in the recipient's inbox; this is called **implicit keep**. When a filterset results in explicit action (except `Fwdexcerpt`) on a message, no implicit keep occurs. To perform multiple actions on messages (for example, both `Keep` and `Fileinto`) you must specify multiple filters for the mailbox using the `Continue` keyword (see `Filter Add` and `Filter Change`). However scope of the pseudo-domain filters (`domain=any`, `local`, or `nonlocal`) does not override the implicit keep action of a domain filter.

The only actions that affect filters at different priorities are those that halt all processing, such as `Discard` or `Quarantine`.

Rules

Each filter has zero or more rules, which determine whether an action is performed on a message. A filter with zero rules and `allof` argument is always true; with `anyof` it is always false. Each rule describes a condition for a particular attribute of incoming messages, such as:

- `Subject contains "make money fast"`
- `From matches "*@spamcity.com"`

The format of each rule is:

`attribute match-type string`

where:

- *`attribute`* is a message header name (case-insensitive) or one of the following keywords, which begin with colon to distinguish them from header names:
  - `:body`—The message body.
  - `:bodydecoded`—The decoded form of MIME text parts of the message, that is, parts with "text/*something*" type.
  - `:bodydecodedbinary`—Decoded form of all MIME parts of the message, both text and various binary types.
  - `:envelopefrom`—Sender address as specified in the message envelope.
  - `:envelopeto`—Recipient address as specified in the message envelope. Does not work for DLs, which are expanded before filtering (use `:tocc`).
  - `:size`—Message size (body and headers) as specified by *`string`*.
  - `:tocc`—The `To` or `Cc` header.

- `:UCE`—Spam-level classification reported by the Antispam facility, used for the “System Junk Mail Rule” (see Chapter 66, The Uce Command for more information). Not allowed in domain filters. See below for match strings. See Fetch on page 251 for an example.
- `:UCEscore`—The junk mail score as a number specified by *`string`*.
- `:attachmentfilename`—The advertised file name of the attachment, from either Content-Type *name* or Content-Disposition *filename*. If these contain invalid characters (according to MIME standards) they are ignored.
- `:attachmenttype`—The Content-Type media type of an attachment (for instance text/plain, image/jpeg, etc.) as defined in RFC 2046. Non-MIME messages are treated as having a single text/plain part.
- `:attachmentsize`—The size in bytes of attachments after being decoded according to `Content-Transfer-Encoding` (e.g. base64, quoted-printable). Only MIME decoding is supported; ZIP or `uucp` files are not decompressed. The values of *`match-type`* are the same as for `:size` (see below).

To select a user it is best to use `Matches` *`Username*`* keywords, or `Patternlist`, to account for rewriting of fully qualified addresses into local addresses.

The simplest way to filter email for a user is with a user filter, which processes mail only for that user, so an `:envelopeto` test is not needed. If an `:envelopeto` test must be done with a domain filter, use `Matches` instead of `Contains` to avoid false hits. In some cases for local users, the `@`*`maildom`* or `@`*`hostname`* part of the address may be removed from the envelope-to address before filtering, so have the filter specification match the user name by itself:

```
filter "BlockUser9" reject "" allof stop
:envelopeto matches "user9"
:envelopefrom contains-not "mail.example.com"
```

With LDAP enabled, `:envelopeto` looks up the fully qualified address, as given by the publishedName. The `:envelopefrom` test looks up the fully-qualified address of user@localhost in some cases. Note that dot-to-underscore mapping and distribution lists might mean that the user name is not what you think it is.

- For all attributes except `:bodydecoded`, `:size`, and `:UCEscore`, *`match-type`* must be one of:

  - `contains`—*`attribute`* contains search pattern *`string`*.
  - `contains-wordlist`—with any *`attribute`* and *`string`* specifying the wordlist name and optionally domain, `(domain=primary)(name=obscene)` for instance. Wordlist comes from the current domain if domain is missing. See Chapter 71, The Wordlist Command for details.
  - `in-ldap-group`—for `:envelope*` attributes that yield an email address, runs an LDAP query to determine if sender or recipient is in the mailgroup specified by the *`string`* following. The `-not` suffix checks nonmembership, while `-casedep` has no effect because mailgroups are case insensitive.
  - `matches`—*`attribute`* matches search pattern *`string`*, which may contain these wildcard characters:
    - *—Matches any sequence of zero or more characters
    - ?—Matches any single character

    For example, `"*money*"` matches all of the following strings:

  - make money fast
  - how to double your money
  - money makes the world go around

  `"xyzzy?6bq5"` matches all of the following strings:

  - xyzzy06bq5
  - xyzzyZ6bq5
  - xyzzy96bq5

- `matches-patternlist`—takes any *`attribute`* and a *`string`* specifying patternlist name and optional domain, `(domain=primary)(name=addrs)` for example. Patternlist comes from the current domain if you do not specify a domain. See Chapter 45, The Patternlist Command for more information.
- `regexmatches`—*`attribute`* matches POSIX 1003.2 regular expression *`string`*. Refer to http://www.dc.turkuamk.fi/docs/gnu/rx/rx_3.html for an introduction to POSIX regular expressions. The *`attribute`* can be any header name or the special keywords `:body` or `:bodydecoded` (see above). Regular expression `^` matches the beginning of line, and `$` matches newline, however email message lines end with carriage-return and newline (`\r\n`) so you must say `"\r$"` to match the end of line.

All match comparisons are case-insensitive. All except `regexmatches` can be make case-sensitive with the `-casedep` modifier (see below). All four match comparisons can be negated with the `-not` modifier. Append these modifiers without intervening whitespace:

- `-not`—Negates match. For example, `contains-not` means "does not contain" and `matches-not *` means the specified field does not exist.
- `-casedep`—Specifies a case-sensitive comparison. For example, `matches-casedep` means "matches considering capitals" distinctions.

For the `:UCE` attribute, *`match-type`* must be either `is` or `is-not`, and *`string`* may be one of the following:

- `off`—Test is always false (junk mail processing off). This is needed because the filter "System Junk Mail Rule" cannot be deleted.
- `normal`—Test is true if the sender is on the black list and the sender is not on the white list, or the message is marked with an `X-Junkmail` header of a domain-specific value and the sender is not on the white list.
- `exclusive`—Test is true if the sender is not on a white list.

For the `:size` and `:UCEscore` attributes, *`match-type`* must be one of:

- `over`—Message size or UCE score is greater than specified by *`string`*.
- `under`—Message size or UCE score is less than specified by *`string`*.

For the `:bodydecodedbinary` attribute, *`string`* must be a Unicode UTF-8 representation of search pattern, with extended format to allow hexadecimal encoding; for example, "\xA2" is the copyright symbol.

For the `:bodydecoded` attribute, *`match-type`* must be one of the following, and *`string`* must be a Unicode UTF-8 representation of the search pattern.

- `includes(compare=nocase)`—The search is case-insensitive and ignores whitespace, both in the search pattern `string` and in the decoded text being searched. Wildcard characters are not allowed.
- `includes(compare=nocase)-not`—Negation of the above; no match.

For the `:size` attribute, *`string`* must be an ASCII integer indicating message size in bytes. Optionally, you can append one of the following unit suffixes:

- K—Indicates a size in kilobytes
- M—Indicates a size in megabytes
- G—Indicates a size in gigabytes

- *`string`* depends on *`attribute`* and *`match-type`*, as described above. To match an empty header line, use `-matches-not "*"` instead of a null *`string`*.

System Junk Mail Rule

This built-in filter rule appears when the Antispam license is applied:

```
filter "System Junk Mail Rule" fileinto "INBOX.Junk Mail" allof stop
:UCE is "off"
```

This rule shows up with the `Filter Fetch` command. By default this filter is off, because POP users lack folders and therefore would never see any false positives directed to their Junk Mail folder. For IMAP and WebMail class of service, where folders are available, you can use the LDAP value `miDefaultJunkmailFilter` and the `Cos` command to enable the UCE filter rule by default, by setting it to `normal`. Individual users can also control their UCE rule from Administration Suite.

```
#@Mirapoint-Filter-1.0
filter "System Junk Mail Rule" fileinto "INBOX.Junk Mail" allof stop
:UCE is "normal"
```

Because this rule has multiple lines, you must base64-encode it for use by LDAP. Here is how it looks in LDIF (two colons indicate binary data):

```
miDefaultJunkmailFilter:: I0BNaXJhcG9pbnQtRmlsdGVyLTEuMApmaWx0ZXIgIlN5
 c3RlbSBKdW5rIE1haWwgUnVsZSIgZmlsZWludG8gIklOQk9YLkp1bmsgTWFpbCIgYWxsb
 2Ygc3RvcAo6VUNFIGlzICJub3JtYWwiCg==
```

You can break the base64-encoded string into as many lines as you like, with one single space at the beginning of each line other than the first. Please note that the string you encode must have a blank line at the end of it (two successive newlines). You can test this by using the `Filter Import` command on the text you plan to base64-encode.

Filtering to Reduce DSN in Queue

Typical mail server queues are clogged with many Delivery Service Notification (DSN) messages, which are often generated in response to junk mail that cannot be delivered because it was sent to non-existent (obsolete or fictitious) addresses. It is now possible to filter these DSN messages out of the queue, with destination domain set to `Any` or `Nonlocal`. The Administration GUI provides menu choices for three X-DSN headers (filter attributes) including two in the CLI example below:

```
Filter Add (domain=any) discard-junk-DSN Discard "" anyof stop
X-DSN-Junkmail contains "UCE"
```

```
X-DSN-Mirapoint-Virus contains VIRUSFOUND
```

You might also want to filter and discard using header `X-DSN-Junkmail-Status` if users filter and reject `Junkmail-Status`, which includes UCE score and reason.

Note that these filtering techniques only work if the DSN messages are generated by a Mirapoint server. Non-Mirapoint servers will not generate the required X-DSN-Junkmail headers. For example, your system could be configured so that the Mirapoint server doesn't check recipients and routes mail to an internal non-Mirapoint messaging server (such as one running Microsoft Exchange), which will generate the DSN notification. In that case, you can use the SMTP parameter `dsnretrycnt`, which will reduce the time spent on processing invalid outbound DSN messages. For example, you could set the parameter to 10, and then tune it down as appropriate (the default value is 0, which means an unlimited number of retries).

Subcommands

Add

Adds a filter to the end of the list of filters that are applied to messages being delivered to the specified domain or mailbox.

There is a per-user and per-domain limit of 200 message filters.

Syntax

tag `Filter Add` *scope fname action arg any-all stop-go {bytes+}*
rules

where:

- *scope* is one of:
  - `"(domain=`*dname*`)"` where *dname* is one of the following, applied in order:
    - The keyword "`Any`" indicates both local and nonlocal SMTP messages.
    - The keyword "`Local`" indicates SMTP messages for users on this host.
    - The keyword "`Nonlocal`" indicates SMTP messages going off-box.
    - The keyword "`Primary`" for local messages to the top-level domain.
    - The name of a delegated domain for local messages to that domain.
    - Null domain name "`(domain=)`" indicating the current domain.
  - `"(mailbox=`*mbox*`)"` where *mbox* is the name of a mailbox.
  - Name of the mailbox; this is equivalent to `"(mailbox=`*mbox*`)"`.
  - `"(priority=`*prio*`)"` where *prio* is a numerical priority. In conjunction with `domain=`*dname* only, determines when the current filter gets executed. If you do not specify *prio*, 500 is the default. Priorities are: 100 to filter before antivirus and antispam scanning, 450 for quarantine filters, and 500 to filter after scanning and domain signature (this is when domain filters are normally executed). You must explicitly specify 450 for quarantine filters.

- `fname` is a name for the filter.
- `action` is one of:
  - `Discard`—Discards the message completely.
  - `Disclaimer`—Adds a disclaimer, as set by the `Message Set` facility.
  - `Fileinto`—Inserts the message into the mailbox specified in *`arg`*.
  - `Fwdexcerpt`—Forwards a message excerpt as specified by *`arg`*. Rewrites the envelope (but not header) `From` field to the forwarding user, so that any bounces go to the excerpt forwarder, not the original sender.
  - `Keep`—Equivalent to "`Fileinto INBOX`".
  - `Modspamscore`—Raise or lower spam score in the `X-Junkmail-Status` header by some number in the hundreds. *`Value`* should be a signed integer beteween -1000 and 1000, added to the original Antispam scan score in the `X-Junkmail-Status` line. Disposition of the message is reevaluated and its header retagged unless it was whitelisted or blacklisted.
  - `Quarantine`—Sends to the specified mailbox a substitute message, which contains 3 MIME sub-parts: Reason, RFC 2821 envelope showing original `MAIL FROM` and `RCPT TO` fields, then the original message in RFC 822 format. Specify priority 450 and perhaps `domain=Any` for quarantine domain filters. The action associated with a user Quarantine JMM filter must be Stop, and the argument should be a single mailbox name (with no period) residing under the `Quarantine.`*`user`* heirarchary (e.g. `Quarantine.juser.held`). These mailboxes are autocreated by the quarantine filter.
  - `Redirect`—Sends the message to the email address or mailhost specified in *`arg`*. But see note about pseudo-domain filters under `stop-go` below.
  - `Reject` (for domains only)—Rejects message with a permanent failure. As of release 3.4, locally generates a bounce message to the original sender, which could cause undeliverable messages to accumulate.
  - `Wiretap` (for domains only)—Copies the message to a specified address, optionally changing the sender, before delivering to the recipient.

All actions (except `Fwdexcerpt`, `Modspamscore`, and `Wiretap`) may include `"(notify=`*`Facility`*`)"` in *`arg`* to enable message notification for domain level filters, where *`Facility`* is some user-definable facility in the `FILTER.NOTIFY.*` name space. To disable message notification, specify `"(notify=)"` in the *`arg`*. An extra argument `"(notify_from=`*`Sender`*`)"` or `"(notify_to=`*`Recipients`*`)"` may be appended. Message notification is disabled by default.

All actions may include `"(extraheader=`*`Field`*`: `*`Value`*`)"` in *`arg`* to insert some arbitrary "*`Field`*`: `*`Value`*" header line. Header must be displayable 7-bit ASCII, no more than 128 bytes long, and may not contain parentheses or line breaks.

For the `Disclaimer` action, *`arg`* must include `"(message=`*`Facility`*`)"` with *`Facility`* previously defined by the `Message Set` command; *`arg`* may include "`(position=`*`Position`*`)`" where *`Position`* may be either `Bottom` or `Top`, and "`(name=`*`Attachment`*`)`" where *`Attachment`* specifies the name of a MIME part, if one is added for the disclaimer; this defaults to `disclaimer.txt`.

For the `Fileinto`, `Keep`, and `Redirect` actions, *`arg`* may include the option `"(removeattachments=yes)"` to request removal of all attachments that meet the filter criteria (not including MIME message body). Users can be notified of removed attachments by means of the `Filter Set` command; see page 255.

For the `Fileinto` action, *`arg`* is a valid mailbox name or a quoted list of parenthesized parameters specifying mailbox name and IMAP flags to be set on the message in this form, where *`mbox`* is the name of a mailbox and *`IMAPflags`* is as described above for the `Keep` action.

```
"(mailbox=mbox)(flags=IMAPflags)"
```

For the `Fwdexcerpt` action, *`arg`* is a series of `(`*`option`*`=`*`value`*`)` pairs where *`option`* is one of the following:

- `dest`—The excerpt's destination, a *`user@domain`* address that may contain up to five of the following LDAP macros, although no actual LDAP lookup is performed:
  - `$(mbox)`
  - `$(domain)`
  - `$(login)` which is the same as `$(mbox)@$(domain)`
- `bodylines`—Number of message body lines to forward; the default is 3, and the maximum is 25 lines, up to a limit of 160 bytes.
- `addheader`—Adds this header string to forwarded excerpt, in addition to default `Date`, `From`, and `Subject` headers; may be used multiple times.
- `fullmessage`—Whether to send the full message body, either `yes` or `no` (default is `no`); may not be specified together with `bodylines` or `addheader`.

For the `Keep` action, *`arg`* may be empty or a quoted, parenthesized list of IMAP flags to be set on the message; the form is as follows, where *`IMAPflags`* is a space-separated list of IMAP flags to be set on the message. Flags must be given with two leading backslashes, or four in the CLI, as in the examples below:

```
"(flags=IMAPflags)"
"(flags=\\Draft \\Flagged)"
"(flags=\\\\Draft \\\\Flagged)"
```

For the `Redirect` action, *`arg`* must include a valid email address. If the address is the only argument, it may be given as a bare address, *`user`*`[+`*`folder`*`]@`*`host`*, or in parenthesized form (`address=`*`user`*`[+`*`folder`*`]@`*`host`*). Also allowed is an alternative parenthesized form (`mailhost=`*`host`*) which transfers the message to a specific mailhost rather than delivering to a single email address. If other arguments are included in *`arg`*, for instance (`extraheader=`*`header`*`: `*`value`*) or (`notify=`*`Facility`*), the parenthesized form of the email address is required. The message is sent to the given system after normal processing, thus enabling sender-based redirect. If `address=` and `mailhost=` are both specified, the new address replaces all envelope recipients in the redirected message. As stated above `(removeattachments=yes)` is also allowed.

For the `Wiretap` action, *`arg`* is a parenthesized list indicating `envelopefrom` (optional) and `envelopeto`. Bounces go to the `envelopefrom` address, or if omitted, the special "<>" address, often rewritten as `MAILER-DAEMON@`*`hostname`*. The message is sent to the `envelopeto` address before delivery to the recipient. Both envelope `(MAIL FROM)` and `(RCPT TO)` are rewritten accordingly. Headers `X-Mirapoint-Old-Envelope-From` and `X-Mirapoint-Old-Envelope-To` are added to preserve the original envelope.

```
"(from=<>)(to=censor@example.com)"
```

- *`any-all`* is one of:

- `Allof`—Requires that *all* the specified *`rules`* must be true for the *`action`* to be taken on a message. Use this keyword for filters without rules, such as inserting a disclaimer.
- `Anyof`—Requires that *at least one* of the specified *`rules`* must be true for the *`action`* to be taken on a message. Do not use for ruleless filters, when this keyword returns false.

◆ *`stop-go`* is one of:

- `Stop`—Prevents processing of further filters within the same scope. (A scope is a filter's combination of domain, mailbox, and priority. See Filter Add details). For example, a stop in a domain filter at priority 100 (a filter executed before spam/virus processing) will not prevent a filter at priority 500 (a normal filter) in the same domain from triggering. To avoid delivery by a later filter action (explicit or implied), specify a `Discard` or `Reject` action.
- `Continue`—Specifies that even if this filter's action is taken on a message, processing should continue with the next filter. Not allowed for quarantine.

◆ *`rules`* is a multi-line literal string with *byte* count containing one rule on each line. For the format of these lines, see Rules on page 240 and Literal Strings on page 48.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator (only that domain's filters)
- User (can access only his or her own filters)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
1 Filter Add user.u9 V1 fileinto "(mailbox=INBOX.Junk Mail)" allof stop {65+}
From contains "administrator"
Subject contains "Virus Warning"
1 OK Completed

1 Filter Add user.u9 V2 fileinto "(mailbox=INBOX.Junk Mail)" allof stop {43+}
X-Mirapoint-Virus contains "VIRUSDELETED"
1 OK Completed

2 Filter Add "(domain=example.com)" guardDL reject "" anyof stop {64+}
To contains "all@example.com"
From matches-not "@*example.com"
2 OK Completed

3 Filter Add INBOX rmVBS keep "(removeattachments=yes)" anyof stop {74+}
:ATTACHMENTFILENAME contains "vbs"
:ATTACHMENTTYPE matches "application/vbs"
3 OK Completed
```

```
4 Message Set en_US.ISO_8859-1 Filter.Disclaimer.Internal {149+}
To: $(to)

Disclaimer:
This e-mail is intended for the addressee shown.
It contains confidential information and protected from disclosure.

OK Completed

4 Filter Add "(domain=any)" internalDisclaimer Disclaimer
  "(message=Filter.Disclaimer.Internal)(position=top)" allof stop {44+}
:envelopeto in-ldap-group internal-users

OK Completed
```

Change

Changes the specified filter on the specified domain or mailbox. This command does not reorder the list of filters for that domain or mailbox.

Syntax

tag `Filter Change` *scope fname action arg any-all stop-go rules*

where:

- *scope* is as described for `Filter Add`.
- *fname* is the name of the filter you want to change.
- *action*, *arg*, *any-all*, *stop-go*, and *rules* are as described for `Filter Add`.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own filters)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
4 Filter Change user.demo test fileinto user.demo.junk anyof stop {66+}
Subject contains "Make money now!"
From contains "@spamcity.com"
4 OK Completed
```

Clear

Deletes all filters from the specified domain or mailbox.

Syntax

```
tag Filter Clear scope
```

where *scope* is as described for `Filter Add`.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own filters)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
15 Filter Clear user.demo
15 OK Completed

16 Filter Clear "(domain=example.com)"
16 OK Completed
```

Count

Responds with the number of filters currently associated with the specified domain or mailbox.

Syntax

```
tag Filter Count scope pattern
```

where *scope* is as described for `Filter Add` and *pattern* is currently ignored.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- User (can access only his or her own filters)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
5 Filter Count user.demo ""
* 5 1
5 OK Completed
```

```
6 Filter Count "(domain=example.com)" ""
* 6 1
6 OK Completed
```

Delete

Deletes the specified filter from the specified domain or mailbox.

Syntax

tag `Filter Delete` *scope fname*

where:

- *scope* is as described for `Filter Add`.
- *fname* is the name of the filter you want to delete (see `Filter Add`).

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own filters)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
13 Filter Delete user.demo test
13 OK Completed
```

```
14 Filter Delete "(domain=example.com)" test2
14 OK Completed
```

Export

Responds with a listing of the filterset associated with the specified domain or mailbox. You can use this output with `Filter Import` to apply the filterset to other domains or mailboxes.

Syntax

tag `Filter Export` *scope*

where *scope* is as described for `Filter Add`.

There is no way to export filters of all priorities simultaneously. You must specify `(domain=any)(priority=N)` where *N* is 100, 450, then 500.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- User (can access only his or her own filters)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
10 Filter Export user.demo
* 10 {145}
#@Mirapoint-Filter-1.0
filter "test" fileinto "user.demo.junk" anyof stop
Subject contains "Make money now!"
From contains "@spamcity.com"

10 OK Completed
```

Fetch

Responds with a detailed listing of the specified filter.

Syntax

tag `Filter Fetch` *scope fname*

- *scope* is as described for `Filter Add`.
- *fname* is the name of the filter you want to fetch (see `Filter Add`).

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- User (can access only his or her own filters)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
8 Filter Fetch user.joe "System Junk Mail Rule"
* 8 {88}
filter "System Junk Mail Rule" fileinto "INBOX.Junk Mail" allof stop
:UCE is "normal"

8 OK Completed
9 Filter Fetch "(domain=example.com)" test2
* 9 {102}
filter "test2" reject "" anyof stop
To contains "all@example.com"
From matches-not "@*example.com"

9 OK Completed
```

Get

Retrieves the value of a specified parameter.

Syntax

tag `Filter Get` *parameter*

where *parameter* may be `RemovedAttachmentsMessage`; see `Filter Set`.

Privilege Levels

- Administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
19 Filter Get removedAttachmentsMessage
{121}
Content-Type: text/plain;charset="us-ascii"

A message filter removed the following attachment(s) from this message: %F

19 Ok Completed
```

Import

Applies the filterset data returned by `Filter Export` to the specified domain or mailbox, replacing any filters previously associated with that domain or mailbox.

Syntax

```
tag Filter Import scope filterset
```

where:

- *scope* is as described for `Filter Add`.
- *filterset* is a literal string consisting of the data returned by an invocation of the `Filter Export` command.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own filters)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
11 Filter Import user.demo {126+}
#@Mirapoint-Filter-1.0
filter "test2" reject "" anyof stop
To contains "all@example.com"
From matches-not "@*example.com"

11 OK Completed
```

List

Responds with names of filters created for the specified domain, classification, or mailbox. This command displays only the filter names. To fetch a detailed listing of a filter, use `Filter Fetch`. To get a detailed listing of all filters associated with a domain or mailbox, use `Filter Export`.

There is no way to list filters of all priorities simultaneously. You must specify `(domain=any)(priority=`*N*`)` where *N* is 100, 450, then 500.

Syntax

```
tag Filter List scope pattern start count
```

where:

- *scope* is as described for `Filter Add`.
- *pattern* is currently ignored.
- *start* indicates the first filter you want to see. The empty string (`""`) implicitly means `0`.

- *count* indicates the number of filters you want to see (see the example below). The empty string ("") implicitly means all filters. If *count* is greater than the total number of filters, `list` returns as many filters as possible.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- User (can access only his or her own filters)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
7 Filter List user.demo "" "" ""
* 7 "test"
7 OK Completed
```

```
8 Filter List "(domain=example.com)" "" "" ""
* 8 "test2"
8 OK Completed
```

Move

Moves the specified filter to a different position in the filter list for the specified domain or mailbox (see Filters on page 239).

Syntax

```
tag Filter Move scope fname position
```

where:

- *scope* is as described for `Filter Add`.
- *fname* is the name of the filter you want to move (see `Filter Add`).
- *position* is one of:
  - Up—exchanges the specified filter with the previous filter in the list
  - Down—exchanges the specified filter with the next filter in the list
  - Top—moves the specified filter to the top of the list
  - Bottom—moves the specified filter to the bottom of the list

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own filters)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
12 Filter Move user.demo test2 Up
12 OK Completed
```

Set

Sets the value of a specified parameter.

Syntax

tag `Filter Set` *parameter value*

where:

- *parameter* must currently be `RemovedAttachmentsMessage`. See Filter Add on page 245 for information on removing attachments.
- *value* is a multi-line literal string specifying text that gets inserted into filtered messages that have had their attachments removed, or null string (`""`) to reinstate the default. Text can contain the following macro substitutions:
  - `%d`—Date and time when the attachment was filtered.
  - `%F`—Filename(s) of the removed attachment(s).
  - `%m`—MIME type(s) of the removed attachment(s).
  - `%s`—Size in bytes (octets) of the removed attachment(s).
  - `%t`—Recipients of the message (to)
  - `%f`—Sender of the message (from).
  - `%h`—Host name where the attachment was filtered.
  - `%%`—Literal percent (%) character.

Privilege Levels

- Administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
19 Filter Set removedAttachmentsMessage {123+}
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

%d: The following attachments were removed: %F

19 Ok Completed
```

Test

Tests the filterset for the specified domain or mailbox on the messages contained in the test mailbox and responds with the unique numeric ID of each message matched by a filter and the name of the filter that matched. To prepare for this test, you must populate the test mailbox with messages you want the filterset to match.

Syntax

```
tag Filter Test scope testmbox
```

where:

- *scope* is as described for `Filter Add`.
- *testmbox* is the name of the test mailbox.

Response

The command responds with a line for each test message matched. Each line has the format:

```
* tag uid fname
```

where:

- *uid* is the unique numeric ID of the test message that was matched by a filter.
- *fname* is the name of the filter that matched.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own filters)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system’s primary domain.

Example

```
17 Filter Test user.demo user.demo.junk
17 OK Completed
* 17 1 "testing 123"
* 17 2 "testing 456"
* 17 3 NIL
17 OK Completed
```

23

The Fwd Command

The Fwd command configures the Mirapoint system to forward all mail addressed to a particular user to the specified address.

Users can set an optional forwarding address from the Account Forwarding form of Administration Suite, or from the Options > Forwarding panel of WebMail Direct. Both interfaces call the Fwd command to set forwarding.

The quarantine filter for JMM, then other filters, are executed before forwarding.

LDAP Interaction

If the administrator has done a Conf Enable Ldapforward, then the Fwd protocol inserts the new forwarding address into the LDAP database. Without Ldapforward, the new forwarding address is stored on the local system only.

For LDAP forwarding, the mirapointMailUser object class should allow the entries miForwardingAddress and miDeliveryOption. The values of miDeliveryOption control how mail is delivered: forward and mailbox. The first line below is required to enable LDAP forwarding. The second line below requests delivery of forwarded messages to the local mailbox (like keepcopy in non-LDAP forwarding):

```
miDeliveryOption: forward
miDeliveryOption: mailbox
```

The attribute miForwardingAddress specifies forwarding addresses, but is ignored if miDeliveryOption forward is not present:

```
miForwardingAddress: juser@example.com
```

Autoprovisioning by SMTP (see Ldap Set on page 321) does not work if Ldapforward is configured On and users already have an LDAP forwarding address. Autoprovisioning by user login still works, though.

The Ldap Addaccess ... Rootpw command should be used to set the machine with appropriate bind DNs and passwords for updating the user's LDAP entry. If a user (who is not Administrator) executes the Fwd command, and that user has LDAP authentication enabled, the user's DN and password are taken as the BindDN and password for LDAP updates. In all other cases (when Administrator is modifying forward information, or when LDAP authentication is not enabled for a user) bindDNs and passwords specified by Ldap Addaccess are used for LDAP writes.

In other words, you probably want to set Ldap Addaccess ... Rootpw so that the administrator can modify forwarding, and to make mixed authentication work.

Subcommands

Get

Responds with the current forwarding configuration for the specified user. If the response specifies two empty strings ("" ""), this means that forwarding is disabled for the specified user account.

A user who doesn't have administration privileges is allowed to get the forwarding status of his own user account.

Syntax

```
tag Fwd Get username
```

where *username* identifies the user whose forwarding configuration you want to see. In the resulting output, after the forwarding address(es), empty string ("") implies NOKEEPCOPY.

Privilege Levels

- ◆ Administrator
- ◆ Helpdesk administrator
- ◆ Domain administrator
- ◆ Backup operator
- ◆ User (can access only his or her own account)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
4 Fwd Get demo
* 4 demo@example.com KEEPCOPY
4 OK Completed
```

```
5 Fwd Get juser
* 5 juser@example.com,juser9@webmail.net ""
5 OK Completed
```

Set

Sets the forwarding configuration for a particular user. Users who lack administration privileges are allowed to set forwarding only for their own user account. Forward works in conjunction with autoreply.

To turn off forwarding, specify the empty string for both the *address* and *keepcopy* parameters.

Syntax

`tag Fwd Set username address keepcopy`

where:

- *username* identifies the user whose forwarding configuration you want to set.
- *address* is the email address or addresses to which you want the user's mail forwarded. The address list is limited to 255 bytes and may not contain angle brackets (< or >). The format of address may be one of the following:
  - `user`
  - `user@`*domain*
  - a comma-separated list of the above, quoted if it contains spaces

  The string may *not* contain angle brackets (< and >).
- *keepcopy* is one of:
  - `KEEPCOPY`—Specifies that each forwarded message also be stored in the user's inbox.
  - `NOKEEPCOPY`—Specifies that each forwarded message not be stored in the user's inbox. This is the default.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own account)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
2 Fwd Set demo demo@example.com KEEPCOPY
2 OK Completed

3 Fwd Set juser "juser@example.com,juser9@webmail.net" ""
3 OK Completed
```

The Getmail Command

The `Getmail` command controls downloading of user email from remote systems. The Mirapoint get-mail facility may be disabled with the `Conf` command.

`Getmail Check` retrieves mail from remote mail servers and forwards it via SMTP to a Mirapoint inbox, where it can be read by the user's preferred email client.

Subcommands

Add

Adds a remote system to the user's list of POP servers to check for email download. If possible, `Getmail` downloads remote email using an SSL connection (POPS) on port 995, otherwise regular POP. Each user is limited to 10 `Getmail` records.

Syntax

tag `Getmail Add` *username nickname server userID password options*

where:

- *username* is the user's login name on the Mirapoint system.
- *nickname* is a convenient name given to the remote POP mailbox.
- *server* is the fully qualified host name of the remote POP server.
- *userID* is this user's login name on the remote POP server.
- *password* is this user's password on the remote POP server.
- *options* specify POP settings on the remote server, as an (attribute=value) list including any of the following items:
  - `Color`—A number between 0 and 9 specifying the color encoding of messages downloaded from this server; defaults to 0. Because color can be branded, numbers may correpond to various colors.
  - `Filtering`—Perform standard Mirapoint filtering on fetched messages. Either `Yes` or `No`; defaults to `Yes`. Mutually exclusive with `mailbox`.
  - `Leaveonserver`—Leave downloaded messages on server. Either `Yes` or `No`; defaults to `Yes`.
  - `Mailbox`—Specifies a mailbox name where fetched messages should be placed without filtering; defaults to unused because filtering is on.

- `Newmessagesonly`—Download only new messages. Either `Yes` or `No`; defaults to `Yes`.
- `Port`—The number of the remote POP port; defaults to 995 and 110.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (for their remote mail)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
2 Getmail Add juser @home pop.rrcny.com joe99 secret "(color=3)"
2 OK Completed
```

Change

Alters information about a remote system added as a user's POP server.

Syntax

```
tag Getmail Change username nickname server userID password options
```

where:

- *`username`* is the user's login name on the Mirapoint system.
- *`nickname`* is a convenient name given to the remote POP mailbox.
- *`server`* is the fully qualified host name of the remote server; the null string (`""`) means no change.
- *`userID`* is this user's login name on the remote POP server; the null string (`""`) means no change.
- *`password`* is this user's password on the remote POP server; the null string (`""`) means no change.
- *`options`* specify the remote server settings, as documented under `Getmail Add`; the null string (`""`) means no change.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

- Users (for their remote mail)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
8 Getmail Change juser @home pop.rrfla.com joe88 secret "(color=3)"
8 OK Completed
```

Check

Locates user mailboxes on remote systems and downloads any messages if present. Remote mailbox information is specified with `Getmail Add`.

The `Check` command retrieves mail from remote mail servers and forwards it via SMTP to the user's Mirapoint inbox, where it can be read by some email client. User filtering, forwarding, and autoreply work as they normally would.

Syntax

```
tag Getmail Check username pattern
```

where:

- *username* is the user's login name on the Mirapoint system.
- *pattern* must currently be `""` or `*` to indicate all remote system nicknames.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (for their remote mail)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
10 Getmail Check juser ""
[time lag]
10 OK Completed (6 messages downloaded)
```

Count

Counts the number of remote POP servers.

Syntax

tag Getmail Count *username pattern*

where:

- *username* is a user's login name, or the null string ("") to indicate current user.
- *pattern* must be the null string ("") to match all the user's added servers.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (for their remote mail)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
3 Getmail Count ""
* 3 2
3 OK Completed
```

Delete

Deletes a remote system from the user's list of POP servers.

Syntax

tag Getmail Delete *username nickname*

where:

- *username* is the user's login name on the Mirapoint system.
- *nickname* is the convenient name given to the remote POP mailbox.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (for their remote mail)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
7 Getmail Delete juser @home
7 OK Completed
```

Get

Responds with the value of the specified parameter.

Syntax

tag `Getmail Get` *parameter*

where *parameter* must currently be `Minpoll`, the minimum time lag between polling for remote mail to fetch.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Getmail Get Minpoll
* 4 10m
4 OK Completed
```

List

Displays information about remote POP servers. `Listbackup` is similar but returns more information, including encoded password, for backup purposes.

Syntax

tag `Getmail List` *username pattern start count*

tag `Getmail Listbackup` *username pattern start count*

where:

- *username* is a user's login name, or the null string (`""`) to indicate current user.
- *pattern* matches POP servers, or the null string (`""`) to match all servers.
- *start* and *count* specify the beginning and duration of servers to list.

Privilege Levels

The `List` command returns the convenient nickname, remote server name, remote user name, null string instead of password, and the most recent status message.

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (for their remote mail)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
5 Getmail List juser "" ""
5 "@home" pop.rrfla.com joe88 "" "Completed (5 messages downloaded)"
5 "@work" mail.example.com juser "" "Completed (29 messages downloaded)"
5 OK Completed
```

Rename

Changes the convenient nickname for a user's remote POP server.

Syntax

```
tag Getmail Rename username oldname nickname
```

where:

- *username* is the user's login name on the Mirapoint system.
- *oldname* is the previous nickname given to the remote POP mailbox.
- *nickname* is a new convenient name for the remote POP mailbox.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (for their remote mail)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
9 Getmail Rename juser @home @house2
9 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

tag `Getmail Set` *parameter value*

where:

- *parameter* must currently be `Minpoll`, the time lag between polling for remote mail to fetch. The default is 10 minutes.
- *value* is the value you want to assign to *parameter*.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Getmail Set Minpoll 15m
6 OK Completed
```

25

The Http Command

The Http command configures behavior of the HTTP server on a Mirapoint system.

Subcommands

Get

Responds with the value of the specified parameter.

Syntax

```
tag Http Get parameter
```

where *parameter* is one of those listed under Http Set.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
2 Http Get Root
* 2 Admin
2 OK Completed
```

```
8 Http Get Security
* 8 "cleartext cleartextout ssl sslout"
8 OK Completed
```

Set

Sets the value of the specified parameter.

To set up HTTP and LDAP routing for Outlook Web Access, use domain-based routing and set Ldaprouteurls and mode Ldapproxy:

```
> Ldap Set Ldif
```

```
mail:@webmail.example.com
mailhost:exchange.example.com
.
OK Completed
> Http Set Ldaprouteurls Always
OK Completed
> Http Set Mode Ldapproxy
OK Completed
```

Syntax

tag Http Set *parameter value*

where:

- *parameter* is one of the following:
  - `Cookies`—Strengthen the current session validation method to check for HTTP cookies (RFC-2109) in addition to session IDs. The *value* is one of:
    - `Disabled`—The system uses "long" session IDs in URLs for session validation, as it did before release 3.3. This is the default.
    - `Auto`—The system attempts to use both "long" session IDs and also HTTP cookies if the browser will accept them.
    - `Strict`—The system uses "short" session IDs and requires cookies. Browsers must be set to accept cookies when using this method.
  - `Ldaprouteurls`—Controls routing of unproxied URLs. The proxy uses existing domain-based routing mechanism to control which server provides information that the proxy cannot get from the session ID. These include: login pages, images, help files, and relative links without session ID. Thus the proxy system can support brands and domains without having them installed on the proxy, and WebMail and Calendar need not be installed on the proxy. Note: if you get the login pages from a different machine than the proxy with domain-based routing, on-page security settings come from that machine (the one generating the web page), so ensure this machine's security settings match the proxy. The *value* is one of:
    - `On`—Enables LDAP routing for URLs without proxying information (images, login pages, and so forth).
    - `Off`—No LDAP routing for unproxied URLs. This is the default.
    - `Always`—The normal Mirapoint routing based on session IDs is not done; all back-end routing is done through LDAP. This makes the proxy more efficient when used to front an Exchange server.
  - `Maxconnections`—Sets the number of simultaneous connections the proxy accepts before it starts refusing them. The proxy cleans up any idle connections it finds, but if there are no idle connections it quits listening until sufficient connections are freed. This setting is important because it influences memory use by the proxy.
  - `Maxdirectorytraversals`—Sets the maximum number of `"/../"` path components in a URL. If a client sends a URL request with more than the allowed number of traversals, it gets an HTTP 400 "Client error" response. Setting this to `""` disables traversal checking.
  - `Maxheaderlength`—Sets the maximum length of a request header line or URL. This helps prevent buffer overflow attacks. If a client sends a request

with overly long header, it gets an HTTP 413 “Request too large” response. Setting this to "" restores the default maximum header length.

- `Maxrequestbodysize`—Sets the maximum size of an HTTP request body. This checks to ensure that the `Content-Length`: field is set to a sane value. If a client sends a request with an invalid content length, it gets an HTTP 413 “Request too large” response. Setting this to "" restores the default maximum request body size.
- `Maxtotalheadersize`—Sets the maximum size of the header portion of an HTTP request. If a client sends a URL request with more data than allowed in the header portion, it gets an HTTP 413, “Request too large” response. Counting starts after the request line, and measures total size of the “Header Name: Value” fields in the request. The *`value`* can be a number between 0 and 100,000. Setting this to "" restores the default maximum.
- `Mode`—Controls how WebMail logins behave with LDAP authentication. The *`value`* is one of:
  - `Normal`—The server does not perform HTTP redirection (default).
  - `Ldapproxy`—HTTP service acts as a proxy for any applications that support HTTP redirection. This is useful for accessing WebMail service on multi-tier systems or behind a firewall. In this mode a system takes incoming HTTP connections, uses LDAP to find the home server, and then routes traffic between the user’s client and their home server. As with POP or IMAP proxying, `user:Mailhost` and `user:Loginid` LDAP queries must be set. All systems involved in proxying must be time synchronized, preferably with NTP. This proxy supports: cookies, session IDs, HTML, WebMail, Calendar, SSL, and XML. This feature was added in release 3.4 and is not compatible with earlier releases.
  - `Ldapredirect`—All applications that support HTTP redirection try to use LDAP data to perform browser redirects for users, helpdesk, and domain administrators. To allow system maintenance, secure HTTP applications do not redirect the administrator. Unlike IMAP and POP redirects, HTTP connections can be redirected (forwarded) only once.
- `Proxytimeout`—Sets the maximum duration a connection can stay idle before the proxy is allowed to close it. The default is 120 seconds.
- `Root`—The application that appears at the root URL of the web server (`http://`*`miHostname`*`/`). With any default, WebMail service appears under `http://`*`miHostname`*`/wm`, Calendar service under `http://`*`miHostname`*`/mc`, Corporate Edition WebMail and WebCal under `http://`*`miHostname`*`/em`, and administration under `http://`*`miHostname`*`/miradmin`. *`Value`* may be:
  - `Admin`—Display Administration Suite as default (deprecated).
  - `Calendar`—Display Mirapoint Calendar as the default.
  - `EnterpriseUi`—Display Corporate Edition for WebMail and WebCal. This might lead to standard edition if users are not allowed access.
  - `Miradmin`—Depending on system licensing, display `madmin`, `rgadmin`, `ocadmin`, and so on as default. With more than one of these options is licensed, the selection precedence is: Message Server, RazorGate, Operations Console, First Use (for system recovery).
  - `Spam`—Display Junk Mail Manager as the default.
  - `Webmail`—Display WebMail Direct as the default.
- `Security`—Security types allowed for HTTP connections. The *`value`* is a quoted, space-separated list specifying any of these data-privacy schemes:

  - `Cleartext`—No encryption of data for incoming HTTP connections.
  - `Cleartextout`—No data encryption for outgoing HTTP connections. Use this setting together with `Ssl` on an HTTP proxy to specify encrypted incoming HTTP connections with cleartext connections to the message server.
  - `Ssl`—Secure Sockets Layer (SSL, versions 2 and 3) and Transport Layer Security (TLS, version 1) encryption of data for incoming HTTP connections. Use this setting on both proxies and servers.
  - `Sslout`—SSL for outbound HTTP connections. Use this setting on an HTTP proxy to encrypt HTTP connections to the message server. When used together with `Cleartextout`, stays with whatever the client used when connecting to the back-end (cleartext stays cleartext and SSL stays SSL).

- *`value`* is the value you want to assign to *`parameter`* (see above).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Http Set Root Webmail
3 OK Completed

4 Http Set Mode Ldapredirect
4 OK Completed

5 Http Set Mode Ldapproxy
5 OK Completed

6 Http Set Security "Cleartext Cleartextout Ssl Sslout"
6 OK Completed

7 Http Set Cookies Disabled
7 OK Completed
```

26

The Imap Command

The `Imap` command configures behavior of the IMAP mail reading service.

IMAP service requires a license, usually based on the number of users allowed.

To see features in the Mirapoint IMAP server, run the CAPABILITY command. Currently supported capabilities are: `IMAP4 IMAP4rev1 ACL QUOTA LITERAL+ NAMESPACE UIDPLUS IDLE AUTH=PLAIN UNSELECT`. Mirapoint IMAP supports both Cyrus and UW namespaces, plus a "merged" namespace that works with both.

IMAP Service Modes

The IMAP service can operate in either of two modes: **normal mode** or **proxy mode**. Use `Imap Set Mode` to set IMAP service mode.

Normal Mode

In normal mode, the IMAP service accepts IMAP connections providing access to mailboxes on the local host only.

Proxy Mode

In this mode, IMAP service acts as an IMAP proxy. When a user connects to an IMAP proxy, the proxy uses an LDAP database to determine the mail host where the user's email is stored. It then logs into the IMAP service on that mail host and passes IMAP commands and responses between the user's mail client and IMAP service on that mail host for the duration of the user's IMAP connection.

For the IMAP proxy to work correctly, you must run the `Ldap Setquery` command to define `user:Mailhost` and `user:Loginid` query specifications correctly for your LDAP database(see LDAP Query Specifications on page 300).

You can use the IMAP service in proxy mode only if an Mail Routing license is installed on the system. Contact your Mirapoint sales representative for details.

IMAP Namespaces

The IMAP service can operate in either of two namespaces: Cyrus (from CMU) or UW (University of Washington), as described in this section. Both Cyrus and UW namespaces are compliant with IMAP standards. Cyrus is the Mirapoint default.

The following table shows differences in naming conventions:

UW name	**Cyrus name**	
INBOX	INBOX	
sent-items	INBOX.sent-items	
%	INBOX.% + INBOX	
anyName	INBOX.*anyName*	
#user.joe.INBOX	user.joe	
#user.joe	user.joe.	[folder only]
#user.*anyName*.INBOX	user.*anyName*	
#user.*anyName*	user.*anyName*.	[folder only]
#archive	archive	
#%	%	
#*anyName*	*anyName*	
#INBOX	**notPossible**	
#INBOX.sent-mail	**notPossible**	
#INBOX.*anyName*	**notPossible**	
notPossible	INBOX.INBOX	
notPossible	INBOX.INBOX.*anyName*	
notPossible	user.*username*.INBOX	
notPossible	user.*username*.INBOX.*anyName*	

In UW namespace, # refers to the root, or top of folder hierarchy.

Namespace type is determined at login time. So if the namespace is changed after a user logs in, the user is not affected until the next login.

Subcommands

Get

Responds with the value of the specified parameter.

Syntax

tag Imap Get *parameter*

where *parameter* is one of:

- Mode—the mode in which IMAP service currently operates (see Imap Set).
- Namespace—the namespace for IMAP, either Cyrus or UW (see Imap Set).
- Quotawarn—the percentage of a quota that must be exceeded on a mailbox before the IMAP service issues quota warnings to clients that have the mailbox open (see Imap Set).
- Proxycapability—the IMAP proxy capability string (see Imap Set).
- Security—security schemes allowed for IMAP connections (see Imap Set).
- Timeout—the IMAP service idle timeout in minutes. The IMAP service closes any connection that remains idle for this period.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

The `Quotawarn` parameter is specific to a delegated domain.

Example

```
3 Imap Get Quotawarn
* 3 90
3 OK

4 Imap Get Security
* 4 "ssl cleartext"
4 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

tag `Imap Set` *parameter value*

where:

- *parameter* is one of:
  - `Mode`—the mode in which the IMAP service is currently operating. The value of this parameter must be one of:
    - `Ldapproxy`—the IMAP service acts as an IMAP proxy (message director).
    - `Normal`—the IMAP service accepts normal IMAP connections, providing access to mailboxes on the local host.
  - `Namespace`—the namespace used for new IMAP logins to organize mailboxes. Setting namespace is best done at system inception, because changing IMAP namespace has convenience and performance implications, particularly on heavily populated systems. The value of this parameter must be one of:
    - `Cyrus`—IMAP service uses the Cyrus namespace. This is the default. In Cyrus namespace, a user's personal mailboxes appear as INBOX and subfolders (examples: `INBOX`, `INBOX.Sent`, and `INBOX.Draft`).
    - `Merged`—IMAP service uses a combined Cyrus and UW namespace. This works better than the others with many IMAP email clients, and is recommended although currently not the default.
    - `UW`—IMAP service uses University of Washington namespace . In the `UW` namespace, a user's personal mailboxes appear to the user at the top

level (examples: `INBOX`, `Sent`, and `Draft`). Some mail clients require the top-level namespace.

- `Proxycapability`—takes as *`value`* an ASCII printable string that the proxy reproduces in response to a `CAPABILITY` command. Either `IMAP4` or `IMAP4rev1` must be present in the string. If `IMAP4` is not present, `IMAP4rev1` must be the first capability. If `IMAP4` is present, it must be the first capability. If both are present, `IMAP4` must be immediately followed by `IMAP4rev1`. The proxy capability string is limited to 512 bytes, and must contain printable alpha-numeric characters with only one newline. This parameter is intended for experienced IMAP administrators only. It was designed for proxying non-Mirapoint servers but may be useful for Mirapoint message servers also. A null *`value`* returns to default.
- `Quotawarn`—the percentage of a quota that must be exceeded on a mailbox before the IMAP service issues quota warnings to clients that have the mailbox open. For example, if this percentage is 90 (the default) and a particular mailbox has a quota of 100 MB, the IMAP service begins issuing quota warnings for the mailbox when usage exceeds 90 MB. In delegated domains, the quota warning level is inherited from the primary domain.
- `Security`—the security schemes allowed for IMAP connections. The value of this parameter must be a quoted, space-separated list of any of the following data-privacy schemes:
  - `Cleartext`—No encryption of data for incoming IMAP connections.
  - `Cleartextout`—No encryption of data for outgoing IMAP connections. Use this setting together with `Ssl` on message proxies to specify encrypted incoming connections from mail clients and cleartext proxy connections to message servers.
  - `Secureauthonly`—When set, prohibits plaintext passwords on a clear-text connection. `Secureauthonly` requires that `Cleartext` be set. If `Secureauthonly` is set alone, then `Cleartext` is set automatically. If `Secureauthonly` is set with other values but not including `Cleartext`, an error message results.
  - `Ssl`—Secure Sockets Layer (SSL, versions 2 and 3) and Transport Layer Security (TLS, version 1) encryption of data for incoming IMAP connections. Use this setting on both message proxies and servers.
  - `Sslout`—SSL for outbound IMAP connections. Use this setting on a message proxy to encrypt IMAP proxy connections to message servers. When used together with `Cleartextout`, stays with whatever the client used when connecting to the back-end (cleartext stays cleartext and SSL stays SSL).
  - `Starttls`—If set, TLS is allowed and supported on incoming IMAP connections. Requires SSL license, weak or strong. The server advertises STARTTLS capability to notify clients that it is supported.
  - `Starttlsout`—If set, the IMAP proxy tries to use TLS when sending messages to remote mailhosts. Requires SSL license, weak or strong.

`Starttls` applies to both proxy and server, while `Starttlsout` applies only to the proxy.

  - `Timeout`—the IMAP service idle timeout in minutes (see `Imap Get`). Although you can specify any value, to comply with the IMAP4 standard, revision 1 (RFC 2060), you should set this timeout to at least 30 minutes.

- *`value`* is the value you want to assign to *`parameter`*.

Privilege Levels

Administrator

Domain Sensitivity

The `Quotawarn` parameter is specific to a delegated domain. Setting `Namespace` is not allowed in delegated domains, being reserved for future expansion. The other parameters (`Mode`, `Proxycapability`, `Security`, `Timeout`) are not domain specific.

Example

```
5 Imap Set Timeout 45
5 OK

6 Imap Set Security "Cleartext Ssl"
6 OK Completed

7 Imap Set Proxycapability \
"IMAP4 IMAP4rev1 ACL QUOTA LITERAL+ NAMESPACE UIDPLUS IDLE AUTH=PLAIN STARTTLS"
7 OK Completed

8 Imap Set Security "Starttls Starttlsout Secureauthonly"
8 OK Completed
```

The Kerb4 Command

The `Kerb4` command lets you configure a Mirapoint system to use Kerberos version 4 to do login authentication for IMAP clients.

Generating Srvtab Data

This section describes how to generate `srvtab` for use with the `Kerb4 Set Srvtab` data using the KTH Kerberos version 4 server on a UNIX platform. You must use a windowing system, such as Motif, that supports "copy and paste."

1. Using a shell window that supports "copy and paste," log into your Kerberos server as `root`.
2. Use "`kdb_edit`" to create a Kerberos instance for a Mirapoint system; use "`imap`" as the principal name and a Mirapoint system's host name as the instance name.
3. Use "`ext_srvtab` *`Mirapoint-hostname`*", where *`Mirapoint-hostname`* is a Mirapoint system's host name, to extract the `srvtab` data for a Mirapoint system; you must enter your Kerberos master key to do this. This generates a file named *`Mirapoint-hostname`*`-new-srvtab` in the current directory.
4. Use "`mimencode` *`Mirapoint-hostname-new-srvtab`*" to encode the `srvtab` information for a Mirapoint system; the encoded information is displayed on the standard output (`mimencode` is part of the freeware Metamail package available for download on the web).
5. Copy this encoded information into your copy-and-paste buffer.
6. Using a `telnet` client that supports "copy and paste," connect to the command-line interface on a Mirapoint system and log in as `Administrator`.
7. Type "**`Kerb4 Set Srvtab`**".
8. Paste the contents of your copy-and-paste buffer into your `telnet` window.
9. Press Enter.

Subcommands

Add

Adds the specified host to the end of a Mirapoint system's list of Kerberos servers.

Syntax

```
tag Kerb4 Add host
```

where *host* is the fully qualified domain name of the Kerberos server to add.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Kerb4 Add kerb.example.com
2 OK Completed
```

Count

Responds with the number of servers in a system's list of Kerberos servers.

Syntax

```
tag Kerb4 Count pattern
```

where *pattern* is currently ignored.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
3 Kerb4 Count ""
* 3 1
3 OK Completed
```

Delete

Deletes the specified host from a Mirapoint system's list of Kerberos servers.

Syntax

```
tag Kerb4 Delete host
```

where *host* is the Kerberos server you want to delete.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Kerb4 Delete kerb.example.com
5 OK Completed
```

Get

Responds with the value of the specified parameter.

Syntax

tag `Kerb4 Get` *parameter*

where *parameter* is one of:

- `Realm`—the name of your Kerberos realm (see `Kerb4 Set`).
- `Srvtab`—a MIME-encoded version of the `srvtab` data for a Mirapoint system(see Generating Srvtab Data on page 281). This value is returned in encoded format.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
2 Kerb4 Get Srvtab
* 2 {50}
aW1hcABod3Rlc3QyAE1JUkFQT0lOVC5DT00AAZG2Rc5k1mRh

2 OK Completed
```

List

Responds with the list of Kerberos servers that a Mirapoint system can query.

Syntax

tag `Kerb4 List` *pattern start count*

where:

- *pattern* is currently ignored.
- *start* is the number of the first Kerberos server you want to see. The empty string ("") implicitly means 0.
- *count* is the number of Kerberos servers you want to see. The empty string ("") implicitly means all servers. If *count* is greater than the total number of Kerberos servers, list returns as many servers as possible.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
4 Kerb4 List "" "" ""
* 4 kerb.example.com
4 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

```
tag Kerb4 Set parameter value
```

where:

- *parameter* is one of:
  - Realm—the value for this parameter must be the name of your Kerberos realm. This must be a valid DNS domain name.
  - Srvtab—the value must be the MIME-encoded srvtab data generated for a Mirapoint system by your Kerberos server (see Generating Srvtab Data on page 281). May be given in "(encodedpass=)" format.
- *value* is the value to which you want to set *parameter*.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Kerb4 Set Srvtab aW1hcABod3Rlc3QyAE1JUkFQT0lOVC5DT00AAZG2Rc5k1mRh
3 OK Completed
```

28

The Kerb5 Command

The `Kerb5` command lets you configure a Mirapoint system to use Kerberos version 5 to do login authentication for IMAP clients.

Supported encoding types for DES: des-cbc-crc, des-cbc-md4, des-cbc-md5. Supported enctypes for Triple DES: des3-cbc-md5, des3-cbc-sha1, des3-hmac-sha1. For AES: aes128-cts-hmac-sha1, aes256-cts-hmac-sha1, aes128-ctc, aes256_ctc.

Generating Srvtab Data

This section describes how to generate **srvtab** data for the `Kerb5 Set Srvtab` command. This example uses the MIT Kerberos version 5 server on a UNIX platform. You must have a windowing system that supports “copy and paste” such as X11 with Gnome or KDE. The `kadmin.local` facility is not available in all Kerberos implementations.

1. Using a shell window that supports “copy and paste,” log into your Kerberos server as `root`.
2. Create service tickets for the services with which you want to use Kerberos 5 (`imap`, `pop`, or `smtp`). Use the service name (`imap`, `pop`, or `smtp`) as the principal. Use the fully qualified domain name of the Mirapoint server as the instance name. For example, using `kadmin.local` to add the services:

   ```
   kadmin.local:  addprinc imap/myhostname.mirapoint.com
   kadmin.local:  addprinc pop/myhostname.mirapoint.com
   kadmin.local:  addprinc smtp/myhostname.mirapoint.com
   ```
3. Use the `ktadd` command in `kadmin` to write the ticket to a keytab file, and to generate a DES-CBC-CRC key. For example:

```
kadmin.local:  ktadd -k /tmp/myhostname.keytab imap/myhostname.mirapoint.com
kadmin.local:  ktadd -k /tmp/myhostname.keytab pop/myhostname.mirapoint.com
kadmin.local:  ktadd -k /tmp/myhostname.keytab smtp/myhostname.mirapoint.com
kadmin.local:  ktadd -k /tmp/qa90.keytab -e DES-CBC-CRC:normal
```

4. Use `mimencode` *`Mirapoint-hostname-new-keytab`* to encode the keytab information for a Mirapoint system; the encoded information is displayed on the standard output. (`mimencode` is part of the freeware Metamail package available for download on the web). For example:

```
./mimencode /tmp/myhostname.keytab
    BQIAAABFAAIADU1JUkFQT0lOVC5DT00ABGltYXAAF3BvbGFyYmVhci5taXJhcG9pbnQuY29tAA
    AAATnuJBQEAAEACEYZZZxXtRxDAAAARAACAA1NSVJBUE9JTlQuQ09NANwb3AAF3BvbGFyYmVhc
    i5taXJhcG9pbnQuY29tAAAAATnuJB8CAAEACMOmbv1bik+X
```

5. Copy this encoded information into your copy-and-paste buffer.

6. Using a telnet client that supports "copy and paste," connect to the command-line interface on a Mirapoint system and log in as Administrator.
7. Type "`Kerb5 Set Srvtab`".
8. Paste the contents of your copy-and-paste buffer into your telnet window and press Enter.
9. Remember to delete the temporary keytab file on your Kerberos 5 server, for example:

```
rm /tmp/myhostname.keytab
```

Subcommands

Add

Adds the specified host to the end of a Mirapoint system's list of Kerberos servers.

Syntax

`*tag* Kerb5 Add *host*`

where *host* is the fully qualified domain name of the Kerberos server to include.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Kerb5 Add kerb.example.com
2 OK Completed
```

Count

Responds with the number of lines in a Mirapoint system's list of Kerberos servers.

Syntax

`*tag* Kerb5 Count *pattern*`

where *pattern* is currently ignored.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
3 Kerb5 Count “”
* 3 1
3 OK Completed
```

Delete

Deletes the specified host from a Mirapoint system’s list of Kerberos servers.

Syntax

tag `Kerb5 Delete` *host*

where *host* is the Kerberos server you want to delete.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Kerb5 Delete kerb.example.com
5 OK Completed
```

Get

Responds with the value of the specified parameter.

Syntax

tag `Kerb5 Get` *parameter*

where *parameter* is one of:

- `Realm`—the name of your Kerberos realm (see `Kerb5 Set`).
- `Srvtab`—a MIME-encoded version of the `srvtab` data for a Mirapoint system (see Generating Srvtab Data on page 287). This value is returned in encoded format.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
2 Kerb5 Get Srvtab
* 2 {50}
aW1hcABod3Rlc3QyAE1JUkFQT0lOVC5DT00AAZG2Rc5k1mRh

2 OK Completed
```

List

Responds with the list of Kerberos servers that a Mirapoint system can query.

Syntax

tag `Kerb5 List` *pattern start count*

where:

- *pattern* is currently ignored.
- *start* is the number of the first Kerberos server you want to see. The empty string ("") implicitly means `0`.
- *count* is the number of Kerberos servers you want to see. The empty string ("") implicitly means all servers. If *count* is greater than the total number of Kerberos servers, `list` returns as many servers as possible.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
4 Kerb5 List "" "" ""
* 4 kerb.example.com
4 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

tag `Kerb5 Set` *parameter value*

where *parameter* is one of:

- `Realm`—the value for this parameter must be the name of your Kerberos realm. This must be a valid DNS domain name.
- `Srvtab`—the value for this parameter must be the MIME-encoded `srvtab` data generated for a Mirapoint system by your Kerberos server (see Generating Srvtab Data on page 287). May be given in "`(encodedpass=)`" format.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Kerb5 Set Srvtab aW1hcABod3Rlc3QyAE1JUkFQT0lOVC5DT00AAZG2Rc5k1mRh
3 OK Completed
```

29

The Key Command

The `Key` command manages secure keys for logins to Mirapoint systems. Currently MTA, AUTH, and POP are the only supported key types. In the future, keys for SSH, SSL, and MOC (Mirapoint Operations Console) may be supported.

Subcommands

Count

Displays the number of keys of a given type.

Syntax

```
tag Key Count type pattern
```

where *type* must currently be `Mta`, `Auth`, `Pop`, or the null string (`""`), and *pattern* may be * or "" to match everything.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

Returns error if run in a delegated domain.

Example

```
6 Key Count "" ""
* 6 1
6 OK Completed
```

Delete

Irrevocably deletes the key for a given interface.

Syntax

```
tag Key Delete type interface
```

where:

- *type* is the key category, which currently must be `Mta`, `Auth`, or `Pop`.
- *interface* is the network address on hostname, as created by `Key New`.

Privilege Levels

Administrator

Domain Sensitivity

Returns error if run in a delegated domain.

Example

```
8 Key Delete Mta ""
8 OK MTA key for thishost.example.com deleted
```

Get

Shows the full key of one type for a given interface.

Syntax

```
tag Key Get type interface
```

where:

- *type* is the key category, which currently must be `Mta`, `Auth`, or `Pop`.
- *interface* is the network address on hostname, as created by `Key New`.

Response

The output of `Key Get` is:

```
* tag type full name interface arguments length
PEM stream
tag OK Completed
```

Where *type* is the key category, either `Mta`, `Auth`, or `Pop`.

The keyword `full` denotes that the key includes both public and private halves. Variables *name*, *interface*, and *arguments* show information from key creation time. Note that these are always represented in the PEM stream, which also gives the key information. `Key Set` with the PEM stream restores the key to its same state as when `Key Get` was run.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

Returns error if run in a delegated domain.

Example

```
3 Key Get Mta ""
* 3 MTA full "thishost.example.com" "" "" {1930}
PEM stream
3 OK Completed
```

Getpublic

Shows the public key of one type for a given interface.

Syntax

```
tag Key Getpublic type interface
```

where:

- *type* is the key category, which currently must be `Mta`, `Auth`, or `Pop`.
- *interface* is the network address on hostname, as created by `Key New`.

Response

The output of `Key Getpublic` is:

```
* tag type public name interface arguments length
PEM stream
tag OK Completed
```

Where *type* is the key category, either `Mta` or `Pop`.

The keyword `public` denotes that the key only the public half. Variables *name*, *interface*, and *arguments* show information from key creation time. Note that these are always represented in the PEM stream, which also gives key information. `Key Set` with the PEM stream restores the key to its state when `Key Get` was run.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

Returns error if run in a delegated domain.

Example

```
4 Key Getpublic Mta ""
* 4 MTA public "thishost.example.com" "" "" {1004}
PEM stream
4 OK Completed
```

List

Displays information about the keys of a given type.

Syntax

```
tag Key List type pattern start count
```

where *type* must be `Mta`, `Pop`, or the null string (""); *pattern* may be * or "" to match everything, *start* is the first element to list, and *count* is the number to list.

Privilege Levels

Administrator

Domain Sensitivity

Returns error if run in a delegated domain.

Example

```
7 Key List Mta "" "" ""
* 7 MTA thishost.example.com "" ""
7 OK Completed
```

New

Create a new key of the given type, on the given interface and hostname, as modifed by optional arguments. Not all key types require all parameters.

Currently the key options are for POP, NTLM (AUTH), and SMTP, the mail transfer agent (MTA). After one is created, its public half is available at `http://hostname/keys/KEY.mpk` so other hosts can access it to establish trust relationships.

Syntax

```
tag Key New type interface hostname arguments
```

where:

- *type* is the key category, which currently must be `Mta`, `Auth`, or `Pop`.
- *interface* is the network address, which must be `""` for both key types.
- *hostname* is the system name for the created key. If it is not null string (""), `localhost`, or the hostname of this system, a warning results, but the key is created anyhow. You can create a useless key by passing a bogus hostname. MTA, AUTH, and POP keys are created for all interfaces on *hostname*.
- *arguments* specify options, and must be "" for all keys.

Privilege Levels

Administrator

Domain Sensitivity

Returns error if run in a delegated domain.

Example

```
2 Key New Mta "" "" ""
2 OK MTA key created for thishost.example.com
```

Set

Alters the full key of one type for a given interface. It is an error to include only the private half or only the public half of the key to `Key Set`.

Syntax

tag `Key Set` *type interface key*

where:

- *type* is the key category, which currently must be `Mta`, `Auth`, or `Pop`.
- *interface* is the network address on hostname, as created by `Key New`.
- *key* is a counted string literal, in PEM format. Both public and private halves of a key must be specified.

Privilege Levels

Administrator

Domain Sensitivity

Returns error if run in a delegated domain.

Example

```
5 Key Set Mta "" {1930+}
#@Mirapoint-Key-1.0
VHIwZT1NVEEKSG9zdG5hbWU9ZG9jMS5taXJhcG9pbnQuY29tCIBF
...
#@Mirapoint-Key-End
5 OK Completed
```

30 The Ldap Command

The `Ldap` command configures a Mirapoint server for login authentication and mail routing using LDAP (lightweight directory access protocol). An **LDAP server** is any host that answers LDAP queries. The Mirapoint system maintains a list of LDAP servers to which it can issue queries. If the first server in the list does not respond, it queries the next server in the list, and so on until a server responds or the list ends.

LDAP mail routing requires a license. This license is a prerequisite for many other licensed features such as SMTP directory-based routing, IMAP or POP proxying, multi-tier Group Calendar, and multi-tier shared folders.

You can perform domain-based routing with LDAP (or LDIF) by having an entry for any type of domain in the following form, where *`domain`* indicates the domain name for routing and *`hostname`* indicates the message server:

```
dn: domain
mail: @domain
mailhost: hostname
```

For information about configuring LDAP, refer to the Mirapoint Customer Service website at http://support.mirapoint.com.

The `Dir Getschema Mirapoint` command displays the Mirapoint LDAP schema.

LDAP Server Specifications

To identify LDAP servers, use an LDAP server specification in the following format:

`protocol``://`*`hostname`*`:`*`port`*

where:

- *`protocol`* (optional) is one of
  - `ldap`—LDAP protocol over a cleartext connection (the default).
  - `ldaps`—LDAP protocol over SSL (see `Admin Set Security`).
- *`hostname`* is the fully qualified domain name of the LDAP server host.
- *`port`* (optional) is the TCP port on which LDAP server *`hostname`* listens.

"*`protocol`*`://`" and "`:`*`port`*" are optional; you may omit either or both.

LDAP Query Specifications

An LDAP query specification is a named collection of configuration data used to retrieve a particular piece of information from an LDAP database. Query mapping is necessary because various LDAP databases use different attribute names to identify the same information.

Each query specification encodes the following information necessary to query the LDAP directory server:

- A base distinguished name (baseDN) specifying an area of the LDAP hierarchy in which to search.
- A filter string, consisting of LDAP attribute-value pairs. Matching records must have the specified values for at least one of the specified attributes.
- The name of the LDAP attribute whose value you want to retrieve.

Query Specification Names

A query specification name is a Mirapoint-defined name for an LDAP attribute. Query specification names have the following format:

object:*name*

where:

- *object* identifies the LDAP object from which to retrieve the attribute value. Recognized values are:
  - `Mailgroup`—Mailing (distribution) lists
  - `User`—User records
- *name* is an identifier for the information to be retrieved.

Here are currently supported query specification names:

- `Mailbox:Mailhost`—Encapsulates location of shared folders (mailboxes), including baseDN of the directory information tree, a query filter to retrieve the list of shared folders, and a mailhost where shared folders reside. This is the primary filter for the miFolder object class.
- `Mailgroup:AllowedBroadcaster`—If a mailgroup has `Allowed*` attribute, then access control is enabled on the mailgroup (access control is disabled otherwise). If there is a `mgrpAllowedBroadcaster` LDAP attribute in `mailto` URL form, for example `mailto:Juser@example.com`, then the sender is allowed to send to this mailgroup.
- `Mailgroup:AllowedDomain`—Same as above but for domains. If there is a `mgrpAllowedDomain` LDAP attribute with the value of a DNS domain name, for instance `example.com`, then any sender in this domain (or any subdomain) is allowed to send to the mailgroup.
- `Mailgroup:Members`—The list of members in a mail distribution list. This is the primary mailgroup query.
- `Mailgroup:Owner`—The owner of a mail group (LDAP distribution list). Mirapoint supports but does not require `owner-`*dlname* for mailgroup *dlname*.

- `User:Fullname`—User's full name; may be a primary or secondary attribute.
- `User:Groupmembership`—Indicates whether user is a member of the specified group (for instance, object class groupOfUniqueNames).
- `User:Localaddr`—An alternate email address for the user.
- `User:Loginid`—Login name with which a user logs into a Mirapoint system.
- `User:Mailhost`—The fully qualified domain name of a user's message server.
- `User:Ntuserid`—The user's email address derived from NTuser@NTdomain. A recommended query specification, different from that for Publishedname, is: `"(&(objectClass=*)(&(NTuser=$(mbox))(NTdomain=$(domain)))"`
- `User:Publishedname`—A user's official email address. Primary user query.
- `User:Quota`—User's disk storage quota in bytes. Disk quotas are optional.
- `User:Routingaddr`—Complete address for delivering mail (uid@mailhost).
- `User:Uuid`—Unique user ID, a string from which login ID and globally unique identifier (GID) can be extracted. If the `Uuid` string starts with the characters "!uuid!" then all following characters until the next "!" constitute the GID. Characters after that constitute login ID. The GID is obtained from IMAP userID and an additional LDAP identifier. Used for calendar rename. You must restart Calendar service after instituting the `Uuid` (miUUID) attribute.

Table 3 summarizes the Mirapoint query specifications and gives the corresponding LDAP attribute(s) normally used.

Table 3 LDAP Query Specifications

Query Spec	Purpose and Usage	LDAP Attributes
user:Fullname	User's full name, only needed so message servers can supply it in the To header of outgoing email.	cn fullname
user:Groupmembership	Indicates whether the user belongs to the specified mail group.	(object class) groupOfUniqueNames
user:Localaddr	Alternate address (email alias) for the user. You can specify many of these.	mailLocalAddress mailAlternateAddress
user:Loginid	User ID that proxies employ to rewrite the given user name so that message servers recognize it.	miLoginID uid
user:Mailhost	Hostname or IPaddress of user's message server.	mailhost
user:Ntuserid	Performs authentication for **Ntlm** command.	NTuser@NTdomain
user:Publishedname	User's official email address.	mail
user:Quota	Disk space limit for the user's mailbox, in bytes.	miMailQuota
user:Routingaddr	Complete address for delivering email, usually loginid@mailhost; required by Group Calendar.	mailRoutingAddress rfc822mailbox
user:Uuid	Site-wide unique user ID, for Group Calendar.	miUUID
mailbox:Mailhost	Locates shared folders in a multi-tier network.	(folder) mailhost

Table 3 LDAP Query Specifications

Query Spec	Purpose and Usage	LDAP Attributes
mailgroup: AllowedBroadcaster	User addresses that are allowed to send email to a mailgroup, if access control is enabled.	mgrpAllowedBroadcaster
mailgroup: AllowedDomain	Mail domains that are allowed to send email to a mailgroup, if access control is enabled.	mgrpAllowedDomain
mailgroup:Members	Users who receive email sent to an LDAP group (object classes mailGroup and groupOfNames).	uniqueMember mgrpRfc822mailMember
mailgroup:Owner	Administrator of an LDAP mail group.	owner

Using Variables in Distinguished Names and Filters

To configure query specifications, use the `Ldap Setquery` command. In specifying filters (and base DNs) for the `Setquery` command, the following variables represent user-specific or mailgroup-specific data:

- `$(mbox)`—The name of the user's mailbox, such as `babs_jensen`.
- `$(domain)`—A user's or mail group's full DNS domain, such as `example.com`.
- `$(login)`—A user's full login name. In delegated domains this is equivalent to `$(mbox)@$(domain)`, but in the primary domain it is equivalent to `$(mbox)`.
- `$(dcmap)`—A comma-separated list of the components of `$(domain)`, such as "`dc=example,dc=com`".
- `$(group)`—Fully qualified address of a mail group, such as `g9@example.com`, like a DL, used to configure `Mailgroup` queries and user:Groupmembership.
- `$(groupname)`—The unqualified name portion (left-hand side) of `$(group)`, such as `g9`, used to configure `Mailgroup` queries and User:Groupmembership.

  If a mail group exists on a message server only, `$(group)` and `$(groupname)` are both unqualified on the message router until you create an LDAP "mail" record for the LDAP mail group and a mailHost record for the message server where the mail group exists, and set LDAP filters accordingly.

For example, the filter for the `User:Publishedname` query might look like this:

```
"uid=$(mbox),$(dcmap)"
```

Similarly, the filter for the `Mailgroup:Members` query might look like this:

```
"(cn=$(group))"
```

With Active Directory LDAP, the base DN for `User:Publishedname` query may be:

```
"cn=Recipients,ou=mail3,o=example,c=US"
```

Related to this, the filter might be:

```
"cn=$(mbox)"
```

LDAP Object Class Specifications

Several Mirapoint applications employ an object class to create LDAP user entries. To allow this, you must define the following **object class specifications**:

- `User:Person`—specifies the object class for user LDAP entries. This object class must allow the `userPassword` attribute and attributes specified by the `User:Publishedname` and `User:Fullname` query specifications. This object class may be mapped to `Person`, `inetOrgPerson`, or other common classes.
- `User:Mailsubscriber`—specifies an object class for extra user attributes. This object class must allow attributes specified by `User:Publishedname`, `User:Mailhost`, `User:Routingaddr`, `User:Loginid`, and `User:Quota` query specifications. If this class requires other attributes, they must be present in user entries when this object class is specified (see `Ldap Setobjclass`).

The OIDs of SunOne's ntGroupType and mailAlternateAddress attributes conflict with Mirapoint's mailRoutingAddress and mailLocalAddress, respectively. If this causes a problem, you must change one schema or the other.

LDAP Access Profiles

Multi-tier installations must be able to access and modify data in different areas of the LDAP database. For a given location in the database, specified by a baseDN, an **access profile** provides the authentication information needed to access and modify that part of the database. This information consists of a distinguished name known as the **bindDN** and a password. Binding is done anonymously, unless an access profile is defined by the `Ldap Addaccess` command, in which case LDAP queries check the access profile. This affects `Ldap Testquery`: if a bindDN and password combination is invalid, `Ldap Testquery` access is refused, and lookup fails.

Changing Your Primary LDAP Server

Mirapoint systems can maintain a list of servers to which they issues LDAP queries. If the first server in the list doesn't respond, the system queries the next server in the list, and continues down the list until an LDAP server (hopefully) responds.

To change your primary LDAP server (the first LDAP server in the list):

1. `Ldap Delete` all directory servers except the primary.
2. `Ldap Add` your new primary server, making it the second server in the list.
3. `Ldap Delete` your old primary server, promoting your new server to be the primary server.
4. `Ldap Add` again any directory servers that you deleted in step 1.

Subcommands

Add

Adds an LDAP server to the end of your Mirapoint system's list of LDAP servers.

When multiple LDAP servers are added, the client waits 5 seconds for a connection, and 30 seconds for an operation, before trying the next LDAP server.

Syntax

tag Ldap Add *server-spec*

where *server-spec* is a server specification identifying the LDAP server you want to add (see LDAP Server Specifications on page 299).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Ldap Add 192.168.0.132
2 OK
```

Addaccess

Adds an access profile to allow the Mirapoint system to access and modify an area of your LDAP database.

Syntax

tag Ldap Addaccess *baseDN bindDN passwd passwd-type*

where:

- *baseDN* is the base distinguished name (baseDN) of the location in the LDAP database that you want your Mirapoint system to access.
- *bindDN* is distinguished name (bindDN) used for authentication.
- *passwd* is the password for the specified bindDN.
- *passwd-type* is one of:
  - Encodedpass—the password is encoded (see User Set)
  - Pass—the password is plaintext (see User Set)

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Ldap Addaccess o=top cn=admin,o=top secret pass
8 OK Completed
```

Addmailhost

Adds knowledge of a Junk Mail Manager “mailhost” to LDAP routing.

Syntax

```
tag Ldap Addmailhost domain mailhost
```

where `domain` is the mail domain for which `mailhost` stores non-spam messages. You are limited to 256 entries.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
31 Ldap Addmailhost example.com mailsrv
31 OK Completed
```

Addquarantinehost

Adds knowledge of a Junk Mail Manager “quarantine host” to LDAP routing.

Syntax

```
tag Ldap Addquarantinehost domain mailhost
```

where `domain` is the mail domain for which `mailhost` stores spam messages. You are limited to 256 entries.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
31 Ldap Addquarantinehost example.com spamsrv
31 OK Completed
```

Count

Responds with the number of servers in your Mirapoint system's list of LDAP servers.

Syntax

```
tag Ldap Count pattern
```

where *pattern* is currently ignored.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

None

Example

```
3 Ldap Count ""
* 3 6
3 OK Completed
```

Countaccess

Responds with the current number of LDAP access profiles (see `Ldap Addaccess`).

Syntax

```
tag Ldap Countaccess pattern
```

where *pattern* must currently be * or "" (empty string), meaning all access profiles.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
11 Ldap Countaccess
11 4 0
11 OK Completed
```

Countobjclass

Responds with the number of supported LDAP object class specifications.

Syntax

tag `Ldap Countobjclass` *pattern*

where *pattern* must currently be * or "" (emtpy string), meaning all object class specifications.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
16 Ldap Countobjclass
* 16 2
16 OK Completed
```

Countmailhost

Count the number of Junk Mail Manager mailhosts.

Syntax

tag `Ldap Countmailhost` *pattern*

where *pattern* must currently be * or "" (emtpy string) to match all mailhosts.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
33 Ldap Countmailhost example.com
* 33 1
33 OK Completed
```

Countquarantinehost

Count the number of Junk Mail Manager quarantine hosts.

Syntax

```
tag Ldap Countquarantinehost pattern
```

where *pattern* must currently be * or "" (emtpy string) to match all mailhosts.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
34 Ldap Countquarantinehost example.com
* 34 1
34 OK Completed
```

Countqueries

Responds with the number of supported LDAP query specifications.

Syntax

```
tag Ldap Countqueries pattern
```

where *pattern* is currently ignored.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
6 Ldap Countqueries ""
* 6 6
6 OK Completed
```

Delete

Deletes the specified LDAP server from your Mirapoint system’s list of LDAP servers.

Syntax

```
tag Ldap Delete server-spec
```

where *server-spec* identifies the LDAP server you want to delete (see LDAP Server Specifications on page 299).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Ldap Delete 192.168.0.132
4 OK Completed
```

Deleteaccess

Deletes the access profile for the specified baseDN.

Syntax

```
tag Ldap Deleteaccess baseDN
```

where *baseDN* is the base distinguished name for which you want to delete the access profile (see LDAP Access Profiles on page 303).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Ldap Deleteaccess o=top
8 OK Completed
```

Deletemailhost

Deletes domain of a Junk Mail Manager “mailhost” from LDAP routing.

Syntax

```
tag Ldap Deletemailhost domain
```

where *domain* is the mail domain for which mailhost handled non-spam messages.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
37 Ldap Deletemailhost example.com
37 OK Completed
```

Deletequarantinehost

Deletes domain of a Junk Mail Manager “quarantine host” from LDAP routing.

Syntax

```
tag Ldap Deletequarantinehost domain
```

where *domain* is the mail domain for which mailhost stored spam messages.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
38 Ldap Deletequarantinehost example.com
38 OK Completed
```

Flushcache

Flushes (empties) the internal cache of LDAP query results.

Syntax

```
tag Ldap Flushcache table key
```

where:

- *table* is one of:
    - `User`—Flush only results from `User` queries.
    - `Mailgroup`—Flush only results from Mailgroup queries.
    - `Loginbeforesmtp`—Reloads remote server data for Logging role.
    - `Cos`—Flush only results from the Class of Service (COS) table.
    - The empty string (`""`)—Flush results from any query. If you specify `""` for *table*, the *key* argument must also be the empty string.
- *key* is the lookup key (login name or a mailgroup name) for which you want to flush cached query results, or the empty string (`""`), meaning all query results for *table* or for all tables. For `Cos` with indirect specification, use COSDN as the lookup key.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Ldap Flushcache user joe
5 OK Completed

6 Ldap Flushcache cos uid=gold,ou=people,o=example.com
6 OK Completed
```

Get

Responds with the value of the specified parameter.

Syntax

```
tag Ldap Get parameter
```

where *parameter* is one of:

- Autoprovision—Whether autoprovisioning of user accounts and mailboxes is enabled (ON or OFF). See Ldap Set.
- Cachetimeout—The period that LDAP query results are cached internally. A value of 0 indicates that caching is disabled. The value has one of the following suffixes indicating time units:
  - w—weeks
  - d—days
  - h—hours
  - m—minutes
  - s—seconds
- Compatv1routing—Whether LDAP-based message routing is done using version 1 semantics (ON or OFF). Deprecated.
- Ldif—LDAP Data Interchange Format (LDIF) stored and used locally for message routing and proxying instead of an LDAP database (see Ldap Set).
- Localcostable—Whether to perform COS lookups against the internal directory server's local routing table.
- Mirabase—The base distinguished name (baseDN) of the location in your LDAP database where Mirapoint-specific configuration is stored.
- Subdomainroutinglevel—Domain or subdomain dots on which to perform domain-based LDAP routing (see Ldap Set).

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

None

Example

```
12 Ldap Get Cachetimeout
* 12 30m
12 OK Completed
13 Ldap Get Compatv1routing
* 13 OFF
13 OK Completed
```

Getaccess

Responds with the access profile (consisting of bindDN and encoded password) for the specified baseDN.

Syntax

```
tag Ldap Getaccess baseDN
```

where *baseDN* specifies the access profile that you want to get.

Response

The response line has the format:

```
* tag "bindDN" enc-pass encodedpass
```

where:

- *bindDN* is the bindDN
- *enc-pass* is an encoded version of the password

The final field is always the literal keyword `encodedpass`, indicating the password type. It is not possible to retrieve a plaintext password.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
14 Ldap Getaccess o=top
14 cn=admin,o=top IPISoOlbt encodepass
14 OK Completed
```

Getmailhost

Shows the mailhost associated with a given Junk Mail Manager domain.

Syntax

```
tag Ldap Getmailhost jmmDomain
```

where *jmmDomain* is a Junk Mail Manager domain returned by `Ldap Listmailhost`.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
35 Ldap Getmailhost example.com
* 35 exchange.example.com
35 OK Completed
```

Getobjclass

Responds with the object class associated with an object class specification (see LDAP Object Class Specifications on page 303 to learn about object class specifications).

Syntax

tag Ldap Getobjclass *ocspec*

where *ocspec* is one of the pre-defined object class specification names (see LDAP Object Class Specifications on page 303).

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
20 Ldap Getobjclass user:Person
* 20 mirapointUser
20 OK Completed
```

Getquarantinehost

Shows the Junk Mail Manager (quarantine) host for a given Junk Mail domain.

Syntax

tag Ldap Getquarantinehost *jmmDomain*

where *jmmDomain* is a Junk Mail domain returned from `Ldap Listquarantinehost`.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
36 Ldap Getquarantinehost example.com
* 36 jmm.example.com
36 OK Completed
```

Getquery

Responds with the base distinguished name, filter, and attribute name for the specified query specification (see LDAP Query Specifications on page 300 to learn about query specifications).

Syntax

tag `Ldap Getquery` *queryspec*

where `queryspec` is one of the pre-defined query specification names (see LDAP Query Specifications on page 300).

Response

The response line has the following format:

`*` *tag* "*basedn*" "*filter*" "*attr*" "*type*"

where:

- *basedn* is the base distinguished name that specifies the area of the LDAP hierarchy in which to search.
- *filter* is a filter string, consisting of LDAP attribute-value pairs; matching records have the specified values for all specified attributes.
- *attr* is the name of the LDAP attribute whose value you want to retrieve. For `Mailgroup:Members`, this is a quoted, space-separated list of members, which can be either `direct` or `indirect`, as indicated by the *type* parameter. For `User:Fullname`, this is a quoted, space-separated list of LDAP attributes for name, indicating `primary` and `secondary`, as indicated by the *type* parameter.
- *type* is used for the `Mailgroup:Members` and `User:Fullname` queries. You must specify a type, one-to-one, for each member in the *attr* list. Concatenation of multiple attributes is not supported.

- For `Mailgroup:Members` the types are as follows:
  - `Direct`—says corresponding member is an RFC 822 email address.
  - `Indirect`—indicates corresponding member is a distinguished name that can be looked up using the `User:Publishedname` query.
- For `User:Fullname` the types are as follows:
  - `Primary`—indicates primary attribute supplying full name.
  - `Secondary`—indicates secondary attribute supplying full name.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
6 Ldap Getquery user:routingaddr
* 6 c=US "(|(mail=$(login))(mailaltaddress=$(login)))" mail ""
6 OK Completed
```

List

Responds with the current list of all the LDAP servers that your Mirapoint system can use(see LDAP Server Specifications on page 299).

Syntax

`tag Ldap List pattern start count`

where:

- *`pattern`* must currently be * or `""` (empty string), meaning all LDAP servers.
- *`start`* indicates the first LDAP server you want to see. The empty string (`""`) implicitly means `0`.
- *`count`* indicates the number of LDAP servers you want to see. See the Example below. The empty string (`""`) implicitly means all LDAP servers. If *`count`* is greater than the total number of LDAP servers, `list` returns as many LDAP servers as possible.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

None

Example

```
7 Ldap List * "" ""
* 7 192.168.0.132
7 OK
```

Listaccess

Displays the current list of LDAP access profiles (see LDAP Access Profiles on page 303).

Syntax

```
tag Ldap Listaccess pattern start count
```

where:

- *pattern* must currently be * or "" (empty string), meaning all access profiles.
- *start* indicates the first access profile you want to see. The empty string ("") implicitly means 0.
- *count* indicates the number of access profiles you want to see. See the Example below. The empty string ("") implicitly means all access profiles. If *count* is greater than the total number of LDAP servers, list returns as many access profiles as possible.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
15 Ldap Listaccess ““ ““ ““
15 o=top
15 OK Completed
```

Listmailhost

Displays the current Junk Mail Manager mailhost domains.

Syntax

```
tag Ldap Listmailhost pattern start count
```

where `pattern` must currently be * or `""` (empty string) to match all mailhosts, `start` is the first to list, and `count` is the number to display.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
35 Ldap Listmailhost ""
* 35 example.com
35 OK Completed
```

Listobjclass

Displays a list of supported LDAP object class specifications (see LDAP Object Class Specifications on page 303).

Syntax

```
tag Ldap Listobjclass pattern start count
```

where:

- `pattern` must currently be * or `""` (empty string), meaning all object class specifications.
- `start` indicates the first LDAP object class specification you want to see. The empty string (`""`) implicitly means `0`.
- `count` indicates the number of LDAP object class specifications you want to see. See the Example below. The empty string (`""`) implicitly means all object class specifications. If `count` is greater than the total number of object class specifications, `list` returns as many object class specifications as possible.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
17 Ldap Listobjclass
* 17 user:mailsubscriber
* 17 user:person
17 OK Completed
```

Listquarantinehost

Displays the current Junk Mail Manager quarantine manager domains.

Syntax

`tag Ldap Listquarantinehost pattern start count`

where *pattern* must currently be `*` or `""` (empty string) to match all quarantine hosts, *start* is the first to list, and *count* is the number to display.

Privilege Levels

- ◆ Administrator
- ◆ Backup operator

Domain Sensitivity

None

Example

```
36 Ldap Listquarantinehost ""
* 36 example.com
36 OK Completed
```

Listqueries

Responds with a list of supported LDAP query specifications (see LDAP Query Specifications on page 300).

Syntax

`tag Ldap Listqueries pattern start count`

where:

- ◆ *pattern* is currently ignored.
- ◆ *start* indicates the first LDAP query specification you want to see. The empty string (`""`) implicitly means `0`.
- ◆ *count* indicates the number of LDAP query specifications you want to see. See the Example below. The empty string (`""`) implicitly means all query specifications. If *count* is greater than the total number of query specifications, `list` returns as many query specifications as possible.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
7 Ldap Listqueries "" "" ""
* 7 user:mailhost
* 7 user:publishedname
* 7 user:routingaddr
* 7 user:groupmembership
* 7 mailgroup:Members
* 7 mailgroup:Owner
7 OK Completed
```

Search

Allows you to search an LDAP database for entries that match the specified filter. With its *server* argument, `Ldap Search` does not require you to have designated an LDAP server with `Ldap Add`, but it may require `Ldap Addaccess` to the database.

Syntax

`Ldap Search` *server-spec basedn filter* `[` *attrs options* `]`

where:

- *server-spec* identifies the LDAP server you want to search (see LDAP Server Specifications on page 299).
- *basedn* is as described for `Ldap Setquery`.
- *filter* is as described for `Ldap Setquery`.
- *attrs* is as described for `Ldap Setquery`.
- *options* is a quoted combination of any of the following optional parenthesized parameters:

  ```
  "(binddn=dn)(bindpasswd=password)(scope=scp)"
  "(sizelimit=slimit)(timelimit=tlimit)"
  ```

  where:

  - *dn* and *password* are a distinguished name and password to be used for authentication. *password* may start with any of the following prefixes indicating the password encoding (if no prefix is specified, the password is assumed to be plaintext):
    - `{UNIX}`
    - `{MD5}`
    - `{CRYPT}`
  - *scp* is one of:

  - `sub`—search the entire subtree (all descendants) under the specified baseDN.
  - `one`—search only the specified baseDN and its immediate descendants.
  - `base`—search only the specified baseDN itself (no descendants).
- *`slimit`* is the maximum number of matching entries you want to see. This must be a non-negative integer.
- *`tlimit`* is the maximum number of seconds to allow for the search. When this period elapses, the command responds with the entries found so far.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
21 Ldap Search ldap dc=example,dc=com mail=juser@example.com "" ""
* 21 dn: uid=juser,dc=example,dc=com
* 21 objectclass: inetOrgPerson
* 21 objectclass: inetLocalMailRecipient
* 21 uid: juser
* 21 cn: Joe User
* 21 sn: User
* 21 userPassword: secret
* 21 mail: juser@example.com
* 21 mailhost: mail2.example.com
21 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

`tag` `Ldap Set` *`parameter value`*

where:

- *`parameter`* is one of:
  - `Autoprovision`—whether autoprovisioning of user accounts and inboxes is enabled (`ON` or `OFF`). If a user LDAP entry contains attribute `Mailhost` matching this system (see LDAP Query Specifications on page 300) and this option is `ON`, an account and inbox are automatically created the first time a message arrives in the inbox, or the user logs in. Quota is taken from the LDAP attribute `miMailQuota` if present, or set unlimited if not. Some variation of “user does not map” error may result from a mismatch between query specifications and LDAP user entry. With `Conf Enable Ldapforward ON`, users are autoprovisioned if they have a forwarding address. As of release 3.8.0, JMM users are autoprovisioned based on the

LDAP attributes miQuarantinehost and miQuarantineQuota; see User:Quarantineprofile under Testquery.

- Cachetimeout—the period that LDAP query results are cached internally. Specifying a value of 0 disables caching and flushes (empties) the cache. Setting the cache timeout to very small values (for example, 1 to 9 seconds) has a negative impact on system performance. The value may have one of the following suffixes indicating a time unit (if no suffix is specified, the value is assumed to be in seconds):
  - w—weeks
  - d—days
  - h—hours
  - m—minutes
  - s—seconds (the default)
- Compatv1routing—whether LDAP-based message routing is done using version 1 semantics (ON or OFF). System software version 2.5 uses version 2 LDAP routing semantics unless you enable this feature. Deprecated.
- Ldif—LDAP Data Interchange Format (LDIF) information to be stored and used locally for message routing and connection proxying in place of an LDAP database. The value must be a literal string containing LDIF data. To append LDIF, type value-attribute pairs separated by one space and terminated by newline, forming records delineated by a single blank line. Continuation lines may start with a single space or tab. Base64 notation with :: is not supported. To delete all previous LDIF data, merely set a null *value*.
- Localcostable—Whether or not to perform COS lookups against the internal directory server's local routing table. The default is Off. Can be set to On only when Junk Mail Manager is licensed. Useful in conjunction with third party LDAP servers that do not support COS schema.
- Mirabase—the base distinguished name (baseDN) of the location in your LDAP database where Mirapoint-specific configuration data is stored. The empty string ("") means the root of your LDAP namespace. If the value contains a backslash (\) character, you must replace it with a pair of backslash characters (\\) when you enter the value.
- Subdomainroutinglevel—Subdomain dot levels on which to perform domain-based LDAP (not LDIF) routing. The *value* is an integer greater than or equal to –1 giving the number of levels. Null string "" equals –1, which means all levels given. For example if a message arrives addressed to user@a.b.c.d, subdomain routing levels are interpreted as follows:

    - –1 routes to domain `@a.b.c.d` and bounces otherwise
    - 3 routes to domains `@*.b.c.d` and bounces otherwise
    - 2 routes to domains `@*.c.d` and bounces otherwise
    - 1 routes to domains `@*.d` and bounces otherwise
    - 0 routes to domains @* going anywhere if possible
  - `Userdn`—Desupported. Instead, use the `User:*` query specifications to configure LDAP-based login authentication (see LDAP Query Specifications on page 300).
- *`value`* is the value to which you want to set *`parameter`*.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
9 Ldap Set Cachetimeout 30m
9 OK Completed
10 Ldap Set Compatv1routing ON
10 OK Completed
```

Setobjclass

Sets the object class name for an object class specification (see LDAP Object Class Specifications on page 303 to learn about object class specifications).

Syntax

`tag` `Ldap Setobjclass` *`ocspec objclass`*

where:

- *`ocspec`* is one of the pre-defined object class specification names (see LDAP Object Class Specifications on page 303):
  - `User:Person`—The object class to create new LDAP entries for users.
  - `User:Mailsubscriber`—The object class to modify existing LDAP entries for users to subscribe them to Mirapoint messaging services.
- *`objclass`* is the name of an object class understood by your LDAP server (see LDAP Object Class Specifications on page 303 for the requirements for this object class).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
18 Ldap Setobjclass user:Person mirapointUser
18 OK Completed

19 Ldap Setobjclass user:Mailsubscriber mirapointMailUser
19 OK Completed
```

Setquery

Sets the base distinguished name (baseDN), filter, and attribute name for the given query specification. You may use variables such as `$(dcmap)` in both the filter and baseDN arguments. For more details about query specifications, see LDAP Query Specifications on page 300.

Each class has a primary query specification, which controls baseDN and filter for the class. Secondary query specifications are assigned the same baseDN and filter, even if you specify different ones. `User:Publishedname` is the primary query spec for the user class, `mailgroup:Members` is the primary query spec for the mailgroup class, and `user:Groupmembership` is primary for the group membership class.

Syntax

tag `Ldap Setquery` *queryspec basedn filter attr type*

where:

- *queryspec* is one of the pre-defined query specification names (see LDAP Query Specifications on page 300).
- *basedn* is the base distinguished name that specifies the area of LDAP hierarchy in which to search. The area you specify must contain your user records.
- *filter* is an LDAP search filter used to select the LDAP record from which to retrieve the value of *attr*. The format of a search filter is defined by RFC 1960. In general, the filter must match only one record. If it matches multiple records, Mirapoint uses the first record returned, which might have unintended results. For `Mailbox:Mailhost` the query's *basedn* is used without template expansion, so "(objectclass=miFolder)" is the recommended filter. For more information, refer to the *Administrator's Guide.*
- *attr* is the schema name of the LDAP attribute whose value is to be retrieved. But for `Mailgroup:Members`, *attr* is a quoted, space-separated list of members, which can be either Direct or Indirect, as indicated by the *type* parameter. And for `User:Fullname`, it is a quoted, space-separated list of attributes for names, designated Primary and Secondary, as indicated by the *type* parameter.
- *type* is used for the `Mailgroup:Members` and `User:Fullname` queries. You must specify a type, one-to-one, for each member in the *attr* list. Concatenation of multiple attributes is not supported.

- For `Mailgroup:Members` the types are as follows:
  - `Direct`—says corresponding member is an RFC 822 email address.
  - `Indirect`—indicates corresponding member is a distinguished name that can be looked up using the `User:Publishedname` query.
- For `User:Fullname` the types are as follows:
  - `Primary`—indicates primary attribute supplying full name.
  - `Secondary`—indicates secondary attribute supplying full name.
- For `Mailbox:Mailhost` the type is ignored.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
9 Ldap Setquery user:Publishedname c=US
    "(|(mail=$(login))(mailAltAddress=$(login)))" mail ""
9 OK Completed

10 Ldap Setquery user:Publishedname dc=example,dc=com
"(|(mail=$(login))(uid=$(login))(mailLocaladdress=$(login)))" mail ""
10 OK Completed

11 Ldap Setquery user:Mailhost "" "" mailhost ""
11 OK Will use basedn and filter from user:publishedname query

12 Ldap Setquery user:Routingaddr "" "" mailroutingaddress ""
12 OK Will use basedn and filter from user:publishedname query

13 Ldap Setquery user:Fullname "" "" "displayname cn" "primary secondary"
13 OK Will use basedn and filter from user:publishedname query

24 Ldap Setquery mailgroup:Members dc=example,dc=com
"(mail=$(group))" "uniquemember mgrpRfc822mailMember" "indirect direct"
24 OK Completed

25 Ldap Setquery mailgroup:Owner "" "" "owner" ""
25 OK Will use basedn and filter from mailgroup:members query

26 Ldap Setquery user:Groupmembership dc=example,dc=com
"(&(objectclass=mailgroup)(cn=$(groupname)*))" uniquemember ""

t Ldap Setquery mailbox:Mailhost o=mifolders (objectclass=miFolder) mailhost ""
t INFO "Synced 76 of 76 objects"
t OK Completed
```

Testquery

Tests a query specification using supplied arguments. The command responds with a diagnostic message and/or the value of the LDAP attribute retrieved by the query specification (see LDAP Query Specifications on page 300 to learn about query specifications). You can use this command to debug your query specification definitions.

Syntax

tag `Ldap Testquery` *queryspec args*

where:

- *queryspec* is one of the pre-defined query specification names (see LDAP Query Specifications on page 300) or one of the following test diagnostics:
  - `Mailgroup:Authorized`—Verifies authorization for the specified user with `AllowedBroadcaster`, or domain of the user with `AllowedDomain`.
  - `User:Authentication`—Attempts authentication for the specified user with the specified password and responds with the results.
  - `User:Deliveryoption`—Responds with LDAP forwarding information (values of `miForwardingAddress` and `miDeliveryOption`).
  - `User:Quarantineprofile`—Gives a Junk Mail Manager host or address for the user, taken from `miQuarantineHost` and `miQuarantineQuota`.
  - `User:Mailprofile`—Responds with information about the specified user. The response line has the format:

  `*` *tag mailhost routingaddr publishedname loginid fullname quota*

  where:

    - *mailhost* is the value of the attribute referred to by the `User:Mailhost` query specification.
    - *routingaddr* is the value of the attribute referred to by the `User:Routingaddr` query specification.
    - *publishedname* is the value of the attribute referred to by the `User:Publishedname` query specification.
    - *loginid* is the value of the attribute referred to by the `User:Loginid` query specification.
    - *fullname* is the value of the attribute referred to by the `User:Fullname` query specification.
    - *quota* is the value of the attribute referred to by the `User:Quota` query specification.

- *args* is a quoted, space-separated list of arguments appropriate for *queryspec*:
  - `Mailgroup:Authorized` requires one argument containing two elements separated by space: *sender* (an RFC 2822 email address) and *mailgroup* (the name of a group expandable by `mailgroup:Members`).
  - `Mailgroup:Members` requires a single argument: a mail group name.
  - `Mailgroup:Owner` requires a single argument: a mail group name.
  - `User:Authentication` requires two arguments in the format:

  "*login-name password*"

  where:

    - *login-name* is a user's login name.
    - *password* the user's password.

  - `User:Fullname` requires a single argument: an RFC 822 email address.
  - `User:Groupmembership` requires two arguments in the format:

  "*member mailgroup*"

  where:

  – *member* is the RFC 822 email address or mail group name for which you want to search in *mailgroup*.
  – *mailgroup* is the mail group you want to search for *member*.

- `User:Localaddr` requires a single argument: an RFC 822 email address.
- `User:Loginid` requires a single argument: an RFC 822 email address.
- `User:Mailhost` requires a single argument: a fully qualified DNS domain name (host name).
- `User:Mailprofile` requires a single argument: a login name or, alternatively, an RFC 822 email address.
- `User:Publishedname` requires one argument: an RFC 822 email address.
- `User:Quarantineprofile` requires one argument: an RFC 822 address.
- `User:Quota` requires a single argument: an RFC 822 email address.
- `User:Routingaddr` requires a single argument: an RFC 822 email address.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Ldap Testquery user:Mailhost juser@example.com
* 2 mail7.example.com
2 OK Completed
3 Ldap Testquery user:Fullname juser@example.com
* 3 Joseph User
3 OK Completed
4 Ldap Testquery user:Groupmembership "engineer@example.com sales"
4 NO user is NOT a member of group
5 Ldap Testquery user:Groupmembership "engineer@example.com engineering"
5 OK user is a member of group
6 Ldap Testquery mailgroup:Members smallgroup
* 6 user1@example.com
* 6 user2@subdomain.example.com
* 6 user3@example2.com
6 OK 3 addresses
7 Ldap Testquery user:Publishedname bogususer
7 NO LDAP Search failed
8 Ldap Testquery mailgroup:Authorized "user@example.com restricted"
8 NO Not authorized to send to this address
9 Ldap Setquery User:Uuid "" "" miUuid ""
OK Will use basedn and filter from user:publishedname query
9 Ldap Testquery User:Uuid juser
97ce1568-2df4-1027-87f8-00d0b7a95312
```

The License Command

The `License` command lets you view the status of your Mirapoint system software licenses and apply new licenses.

Subcommands

Apply

Applies the specified license to a Mirapoint system. Applying a license usually enables and sometimes starts the service. Some services may be interrupted by this command, especially if a service is somehow related to the license.

The following licensed products are auto-enabled: Antispam, Antivirus (both with `Conf`), Directory Server, and WebMail (both with `Service Enable` and `Start`).

Syntax

```
tag License Apply key
```

where *key* is a license key purchased from Mirapoint Software, Inc.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 License Apply 2aP1AjEZeha
* 2 User-limit unlimited
2 OK Completed
```

Count

Counts the number of active licenses on a Mirapoint system.

Syntax

```
tag License Count pattern
```

where *pattern* must be "" or *, both of which match everything.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 License Count *
* 4 3
4 OK
```

Fetch

Attempts to automatically download and apply license keys. Internet connectivity is required: license keys are downloaded from the http://license.mirapoint.com site, based on hostID. If a license has not yet been applied to the system, `Fetch` does so. All keys on the Mirapoint license server are encrypted for security. This command honors settings of `Conf Httpproxy`.

Syntax

```
tag License Fetch
```

Privilege Levels

Administrator

Domain Sensitivity

Returns an error if a delegated domain is current.

Example

```
3 License Fetch
* 3 Junkmail Manager unlimited users
* 3 Sophos Antivirus unlimited users
* 3 Signature Edition Rapid Antispam unlimited users
3 OK Completed
```

Hostid

Responds with the unique identifier for your Mirapoint system. When you purchase a license, your Mirapoint customer service representative asks you for this identifier to generate a unique license key for your system.

Syntax

tag `License Hostid`

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 License Hostid
* 6 00a0c9adfe6b
6 OK Completed
```

List

Responds with details about active licenses on a Mirapoint system. There are three output columns: the license key, a descriptive name similar to what `License Status` shows, and the license expiration date (if any).

Syntax

tag `License List` *pattern start count*

where:

- *pattern* must be "" or *, both of which match everything.
- *start* is the first match to display. The empty string ("") means `0`.
- *count* is the number of matches to show. The empty string ("") means all.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 License List "" "" ""
* 5 whKyyTovjW0 "System OPERATING" ""
* 5 lWKRj8ntt+1 "User-limit unlimited" ""
* 5 MlVGFoQQfKb "SSL (strong encryption)" ""
* 5 MS2x51jRWZ9 "SSH Licensed" ""
* 5 eQAPolAuxte "Mail Routing" ""
* 5 SJTTE9ljQCa "Upgrades Allowed" ""
* 5 x4g1PSHOEv4 "Corporate Edition unlimited users" ""
* 5 Ab1fXoc8sM6 "Delegated Domain Administration" ""
* 5 Kfa6E-7s1A2 "WebMail Direct Standard Edition unlimited users" ""
* 5 TTkeeCbZw9f "POP unlimited users" ""
* 5 DPJSEIDrws9 "IMAP unlimited users" ""
* 5 DZVqAqU7x1d "Directory Server Access unlimited users" ""
* 5 aQoLN33pjpb "WebCal Direct Personal Edition unlimited users" ""
* 5 yzFyQgwB+V3 "GroupCal Direct Standard Edition unlimited users" ""
* 5 h9EypDqqmgf "SMTP FastPath" ""
* 5 7GCa3TwZDva "XML unlimited users" ""
* 5 xhV4ZULTPm7 "Sophos Antivirus unlimited users" ""
* 5 7TepzyRuelf "Principal Edition Antispam unlimited users" ""
* 5 EPfhSuHJq+e "Message Server" ""
* 5 Y6kPpDdFHJ4 "Operations Console" ""
5 OK Completed
```

Override

Applies a license from a different system, temporarily overriding the checks that would ordinarily prevent this. This is necessary only for completing a system recovery operation from backups onto a spare system. The license is applied as a temporary 30-day license. This allows sufficient time for you to obtain from Mirapoint new license keys for your new main system unit.

Syntax

tag `License Override` *key hostid*

where:

- *key* is a license key for the system identified by *hostid*.
- *hostid* is the host ID of the system on which the license was originally installed.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
7 License Override 2aP1AjEZeha mail1
* 7 User-limit unlimited
7 OK Completed
```

Revoke

Revokes the specified license.

Syntax

tag `License Revoke` *key*

where *key* is the license key identifying the license you want to revoke. This is the same key used to apply the license using `License Apply`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 License Revoke 1-QQYIOWbte
8 OK Completed
```

Status

Responds with a list of active licenses on your Mirapoint system.

Syntax

```
tag License Status
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 License Status
* 4 System OPERATING
* 4 User-limit unlimited
* 4 SSL (weak encryption)
* 4 SSL (strong encryption)
* 4 SSH Licensed
* 4 Mail Routing
* 4 Upgrades Allowed
* 4 Junkmail Manager 20 users
* 4 Signature Edition Rapid Antispam 5000 users
* 4 F-Secure Antivirus 499 users
* 4 Corporate Edition unlimited users
* 4 WebMail Direct Standard Edition unlimited users
* 4 POP unlimited users
* 4 IMAP unlimited users
* 4 WebCal Direct Personal Edition unlimited users
* 4 GroupCal Direct Standard Edition unlimited users
* 4 SMTP FastPath
* 4 XML unlimited users
* 4 Sophos Antivirus unlimited users
* 4 RazorGate
4 OK Completed
```

32

The Locale Command

The `Locale` command sets the system **locale**, which specifies a translated version of the system's web interfaces appropriate for a particular geographic region. Locales issued for current Mirapoint releases are:

In a multi-tier installation, desired locales must be installed on all message servers and connection proxies.

- en_US.ISO_8859-1 (US English)
- de_DE.utf-8 (German)
- es_MX.utf-8 (Universal Spanish
- fr_FR.utf-8 (French)
- ja_JP.utf-8 (Japanese)
- zs_CN.utf-8 (Simplified Chinese)
- zt_TW.utf-8 (Traditional Chinese)

Locales

Each locale is identified by a unique, case-insensitive, US-ASCII string, which has the following format, where:

la-RE.charset

- *la* is a two-character code identifying the human language for the locale, such as `en` for English.
- *RE* is a two-character code identifying the geographic region for the locale, such as US for the United States. Though case is not significant, by convention, this code is usually given in upper case.
- *charset* is the unique identifier for the character set to be used for displaying text, such as `ISO-8859-1` for the ISO Latin 1 character set, used in Western Europe and the Americas.

The only locale string that is always supported is `en-US.iso-8859-1`. Other locales are supported only if they are installed on a system. For more information, contact your Mirapoint sales representative or reseller.

Subcommands

Count

Responds with the number of available locales (see Locales on page 335).

Syntax

tag `Locale Count` *pattern*

where *pattern* is currently ignored.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
1 Locale Count ""
* 1 1
1 OK Completed
```

List

Responds with a list of supported locales (see Locales on page 335).

Syntax

tag `Locale List` *pattern start count*

where all parameters are currently ignored.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
2 Locale List "" "" ""
* 2 en_US.ISO_8859-1
2 OK Completed
```

Get

Responds with the value of the specified parameter.

Syntax

`tag Locale Get parameter`

where *parameter* may be:

- `Default`—Refers to the default locale (see Locales on page 335).
- `Loginfooter`—Refers to the list of language translation locales appearing on the login page for WebMail Direct or Administration Suite. It may be on or off.
- `Unannounced`—The locale that filtering uses as a guess for unannounced text. Here are the current mappings:
  - `ko_KR.utf-8` becomes EUC-KR
  - `zs_CN.utf-8` becomes GB2312
  - `zt_TW.utf-8` becomes Big5
  - `ja_JP.utf-8` becomes ISO-2022-JP

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
6 Locale Get Default
* 6 en_US.ISO_8859-1
6 OK Completed
```

```
7 Locale Get Unannounced
* 7 zs_CN.utf-8 GB2312
7 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

`tag Locale Set parameter value`

where *parameter* may be:

- `Default`—Refers to the locale you wish to set as default (see Locales on page 335). For a list of installed locales, run `Locale List`.
- `Loginfooter`—Refers to the list of language translation locales appearing on the login page for WebMail Direct and Administration Suite. You can set it `on` to allow users to choose a different locale, or `off` to force use of the default locale.
- `Unannounced`—Sets the locale that filtering should use when it encounters unannounced (unlabeled) text; *`value`* should be the name of a locale. To disable this feature, specify the empty string, which indicates no locale. Character set guessing occurs when filtering encounters a message containing multi-byte data for which no character set was marked. This is meaningful for Asian locales ko_KR.utf-8, zs_CN.utf-8, zt_TW.utf-8, or ja_JP.utf-8. The filtering code (both user and domain) treats the message part as if it were marked for the character set commonly used in that locale. For headers it only does this if muti-byte data is valid for the locale.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Locale Set Default ja_JP.utf-8L2_7_1-1
3 OK Completed

4 Locale Set Loginfooter on
4 OK Completed

5 Locale Set Unannounced zs_CN.utf-8
5 OK Completed
```

The Log Command

The `Log` command controls the Mirapoint logging facility, by routing **events** into **output methods**. Event information includes subsystem, type, subtype, origination time, processed time, sequence number, and an event-specific message. Output methods include `Syslog`, which produces UDP data in system log format, and `Watch`, which produces data for interactive viewing. Events are logged right after they occur, with a short delay for buffering and system load. On most servers event data lasts seven days.

For example: to monitor bad WebMail logins, issue the following commands:

```
log addroute "" webmail.user.badlogin WATCH ""
log watch webmail.user.badlogin NEW
```

To provide UDP notification of all newly received messages to remote host `mailA`'s port 512, issue the following command. On `mailA`, Mirapoint professional services can provide a port-listening program that notifies users of received mail.

```
log addroute YGM mta.mailbox.newmail SYSLOG mailA:512
```

Data Formats

Event data are serialized into a simple string representation for transmission. Fields in a packet are whitespace delimited. Two adjacent whitespace characters indicate missing data. The first four fields comprise the header. All fields, including numeric, are recorded in ASCII. The header fields are:

- Sequence number—Number uniquely identifying each packet. It is a steadily increasing number with 64 bit precision. Though sequence number is arbitrary (not guaranteed to be contiguous) it is guaranteed to order log messages. Using the `Log Watch` command, a sequence number may be used to replay previously recorded logs, allowing for reliable transmission of logs to remote systems.
- Host name—name of the system on which this packet was logged.
- Origination time—Time in seconds with a millisecond fraction since midnight January 1st 1970 GMT.
- Identifier—A triple of the form *`SUBSYSTEM.TYPE.SUBTYPE`*. See the Identifiers section for details.
- Event-specific data—See the Identifiers section for descriptions of these fields.

Data fields are separated by Tab characters. The string as a whole ends with CRLF. The sequence number is represented as an 8-byte integer. The host name is that of the sending machine at time of transmission. The identifier is represented as shown below. Event-specific data are given as a sequence of whitespace-separated strings.

Identifiers

Client subsystems (including ADMIN, IMAP, POP, CALENDAR, WEBMAIL, and XMLCALENDAR) use the following identifiers:

- *SUBSYSTEM*.USER.BADLOGIN—A login has failed. After identifier, fields are:
  - From—Transport type (clr, ssl, ssh, tls) and IP address [in brackets].
  - User—The name of the user who attempted to login.
  - Authentication Type—The type of password authentication used.
  - Reason—A description of the failure.
- *SUBSYSTEM*.USER.LOGIN— A login has succeeded. After identifier, fields are:
  - From—Transport type (clr, ssl, ssh, tls) and IP address [in brackets].
  - User—The name of the user who logged in.
  - Authentication Type—The type of password authentication used.
  - Reply—Any additional information.
  - Optional—UserAgent information, for CALENDAR only.
- *SUBSYSTEM*.USER.LOGOUT—A user has logged out. After identifier, fields are:
  - From—Transport type (clr, ssl, ssh, tls) and IP address [in brackets].
  - User—The name of the user who logged out.
  - Authentication Type—The type of password authentication used.
  - Time—The length, in milliseconds, of the connection.

  Note: if all fields are empty, it indicates that all users have been logged out.

The POP service uses identifier POP.USER.MINPOLL, meaning a user has connected to POP more than once during the minimum poll period (see Pop Set Minpoll).

Administration Service

The Administration service uses these identifiers:

- ADMIN.PROTOIN.COMMAND—Administration command (inbound connection). Not all administration commands are logged. Stat Get occurs so frequently it would fill up the logs, and Log Diagwatch is hidden. After identifier, fields are:
  - SessionID—Session number, actually the *admind* process ID.
  - User—Name of the user executing this command.
  - Command—Tagged command.
- ADMIN.PROTOIN.RESPONSE—Administration command response to above. Replaces output side of the former admind logs. Fields same as for COMMAND.
- ALERT.*ITEM.STAT*—Notification of an alert condition, or cleared alert, one log event per alert. The name *ITEM.STAT* follows the Stat Get naming of alerts. After identifier, fields are:
  - State—New, Outstanding, or Cleared.
  - Reason—The cause of the alert.

Calendar Service

The Mirapoint calendar service uses these identifiers:

- `CALENDAR.USER.NEWEVENT`—A new event has been created in the user's calendar. After identifier, fields are as follows:
  - User—User name of the calendar affected.
  - EventId—Mirapoint event ID of the event affected. The format is the same as used in the CAP protocol: `mcal:`*`eventid@hostname`*
  - Version—Always 0 for new events. The version number is incremented with each subsequent change to the event in the Mirapoint calendar store (see other message types below).
  - Source—A string that identifies what caused the event to be created. Some interfaces to the calendar allow the client to specify the source string, so this may be any valid string "x-" at the beginning. These string values are reserved for internal Mirapoint applications:
    - `web`—the HTML interface.
    - `admind`—the Administration protocol.
    - `xml1`—XML version 1, if the client did not specify a source.
    - `mcap`—Mirapoint Calendar Access protocol.
  - Sourceuser—The login ID of the user that caused the change to occur.
- `CALENDAR.USER.EVENTCHANGE`—An existing event has been modified. Fields are the same as for `CALENDAR.USER.NEWEVENT`.
- `CALENDAR.USER.EVENTDELETE`—An existing event has been deleted from the calendar store. Fields are the same as for `CALENDAR.USER.NEWEVENT`.

Mail Storage Subsystem

The mail storage subsystem uses these identifiers:

- `MAILBOX.COPY.STATUS`—One mailbox has been copied to another. After identifier, fields are as follows, and include optional flags:
  - Copier—Name of user@domain who initiated the copy.
  - Hostname—Remote hostname from `Mailbox Copy` command.
  - Mailbox—Remote mailbox from `Mailbox Copy` command.
  - User—Name of remote user whose mailbox was copied.
  - Destination—Local mailbox name from `Mailbox Copy` command.
  - Status—Returned from `Mailbox Copy` command.
- `MAILBOX.FILTER.ACTION`—Notification that a user filter action was triggered. After identifier, fields are:
  - User—User of the filter in question.
  - Sender—The envelope from address for the message.
  - QID—Queue ID to uniquely identify the message.
  - Msg-ID—SMTP message ID contained in header (generated if necessary).
  - Filter—Filter name that fired.
  - Action—Filter action that took place.
  - Argument—Literal argument to filter command, `INBOX` or `Fwdexcerpt`.

- MAILBOX.MESSAGE.APPEND—A message has been appended to this mailbox. After identifier, fields are:
    - Mailbox—Name of the mailbox being operated on.
    - User—Name of user doing the append.
    - UID—POP or IMAP user ID.
    - Msg-ID—SMTP message ID contained in header (generated if necessary). The RFC 822 message header, used to track messages between systems.
- MAILBOX.MESSAGE.COPY—Indicates when a message is copied to another folder, either by IMAP's COPY command or by XML, WebMail, and so forth mail copy. One log is issued per message. After identifier, fields are:
    - User—Name of user doing the copy.
    - SrcMbox—Source mailbox name.
    - SrcUID—Source mailbox UID validity, as in RFC 2060.
    - SrcMsg—Source message UID.
    - DestMbox—Destination mailbox name.
    - DestUID—Destination mailbox UID validity.
    - DestMsg—Destination message UID.
- MAILBOX.MESSAGE.EXPUNGE—A message was expunged from this mailbox. After identifier, fields are:
    - Mailbox—Name of the mailbox being operated on.
    - User—Name of user doing the expunge.
    - UID—POP or IMAP user ID.
    - Msg-ID—SMTP message ID contained in header (generated if necessary).
- MAILBOX.MESSAGE.FLAGSET—A user has changed a flag on a message to the "set" state. After identifier, event-specific fields are:
    - Mailbox—Name of the affected mailbox.
    - User—Full login name of the user who set the flag.
    - UID—The IMAP UID for which the flag was set.
    - Validity—The UID validity for the mailbox, as described by RFC 2060.
    - Flagname—The name of the flag as described by RFC 2060.
- MAILBOX.MESSAGE.FLAGUNSET—A user has changed a flag on a message to the "unset" state. After identifier, event-specific fields are:
    - Mailbox—Name of the affected mailbox.
    - User—Full login name of the user who unset the flag.
    - UID—The IMAP UID for which the flag was cleared.
    - Validity—The UID validity for the mailbox, as described by RFC 2060.
    - Flagname—The name of the flag as described by RFC 2060.
- MAILBOX.MESSAGE.IMAPFETCH—A user has issued an IMAP FETCH command on a message. After identifier, event-specific fields are:
    - Mailbox—Name of the affected mailbox.
    - User—Full login name of the user who requested the fetch.
    - Reserved—For future use.
    - Validity—The UID validity for the mailbox, as described by RFC 2060.
    - Sequence—The IMAP sequence describing which messages were fetched.

    - Argument—Parameter passed to the IMAP `FETCH` command (1 KB limit).

- `MAILBOX.MESSAGE.OVERQUOTANOTIFY`—An over-quota notification was sent to Mailbox after email delivery failed. After identifier, event-specific fields are:
    - Mailbox—Name of the affected mailbox.
    - QuotaRoot—Mailbox's quota root.
    - QuotaUsed—Mailbox space currently in use (in 1 KB blocks).
    - QuotaTotal—Quota assigned to the mailbox (in 1 KB blocks).
- `STORE.MAILBOX.ADD`—A mailbox was created. Event specific flags are:
    - Mailbox—Name of the created mailbox.
    - User—Creator or @domain (+*N* prepended for autoprovisioning).
    - Reserved—For future use.
    - Validity—The UID validity for the mailbox, as described by RFC 2060.
- `STORE.MAILBOX.DELETE`—A mailbox was deleted. Event specific flags are:
    - Mailbox—Name of the deleted mailbox.
    - User—Login name of user deleting the mailbox.
    - Reserved—For future use.
    - Reserved—The UID validity cannot be determined for deletes.
- `STORE.MAILBOX.PUBLISH`—A mailbox was published into the `miFolders` DIT of the LDAP database. Event specific flags are:
    - Mailbox—Name of the published mailbox.
    - Domain—Domain of the mailbox, or `Primary` for top-level domain.
    - Timestamp—Clock time when the mailbox was queued for publishing.
    - Server— LDAP server where write was attempted; could be LDAP URL.
    - DN—The distinguished name of a folder for the attempted write.
    - Status—Operation status:
        - "Completed successfully" if the record was published to LDAP.
        - If not, a standard message such as "Error: Permission denied".
- `STORE.MAILBOX.RENAME`—A mailbox was renamed. Event specific flags are:
    - Mailbox1—Name of the mailbox before rename.
    - Mailbox2—New name of the mailbox after rename.
    - User—Full login name of the user.
    - Reserved—For future use.
    - Validity—The UID validity for the mailbox, as described by RFC 2060.

Mail Transfer Agent

The mail transfer agent (SMTP) uses these identifiers:

- `ANTIVIRUS.`*`VENDOR`*`.SCANFAILURE`—A scan failure occurred, so a header was added to the given message. After identifier, event-specific fields are:
    - QID—Queue ID to uniquely identify the message.
    - Msg-ID—SMTP message ID contained in header (generated if necessary).
    - Reason—A description of the failure.
    - MIME—Attachment index, for example 1.2 or 2.1.1.

  - ❖ AttachName—Attachment name, according to the MIME headers.
  - ❖ PeerAddr—The source machine.
  - ❖ EnvFrom—Message sender, according to the envelope.
  - ❖ EnvTo—Message recipients, according to the envelope.

- `ANTIVIRUS.`*`VENDOR`*`.VIRUSFOUND`—Virus was found and infected attachment was passed on to recipient. After identifier, event-specific fields are:
  - ❖ QID—Queue ID to uniquely identify the message.
  - ❖ Msg-ID—SMTP message ID contained in header (generated if necessary).
  - ❖ Virus Name—Generally accepted name for this virus.
  - ❖ MIME—Attachment index, for example 1.2 or 2.1.1.
  - ❖ IP connection—IP address from which the message arrived.
  - ❖ Sender—Originator of this contaminated message.
  - ❖ Recipient(s)—Intended recipient or recipients of this message.
- `ANTIVIRUS.`*`VENDOR`*`.VIRUSCLEANED`—Virus was found and the infected attachment was purged of virus. Event-specific fields same as for `VIRUSFOUND`.
- `ANTIVIRUS.`*`VENDOR`*`.VIRUSDELETED`—Virus was found and the infected attachment was removed. Event-specific fields same as for `VIRUSFOUND`.
- `ANTIVIRUS.`*`VENDOR`*`.VIRUSQUARANTINED`—Virus was found and forwarded to the quarantine address. Event-specific fields same as for `VIRUSFOUND`.
- `FILTER.TAGGED.DOMAIN`—Extra header was inserted by domain filter action. After identifier, fields are:
  - ❖ Domain—Domain of the filter in question.
  - ❖ Msg-ID—SMTP message ID contained in header.
  - ❖ Field—Arbitrary header field specified by (`extraheader=`).
  - ❖ Value—Header field value specified by (`extraheader=`).
- `FILTER.TAGGED.USER`—Extra header was inserted by a user filter action. Event-specific fields as for `TAGGED.DOMAIN`, except User instead of Domain.
- `MTA.FILTER.ACTION`—Notification that a domain filter action was triggered. After identifier, fields are:
  - ❖ Domain—Domain of the filter in question.
  - ❖ Sender—The envelope from address.
  - ❖ Recipients—A space-separated list of the envelope recipients.
  - ❖ QID—Queue ID to uniquely identify the message.
  - ❖ Msg-ID—SMTP message ID contained in header (generated if necessary).
  - ❖ Filter—Filter name that fired.
  - ❖ Action—Filter action that took place.
  - ❖ Argument—Literal argument to filter command, `INBOX` or `fwdexcerpt`.
- `MTA.GETMAIL.FAILURE`—Either remote mailbox check or message download failed. After identifier, event-specific fields are:
  - ❖ User—Mirapoint user name currently being serviced.
  - ❖ Nickname—Host name of remote server being checked for messages.
  - ❖ Messages—Number of messages successfully downloaded.
  - ❖ Status—OK or error condition of this fetch.

- `MTA.GETMAIL.SUCCESS`—The remote mailbox check and message download have succeeded. After identifier, event-specific fields are same as for `FAILURE`.
- `MTA.MAILBOX.ADDINDEX`—A mailbox has been locally indexed. The field is:
  - Mailbox—Name of the newly indexed mailbox.
- `MTA.MAILBOX.ALLSEEN`—A user has retrieved, expunged, or set the `\Seen` flag for all unseen messages in a mailbox. After identifier, event-specific fields are:
  - Mailbox—Name of the affected mailbox.
  - User—Full login name of the user.
  - UID—The last message (highest UID) for which the `\Seen` flag was set.
  - Validity—The UID validity for the mailbox, as described by RFC 2060.
- `MTA.MAILBOX.CHECK`—A user has cleared the `\Recent` flag from all messages in a mailbox. After identifier, event-specific fields are:
  - Mailbox—Name of the affected mailbox.
  - User—Full login name of the user.
  - UID—Last message (highest UID) for which a `\Recent` flag was cleared.
  - Validity—The UID validity for the mailbox, as described by RFC 2060.
- `MTA.MAILBOX.COPY`—A messages was copied to another folder, either by the IMAP `COPY` command or by WebMail or XML mail copy. One line appears per message. After identifier, event-specific fields are:
  - User—Full login name of the user doing the copy.
  - Mailbox—Name of the source mailbox.
  - UID—The UID of the source mailbox, as in RFC 2060.
  - Validity—UID validity of the source mailbox, as in RFC 2060.
  - Mailbox—Name of the destination mailbox.
  - UID—The UID of the destination mailbox, as in RFC 2060.
  - Validity—UID validity of the destination mailbox, as in RFC 2060.
- `MTA.MAILBOX.DELETE`—A user deleted a mailbox. After identifier, fields are:
  - Mailbox—Name of the affected mailbox.
  - User—Full login name of the user.
  - Reserved—This field is reserved for future use.
  - Validity—The UID validity for the mailbox, as described by RFC 2060.
- `MTA.MAILBOX.DELINDEX`—A mailbox index has been deleted. The field is:
  - Mailbox—Name of the formerly indexed mailbox.
- `MTA.MAILBOX.EMPTY`—A mailbox has become empty. After identifier, event-specific fields are:
  - Mailbox—Name of the empty mailbox.
  - User—Full login name of the user who emptied the mailbox.
  - UID—The last message (highest UID) that the mailbox had used.
  - Validity—The UID validity for the mailbox, as described by RFC 2060.
- `MTA.MAILBOX.NEWMAIL`—A new message has arrived in a mailbox. After identifier, event-specific fields are:

- Mailbox—Name of the affected mailbox.
- Reserved—This field is reserved for future use; maybe user name.
- UID—The UID of the new message.
- Validity—The UID validity for the mailbox, as described by RFC 2060.
- Msg-ID—SMTP message ID contained in header (blank if not present).

- `MTA.MESSAGE.FILTERED`—Indicates that a particular message has already been filtered upon arrival. After identifier, event-specific fields are:
  - QID—Queue ID to uniquely identify the message.
  - Filter—Comma-separated list of filters; see `MTA.MESSAGE.REMOTEFILTER`.
  - HostType—System that filtered the message, usually Mirapoint.
- `MTA.MESSAGE.HEADERS`—A message has been received by SMTP with the following headers. It might not be delivered yet. The Received headers are filtered out. The individual headers are Tab separated; internal Tabs and newlines are translated to spaces. After identifier, event-specific fields are:
  - QID—Queue ID to uniquely identify the message.
  - Header fields (exclusive of `Received:` lines) with value
  - More header fields with their values
- `MTA.MESSAGE.LOCAL`—A message has been delivered to a local mailbox. After identifier, event-specific fields are:
  - QID—Queue ID to uniquely identify the message.
  - From—The envelope "from" address.
  - To—Space-separated list of recipients handled by this delivery.
  - Size—Message size in bytes before inserting `Received` and other headers.
  - Attachment—1 if there is at least one attachment, 0 otherwise.
- `MTA.MESSAGE.QUEUED`—A message has been queued for later delivery. After identifier, event-specific fields are:
  - QID—Queue ID to uniquely identify the message.
  - To—Space-separated list of recipients.
- `MTA.MESSAGE.RECEIVED`—A message has been received for delivery by SMTP. After identifier, event-specific fields are:
  - QID—Queue ID to uniquely identify the message.
  - From—The envelope "from" address.
  - Msg-ID—SMTP message ID contained in header (generated if necessary).
  - Remote—Host name of remote machine that sent the message.
  - Encryption—Either `(TLS)` indicating secure connection, or `()` indicating the message was received as cleartext.
  - Size—Message size in bytes before inserting `Received` and other headers.
  - Recipients—How many addresses this message was sent to.
- `MTA.MESSAGE.REMOTE`—A message has been delivered to a remote machine. After identifier, event-specific fields are:
  - QID—Queue ID to uniquely identify the message.
  - From—The envelope "from" address.
  - To—Space-separated list of recipients handled by this delivery.

- Status—Additional information returned by remote machine. This field is free-form, diagnostic, and intended for human interpretation.
- Size—Message size in bytes before inserting `Received` and other headers.
- Attachment—1 if there is at least one attachment, 0 otherwise.
- Transport—CLR for cleartext or TLS for SSL-encoded

◆ `MTA.MESSAGE.REMOTEFILTER`—A message has been transferred to a remote filter host. After identifier, event-specific fields are:

- QID—Queue ID to uniquely identify the message.
- From—The envelope "from" address.
- To—Space-separated list of recipients handled by this delivery.
- Status—Additional information returned by remote machine.
- Size—Message size in bytes before inserting `Received` and other headers.
- Attachment—1 if there is at least one attachment, 0 otherwise.
- Filter—Comma-separated list of filters, abbreviated as follows:
  - AV—Antivirus scanning: Fs = F-Secure, So = Sophos.
  - AS—Antispam scanning: Ct = Signature Edition Rapid Antispam, Sa = Principal Edition Antispam.
  - DS—Domain signature.
  - DF—Domain filtering.
  - QN—Quarantine.
  - WL—Whitelisting.
  - SS—Spam in Subject, see AS above.
  - An action code follows each: N=not allowed by COS, A=already done, D=default, S=sender triggered, R=recipient triggered

◆ `MTA.MESSAGE.REMOVEDATTACHMENTS`—Message had its attachments stripped. After identifier, event-specific fields are:

- QID—Queue ID to uniquely identify the message.
- Who—Filter owner (user or domain) if available.
- Filter—Name of the filter that caused removal.
- What—Name(s) of the attachments removed.

◆ `MTA.MESSAGE.STATUS`—Something unusual happened. After identifier, event-specific fields are:

- QID—Queue ID to uniquely identify the message.
- From—The envelope "from" address if appropriate
- To—Space-separated list of recipients if applicable.
- Info—What happened, for example one of the following error messages:
  - RBL server rejected connection—due to RBL server query.
  - Recipient does not exist—due to SMTP `Recipientcheck`.
  - Sender blacklisted—due to `Uce Addexception` blacklist.
  - Too many inbound SMTP connections—due to connection throttling.
  - UCE blacklist triggered—due to `Uce Add` block (reject list).

◆ `MTA.USER.AUTH`—SMTP authentication succeeded. The field is:

- User—Who the remote sender (user and host name) claimed to be.

◆ `MTA.USER.BADAUTH`—SMTP authentication (AUTH) failed. The field is:

  - User—Who the remote sender (user and host name) claimed to be.

- `MTAVERIFY.RECIPIENT.DELAY`—The SMTP service chose to delay a message with this (probably unknown) triplet. After identifier, fields are:
  - Host—The peer IP address of the originating mail server.
  - MailFrom—Envelope sender address for the message.
  - RcptTo—Envelope recipient address for the message.
- `MTAVERIFY.RECIPIENT.EXEMPTED(`*`diag`*`)`—The message was exempted from MailHurdle processing, for *`diag`* reason.
- `MTAVERIFY.SERVER.DOWN`—The MailHurdle server is not responding. Fields:
  - Host—Name of the MailHurdle server.
  - Reason—A description of why, if available.
- `MTAVERIFY.SERVER.UP`—The MailHurdle server is responding. Fields are:
  - Host—Name of the MailHurdle server.
  - Null—Empty field.
- `UCE.MESSAGE.BLACKLIST`—Sender is black listed. After identifier, fields are:
  - Domain/User—Whether message was filtered at the domain or user level.
  - Name—Name of the domain or user.
  - QID—Queue ID to uniquely identify the message.
  - Msg-ID—SMTP message ID contained in header (generated if necessary).
  - Sender—Originator of the message.
- `UCE.MESSAGE.JUNKMAIL`—This message was either scored as spam or was previously domain-level blacklisted. After identifier, fields are:
  - QID—Queue ID to uniquely identify the message.
  - MailFrom—Envelope sender address for the message.
  - RcptTo—Envelope recipient address, or space-separated addresses.
  - Host—The peer IP address of the originating mail server.
  - Source—Which facility was responsible for logging this event.
  - Score—Junkmail score or <NA> for blacklisting.
  - Msg-ID—SMTP message ID contained in header (generated if necessary).
  - Domain—Either blank, or gives the domain recognizing this junk mail.
- `UCE.MESSAGE.SCORE`—Junk mail recognized score. After identifier, fields are:
  - Domain—Either blank, or gives the domain recognizing this junk mail.
  - QID—Queue ID to uniquely identify the message.
  - Msg-ID—SMTP message ID contained in header (generated if necessary).
  - Sender—Originator of the message envelope.
  - Score—A positive number indicating degree of junkiness.
  - Recipients—A space-separated list of To or Cc addresses.
- `UCE.MESSAGE.WHITE`—Sender is white listed. After identifier, fields are:
  - Domain/User—Whether message was filtered at the domain or user level.
  - Name—Name of the domain or user.
  - QID—Queue ID to uniquely identify the message.
  - Msg-ID—SMTP message ID contained in header (generated if necessary).

  - Sender—Originator of the message.

- `UCE.MESSAGE.WHITEIP`—Sender is white listed by IP address rather than domain name. Fields are the same as above, with IPaddr instead of Name.
- `UCE.MESSAGE.WHITETO`—Recipient is white listed. Fields are the same as for `UCE.MESSAGE.WHITE`.
- `UCE.RECIPIENT.BLACKLIST`—The SMTP service rejected this message due to `Smtp Set Uceblacklist`. After identifier, fields are:
  - Host—The peer IP address of the originating mail server.
  - MailFrom—Envelope sender address for the message.
  - RcptTo—Envelope recipient address for the message.

Mirapoint Directory Server

The Mirapoint directory server (the `Dir` command) uses these identifiers:

- `DIR.SERVER.AUTH`—LDAP database authentication was successful. After the identifier, fields are:
  - Msg—A fixed text string indicating the authentication type.
  - DN—Distinguished name, from LDAP **Bind** request for this connection.
  - Host—The IP address of the client system, from socket peer connection.
- `DIR.SERVER.BADAUTH`—LDAP database authentication failed. After identifier, fields are:
  - Msg—A fixed text string indicating the specific error.
  - DN—Distinguished name, from LDAP **Bind** request for this connection.
  - Host—The IP address of the client system, from socket peer connection.
- `DIR.SERVER.ERROR`—Some problem occurred while the LDAP server was running. After the identifier, fields are:
  - Msg—A fixed text string indicating the specific error.
  - Info—Additional error information.
- `DIR.SERVER.INDEX`—Indexing of the LDAP server. After identifier, fields are:
  - Attribute—The name of the attribute being indexed.
  - Match—Type of matching for this attribute index.
  - Msg—A fixed text string indicating the index status.
- `DIR.SERVER.PROTOCOL`—LDAP server protocol. After identifier, fields are:
  - Conn-ID—The numeric connection ID.
  - Msg-ID—The numeric message ID.
  - Msg—The textual LDAP protocol message type.
  - Args—String containing optional arguments for the message.
- `DIR.SERVER.STATUS`—Status of the LDAP server. After identifier, fields are:
  - Msg—A fixed text string indicating the specific error.
  - Info—Additional status information.
- `DIR.REPLICATION.DEBUG`—Information about LDAP database replication to help in resolving problems. After identifier, fields are:

    - Msg—A fixed text string indicating the specific error.
    - Replica—Name of a replication agreement and URL of the slave system, in the format `name:url`.
    - Info—Extended error information.

- `DIR.REPLICATION.ERROR`—Some problem with LDAP database replication occurred. After identifier, fields are:
    - Msg—A fixed text string indicating the specific error.
    - Replica—Name of a replication agreement and URL of the slave system, in the format `name:url`.
    - Info—Additional error information.
- `DIR.REPLICATION.STATUS`—Status of replication. After identifier, fields are:
    - Msg—A fixed text string indicating the specific error.
    - Replica—Name of a replication agreement and URL of the slave system, in the format `name:url`.
    - Info—Additional status information.

Diagnostics Subsystem

The `Diagwatch` subcommand uses these identifiers:

- `DIAG.FILTER.STATUS`—Indicates the status of a particular group of recipients for a particular message. For the last three fields, AS means antispam, AV means antivirus, DF means domain filters, DS domain signatures, and QN quarantine. Fields are:
    - QID—Queue ID to uniquely identify the message
    - Msg-ID—The numeric message ID.
    - Recipient(s)—Intended recipient or recipients of this message.
    - Total—Comma-separated list of filtering types the message should get.
    - Have—Comma-separated list of filtering types the message has received.
    - Need—Comma-separated list of filtering types the message still requires.
- `DIAG.HTTPPROXY.LOGFAILURE`—A web connection attempt failed. This is logged only if a user could not be found in LDAP. After the identifier, fields are:
    - From—Transport type (`ssl` or `clr`) and IP address [in brackets].
    - User—The user for whom the LDAP lookup failed.
    - Location—The path portion of the request's URL.
    - AuthType—Empty because the proxy does not authenticate.
    - Reason—A description of the failure.
- `HTTPPROXY.USER.LOGINREQUEST`—A web login request was transferred. After the identifier, fields are:
    - From—Transport type (`ssl` or `clr`) and IP address [in brackets].
    - User—The user found in LDAP.
    - Location—The path portion of the request's URL.
    - AuthType—Empty because the proxy does not authenticate.
    - ServerUser—The name of the user logged into the back-end server.
    - Server—Destination back-end server for the request.

- `HTTPPROXY.USER.LOGOUTREQUEST`—A web logout request was transferred. After the identifier, fields are:
  - From—Transport type (`ssl` or `clr`) and IP address [in brackets].
  - User—Empty because the proxy cannot know user name at logout.
  - Location—The path portion of the request's URL.
  - AuthType—Empty because the proxy does not authenticate.
  - Time—Empty because the proxy does not track login time.
  - Server—Destination back-end server for the request.
- `DIAG.HTTPPROXY.PROTOIN(`*`selector`*`)`—Records an inbound connection to the HTTP proxy. Actions can be page requests, logins, errors, and connections. The *`selector`* is a comma-separated list of key-value pairs, possibly using * as a wildcard, for example:

  ```
  host=99.999.99.1,port=*
  ```

  Logging is selected by a combination of host, port, and session. Some *`selector`* must be given, for example (`host=*`). The *`selector`* may combine:
  - `host`—Host name or IP address of the connecting remote machine. Netmasked subnet addresses are not yet supported.
  - `port`—If a host name is specified, the port number on that host.
  - `session`—Unique identifier for a particular connection (optional).
- `DIAG.HTTPPROXY.PROTOOUT(`*`selector`*`)`—Records an outbound connection from the HTTP proxy; *`selector`* can be `host` only.
- `DIAG.IMAP.PROTOIN(`*`selector`*`)`—Records an IMAP session; *`selector`* is the same as for `DIAG.HTTPPROXY.PROTOIN`, for example (`host=*`).
- `DIAG.IMAPPROXY.BADLOGIN`—The back-end server rejected a login. After the identifier, fields are:
  - From—Transport type (`ssl` or `clr`) and IP address [in brackets].
  - User—The user for whom proxying failed.
  - AuthType—Empty because the proxy does not authenticate.
  - Server—IP address of the back-end server.
  - Reason—A description of the failure.
- `DIAG.IMAPPROXY.LOGFAILURE`—A connection attempt failed. This is logged only if a user could not be found in LDAP. After the identifier, fields are:
  - From—Transport type (`ssl` or `clr`) and IP address [in brackets].
  - User—The user not found in LDAP.
  - AuthType—Empty because the proxy does not authenticate.
  - Reason—A description of the failure.
- `DIAG.IMAPPROXY.LOGIN`—A login to the back-end server was successful. After the identifier, fields are:
  - From—Transport type (`ssl` or `clr`) and IP address [in brackets].
  - User—The user who logged in.
  - AuthType—Empty because the proxy does not authenticate.
  - Server—IP address of the back-end server.

- `DIAG.IMAPROXY.LOGOUT`—A user logged out from the back-end server. After the identifier, fields are:
  - From—Transport type (`ssl` or `clr`) and IP address [in brackets].
  - User—The user who logged out.
  - AuthType—Empty because the proxy does not authenticate.
  - Server—IP address of the back-end server.
- `DIAG.IMAPPROXY.PROTOIN(`*`selector`*`)`—Records an inbound connection to the IMAP proxy (see `Imap Get Mode`); *`selector`* is the same as above. Except for the IMAP `login` command, no protocol events are logged.
- `DIAG.IMAPPROXY.PROTOOUT(`*`selector`*`)`—Records IMAP proxy connection that is outbound from the server; *`selector`* can be `host` only.
- `DIAG.MTA.PROTOIN(`*`selector`*`)`—Records a session that is inbound to SMTP; *`selector`* can be `host` or `port` only.
- `DIAG.MTA.PROTOOUT(`*`selector`*`)`—Records a session that is outbound from SMTP on the server; *`selector`* can be `host` only.
- `DIAG.POP.PROTOIN(`*`selector`*`)`—Records a POP session; *`selector`* is the same as for `DIAG.HTTPPROXY.PROTOIN`, for example (`host=*`).
- `DIAG.POPPROXY.BADLOGIN`—The back-end server rejected a login. After the identifier, fields are the same as for `DIAG.IMAPPROXY.BADLOGIN`.
- `DIAG.POPPROXY.LOGFAILURE`—A connection attempt failed. This is logged only if a user could not be found in LDAP. After the identifier, fields are the same as for `DIAG.IMAPPROXY.LOGFAILURE`.
- `DIAG.POPPROXY.LOGIN`—A login to the back-end server was successful. After the identifier, fields are the same as for `DIAG.IMAPPROXY.LOGIN`.
- `DIAG.POPPROXY.LOGOUT`—A user logged out from the back-end server. After the identifier, fields are the same as for `DIAG.IMAPPROXY.LOGOUT`.
- `DIAG.POPPROXY.PROTOIN(`*`selector`*`)`—Records an inbound connection to the POP proxy (see `Pop Get Mode`); *`selector`* is the same as above. Except for the POP `user` command, no protocol events are logged.
- `DIAG.POPPROXY.PROTOOUT(`*`selector`*`)`—Records a POP proxy connection that is outbound from the server; *`selector`* can be `host` only.

System Console and Proxies

The system console and proxies log events using these identifiers:

- `CONSOLE.USER.BADLOGIN`—A login has failed. After identifier, fields are:
  - From—The word "`CONSOLE`" indicating subsystem.
  - User—The name of the user who attempted to login.
  - Type—The word "`PLAINTEXT`" indicating encoding.
  - Reason—An extended-reply string describing the failure.
- `CONSOLE.USER.LOGIN`— A login has succeeded. After identifier, fields are:
  - From—The word "`CONSOLE`" indicating subsystem.
  - User—The name of the user who attempted to login.

- Type—The word “PLAINTEXT” indicating encoding.
- Reason—An extended-reply string describing the event.

- CONSOLE.USER.LOGOUT—A user has logged out. After identifier, fields are:
  - From—The word “CONSOLE” indicating subsystem.
  - User—The name of the user who attempted to login.
  - Type—The word “PLAINTEXT” indicating encoding.
  - Time—Number of milliseconds during which user was logged in.
- OLD.SYSTEM.LOG—The CGI put this event into its System Log. Field is:
  - LogMessage—String with log message, as it appears in the System Log.
- SECURITY.HTTPPROXY.BODYTOOLARGE—The proxy received a request body that was too large. After identifier, fields are:
  - RemoteIP—The offending IP address.
  - Location—The URL of the request.
  - ContentLength—Length of the overly large body.
- SECURITY.HTTPPROXY.HEADEROVERFLOW—The proxy received an overly long header line. After identifier, fields are:
  - RemoteIP—The offending IP address.
  - Location—The URL of the request.
- SECURITY.HTTPPROXY.HEADERSTOOLONG—The header portion of a request was too long. After identifier, fields are:
  - RemoteIP—The offending IP address.
  - Location—The URL of the request.
  - HeaderSize—The size of the header in bytes.
- SECURITY.HTTPPROXY.TOOMANYTRAVERSALS—The proxy received a URL containing too many pathname components. After identifier, fields are:
  - RemoteIP—The offending IP address.
  - Location—The URL of the request.

Subcommands

Addroute

Starts routing log events to a particular output method, including over the network. Mirapoint systems have default routes, which you can show with `Log Listroutes` before adding any. Default routes are created at boot time. The administrator can delete them, but they reappear after reboot unless a name collision occurs. You add routes by specifying one or more logging identifiers, optionally using wildcards.

Syntax

tag `Log Addroute` *routingname identifierSet method methodArgs*

where:

- *routingname* is a tag that you can use later to delete the log routing. It may be specified as "" (null string), in which case a name is assigned automatically.
- *identifierSet* is a pattern string containing a list of space-separated identifiers (see Identifiers on page 340). Star * wildcards may be used. The *identifierSet* selects which events to log.
- *method* is the output method for this profile, such as `SYSLOG` or `WATCH`.
- *methodArgs* is a list of space-separated arguments as given below. The `SYSLOG` method transmits each serialized string in a single UDP packet to a given host and port, presenting it as a system log message. The serialized string is truncated at 8KB, including CRLF. The *methodArgs* take the following form where *host* is required but later arguments are increasingly optional:

  host:*port*:*facility*.*priority*

  - *host* specifies the destination host name, for instance a `SYSLOG` server.
  - *port*, if present, specifies destination port. Must be present if *facility* is given, but if missing, defaults to 514.
  - *facility*, if present, specifies one of the system log categories given below. If missing, this defaults to the `mail` facility.
    - `auth`—Programs to check primary and secondary user authorization.
    - `authpriv`—Programs to authenticate users' system privileges.
    - console—Output to the system console device.
    - `cron`—Facility to execute scheduled commands.
    - `daemon`—System background processes.
    - `kern`—The system kernel.
    - `mail`—The mail system.
    - `mark`—Time marks generated internally every 20 minutes.
    - `ntp`—Network Time Protocol subsystem.
    - `security`—Security subsystems, including firewall.
    - `syslog`—Generated internally by the system logging facility.
    - `user`—User processes (the default configuration on `SYSLOG` servers).
    - `uucp`—Programs for Unix-to-Unix copy.
    - `local0-local7`—Reserved for local use.
  - *priority*, if present, specifies the `SYSLOG` priority. If absent or empty, the default is priority `Notice`, with priority `Emerg` for `*.ALERT` items, priority `Err` for `*.ERROR` items, and priority `Debug` for `*.DEBUG` items. If you specify a priority, all messages matched by *identifierSet* receive that priority.

- `emerg`—Highest priority: a panic condition indicating that the system is unusable, normally broadcast to all users.
- `alert`—A condition that should be corrected immediately, such as a corrupted system database.
- `crit`—Critical condition, for example, hard device errors.
- `err`—Error condition, not too severe.
- `warning`—Warning condition, not severe.
- `info`—Informational only.
- `notice`—Not an error condition, but may require special handling.
- `debug`—Low priority: normally used only for debugging a program.
- `none`—Disable messages from the associated facilities.

The `WATCH` method takes a null string argument. The aggregate of `WATCH` routes is recorded as historical information for the `Log Watch` command. If all `WATCH` routes are deleted, then no historical information is recorded and `Log Watch` sessions display no data.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Log Addroute login "*.USER.LOGIN *.USER.LOGOUT" SYSLOG syslog:514:user
2 OK Completed

3 Log Addroute watchlogin "*.USER.LOGIN *.USER.LOGOUT" WATCH ""
3 OK Completed
```

Countroutes

Counts the number of active event-to-output routings.

Syntax

tag `Log Countroutes` *pattern*

where *pattern* must be * (asterisk) or "" (null string).

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
4 Log Countroutes ""
* 4 2
4 OK Completed
```

Deleteroute

Deletes the named routing. It is an error if the routing name does not exist. Deleting a `WATCH` routing affects data displayed by any running `Log Watch` command.

Syntax

`tag Log Deleteroute routingname`

where *routingname* is the name of a routing previously set with `Log Addroute`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Log Deleteroute connecttime
6 OK Completed
```

Diagwatch

Watches diagnostic events, which could be helpful for troubleshooting a system. The `Log Diagwatch` command is similar to `Log Watch`, although it lacks sequence numbering and historical data. Output can be halted by typing `DONE`.

You do not have to run `Log Addroute` before using `Diagwatch`, but you must specify some pattern as documented under Diagnostics Subsystem on page 350. For example, `DIAG.*.PROTOIN(*)` traces all inbound protocols.

Syntax

`tag Log Diagwatch identifierSet`

where *identifierSet* is a string containing a list of space-separated identifiers, as listed in the Data Formats Identifiers section (see Identifiers on page 340).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Log Addroute unused DIAG.*.* watch ""
8 OK Completed

9 Log Diagwatch DIAG.MTA.PROTOIN(host=*)
+ idling
* 432 hostname 969929449.000 DIAG.MTA.PROTOIN(host=mail.example.com) <<<...
* 435 hostname 969929450.000 DIAG.MTA.PROTOIN(host=mail.example.com) >>>helo
* 437 hostname 969929452.000 DIAG.MTA.PROTOIN(host=mail.example.com) <<<...
done
9 OK Log diagwatch terminated
```

Get

Responds with the value of the specified parameter.

Syntax

tag `Log Get` *parameter*

where *parameter* is one of the following (see `Log Set`):

- `History`—Time for which history is preserved for the `Log Watch` command
- `Markinterval`—How often to update the last functioning time on disk.
- `Syncinterval`—Specifies how often the historical `WATCH` is flushed to disk.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
10 Log Get History
* 10 12
10 OK Completed
```

Listroutes

Shows the active event-to-output routings, including all arguments supplied when each routing was created with `Log Addroute`.

Syntax

tag Log Listroutes *pattern start count*

where:

- *pattern* is currently ignored.
- *start* is the first of the routes to list. The empty string ("") implicitly means 0.
- *count* is number of routes to list. The empty string ("") implicitly means all. If greater than the total number of routes, Listroutes shows as many as possible.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
5 Log Listroutes "" "" ""
5 connecttime "*.USER.LOGIN *.USER.LOGOUT" SYSLOG "loghost"
5 watchconnect "*.USER.LOGIN *.USER.LOGOUT" WATCH ""
5 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

tag Log Set *parameter value*

where:

- *parameter* is one of:
  - History—Sets the time for which history is preserved for the Log Watch command. The *value* hours must be in the range zero to 168 (the default). Data gets stored daily, not hourly, so a *value* under 24 has the same effect as 24. *Value* may be specified as (hours=*n*) or (days=*n*) but either way, the time segment is always at least one day.
  - Markinterval—An internal marker kept to help recover system data when the system is rebooted or shuts down for some unexpected reason. The *value* in seconds indicates how often to update the last functioning time on disk. This is used at system startup to help salvage account logging information by generating a synthetic logout log message with the mark time. The range of this setting is zero to 3600, where zero indicates that there is a mark after each log event. This defaults to 30 seconds and should be as large as possible to avoid extra system overhead.

- ❖ `Syncinterval`—Specifies how often the historical `WATCH` is flushed to disk. It may be flushed much more frequently, but this serves as a guarantee that all data makes it to disk. The range of this setting is zero to 3600, where zero indicates that the logs are running as synchronously as possible. The default is 60 seconds.

- ◆ *`value`* is the value you assign to *`parameter`*.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
9 Log Set History 12
9 OK Completed
```

Watch

Watches current log events. Displays any saved events matching *`identifierSet`* (from `Log Addroute`) with sequence numbers greater than or equal to *`sequencing`*. Afterwards, it displays new events matching *`identifierSet`* as the logging arrives. You can stop `Log Watch` by typing `DONE`.

Note: if `Log Addroute` modifies the list of identifiers that are being handled for the `WATCH` method, this is reflected in any new data that are seen. Thus it is possible for nothing to be printed on the connection if all `WATCH` routes are removed.

A reboot or failover terminates all ongoing login connections. The system attempts to synthesize a series of logout events (`POP.USER.LOGOUT`, `IMAP.USER.LOGOUT`, etc.) with a user name of "*" upon restart, and a time based on the last mark or last log message, whichever appears to be most recent.

A sequence number is a 64-bit integer that is incremented for each event handled by `Log`. All historical data disappears and the sequence number is reset to 0 whenever system software is reinstalled. This is the only time the sequence number gets reset.

Syntax

```
tag Log Watch identifierSet sequencing
```

where:

- ◆ *`identifierSet`* is a string containing a list of space-separated identifiers, as previously specified in a `Log Addroute` command.

- ◆ *`sequencing`* is a keyword indicating how to begin: `ALL` displays all saved events matching the given identifier set, `NEW` displays no saved events, `0` (zero) displays events starting at the first, and a sequence number starts at that event.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
7 Log Watch "*.USER.LOGIN *.USER.LOGOUT" new
+idling
10009 host 10183069.438 ADMIN.USER.LOGIN clr[0.1.0.84] juser plaintext
10010 host 10183177.747 ADMIN.USER.LOGOUT clr[0.1.0.84] juser plaintext 108320
done
7 OK Log watch terminated
```

34

The Mailbox Command

The `Mailbox` commands create and delete mailboxes, regulate access to mailboxes, copy or rename mailboxes, list mailboxes, and undelete or expire messages.

Mailboxes are usually named `user.`*`name`* where "`user`" is fixed and *`name`* represents the user account name, as set by the `User Add` command. Shared folders often have a different prefix, and Junk Mail Manager assigns the "`quarantine`" prefix.

Item count in a mailbox cannot exceed 50,000 (or 150,000 on Series 5 hardware). Although not technically a quota, users get a "Mailbox full" error when a message arrives at a mailbox containing more than this number of items.

Access Control Lists

Associated with each mailbox is an **access control list** (ACL), which specifies the permissions that various users have on the mailbox. The form of the ACL, as defined by an extension to the IMAP4 standard (see RFC 2086), is a list of pairs of user names and sets of permissions:

`username permissions username permissions ...`

where:

- *`username`* is the login name of a user (which is not required to exist on the system) or one of these reserved names:
  - `administrator`, system manager allowed to delete mailboxes
  - `Administrators`, all users with administrator privileges
  - `Anonymous`, reserved
  - `Anyone`, a wildcard representing all users
- *`permissions`* is a string containing one or more single-letter codes, each of which represents an access permission:

Table 4 Access Permissions

Code	Permission	Description
a	administer	The user may change the ACL of the mailbox. Administrators implicitly have this permission on all mailboxes.
c	create	The user may create submailboxes of the mailbox.

Table 4 Access Permissions

Code	Permission	Description
d	delete	The user may mark messages in the mailbox for deletion, may expunge these messages, and may delete or rename the mailbox itself. To rename a mailbox, the user must also have the `create` permission on the parent mailbox. Administrators do not implicitly have the `delete` permission on mailboxes, but can add this permission for themselves to any mailbox.
i	insert	The user may insert messages into the mailbox using the IMAP `COPY` and `APPEND` commands.
l	lookup	The user may see the mailbox name in listings.
p	post	The user may post messages to the delivery address for the mailbox for delivery by the SMTP service. ACL
r	read	The user may open the mailbox, read and search messages in the mailbox, and copy message from the mailbox.
s	seen	The system keeps a persistent record of which messages in the mailbox the user has read (seen).
w	write	The user may modify flags on messages in the mailbox other than `\Seen` and `\Deleted`. RFC 2060 defines the possible flags.

The `p` flag for wildcard user `Anyone` must be set to allow delivery of email addressed to a shared folder. Two examples of such email: userid+sharedfolder@example.com and +shared.folder.subfolder@example.com.

The Access Control page in the GUI (under WebMail Options) only has four settings, which correspond to the above permissions as follows:

```
Read:   l+r+s
Write:  w+i+d
Mail:   p
Admin:  c+a
```

Subcommands

Add

Creates a mailbox with the specified name.

Folder names may include characters from the range `[A-Za-z 0-9_-]`. However dot (.) and slash (/) are restricted for use as separators, leading plus (+) designates shared folders, and asterisk (*), percent (%), and quote (") are limited by IMAP. Note that mailbox folder names are stored as Unicode modified UTF-7, so the ampersand (&) is actually “&-” in the file system.

Mailbox names, including any subfolders, are limited to 200 bytes in length.

Syntax

```
tag Mailbox Add mbname
```

where `mbname` is the name of the new mailbox, for example `user.name` where `name` represents the user name as set by the `User` command.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (can access only their own mailboxes)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
2 Mailbox Add user.glenn
2 OK Completed
```

Copy

Makes a copy of the specified mailbox, possibly from a different system. Copying uses IMAP service, which must be enabled and started on the source host.

Syntax

```
tag Mailbox Copy srchost srcmbox srcuser srcpasswd destmbox
```

where:

- *srchost*—The name of an IMAP server where the original mailbox exists, or "`(hostname=srchost)(ssl=Yes)`" to select IMAP-over-SSL on port 993. The default `(ssl=No)` selects the regular IMAP port 143.
- *srcmbox*—The name of a mailbox to be copied, or an arbitrary collection of the following parenthesized flags:
  - `(mailbox=srcmbox)`—Specifies the source mailbox in presence of other options. Unlike other parenthesized flags, this one is mandatory. Note: in UW namespace, the default starting point is the current user's INBOX; use sharp (#) to start at root for another user's mailbox. See IMAP Namespaces on page 275 for details.
  - `(error=Continue)`—Keep copying in the face of errors. In case of error, displays the `MAILBOXSTATUS` line, and prints an error line for each message that cannot be copied, including the UID, MsgID, and Reason. Mailbox repair on the destination must be done for files warned about, otherwise messages will be lost. The default `(error=Stop)` gives up on error, and cleans up by deleting any mailbox(es) copied beforehand, recursively if so specified with the `(recursive=Yes)` option.
  - `(recursive=Yes)`—Copies the named mailbox and all its submailboxes. Displays the `MAILBOXSTATUS` line if not already seen. Copy recursion is

intended to support non-Mirapoint systems as well; the remote hierarchy delimiter is used. Wildcards are not supported. The administrator must correctly specify mailbox(es) on the remote system, including UW-Cyrus mappings according the Mirapoint IMAP `Namespace` setting (two `Copy` commands are required from UW servers: one for `INBOX`, and another for recursively copying `~/` locally, since there is no way to specify both in a single pass). If a remote mailbox name contains an unsupported character, an underscore (_) is substituted in the local name.

- ❖ `(verbose=Yes)`—Provides additional error reporting. Displays the `MAILBOXSTATUS` line after IMAP has opened the mailbox. Fields in this line include: the remote mailbox name, the remote mailbox's UID validity, the remote mailbox's UID next message, the number of messages in the remote mailbox, and the local mailbox copy name. Additionally, every 100 messages a line displays the number of messages copied. Default is `(verbose=No)`, which by itself provides no reporting.

- ◆ *`srcuser`*—Name of the user who owns the original mailbox. The remote login triplet always gives authentication identity, authentication password, and authorize identity. When the administrator copies a remote mailbox, the triplet could be mailbox owner, owner's password, and administrator (the connection is processed with Administrator login, but owner permissions). If the triplet is specified as "`(user=`*`Administrator`*`)(authorizeId=`*`srcuser`*`)`" then the administrator password substitutes for the mailbox owner's password, so that administrators can copy without knowing the user password. Remote copy attempts regular authentication using login and password if `authorizeId` is not specified, else a triplet with `AUTHENTICATE PLAIN` if `authorizeId` is specified. However if the remote host does not support `AUTHENTICATE PLAIN`, copy fails. IMAP must be in proxy mode to support `authorizeId`.
- ◆ *`srcpasswd`*—The user authentication password corresponding with *`srcuser`*, required for security checking.
- ◆ *`destmbox`*—The name you assign to the copied mailbox. Copy creates the mailbox if it does not exist, otherwise it just adds messages to the mailbox.

If *`destmbox`* identifies a user mailbox, such as `user.jan`, and the user account `jan` does not already exist, *`destmbox`* is created with no ACL. If the user `jan` does exist, *`destmbox`* is created using the default ACL set for a mailbox.

Privilege Levels

- ◆ Administrator
- ◆ Helpdesk administrator
- ◆ Domain administrator
- ◆ Users (can access only their own mailboxes)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

The first example is a local mailbox copy in Cyrus namespace. The second example is a local mailbox copy in UW namespace. The third example is a remote copy from an IMAP server in Cyrus namespace. The fourth example is a remote copy from an IMAP server in UW namespace.

```
1 Mailbox Copy localhost user.joan joan joanpass user.jane
1 OK Completed
```

```
2 Mailbox Copy localhost #user.joan joan joanpass user.jane
2 OK Completed
```

```
3 Mailbox Copy "(hostname=cyrus.example.com)(ssl=yes)"
"(mailbox=user.u22)(recursive=yes)(error=continue)(verbose=yes)"
"(user=Administrator)(authorizeid=u22)" adminpass  user.u22
3 OK Completed
```

```
4 Mailbox Copy "(hostname=uw.example.com)(ssl=yes)"
"(mailbox=#user.u23)(recursive=yes)(error=continue)(verbose=yes)"
"(user=Administrator)(authorizeid=u23)" adminpass  user.u23
4 OK Completed
```

Count

Responds with the number of mailboxes that match the specified pattern.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- Users (can access only their own mailboxes)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Syntax

`tag Mailbox Count pattern`

where *pattern* is a mailbox name, which can include these wildcard characters:

- *—matches zero or more characters of any kind; this includes mailbox hierarchy separators ('.').
- %—matches zero or more characters, not including mailbox hierarchy separators; this wildcard is provided for compatibility with the IMAP4 protocol.

Example

```
4 Mailbox Count *
* 4 1
4 OK Completed
```

Delete

Removes the specified mailbox and subfolders.

Administrators do not have automatic permission to delete any mailbox. Use the `Mailbox Setacl` command to grant administrators permission to delete a mailbox before attempting removal. With a delegated domain current, you must use the "administrators" keyword (not Administrator). See Access Control Lists on page 361 for more information.

Deleting an Inbox also removes all the user's messages within the deletedmessages hierarchy; see `Mailbox Undelete`.

Syntax

```
tag Mailbox Delete mbname
```

where *mbname* is the name of the mailbox you want to delete.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (can access only their own mailboxes)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
11 Mailbox Setacl user.glenn Administrator +d
11 OK Completed
12 Mailbox Delete user.glenn
12 OK Completed
```

Get

Responds with the value of the specified parameter.

Syntax

```
tag Mailbox Get parameter
```

where *parameter* may be:

- `Broadcast`—a shared mailbox that addresses all users in a domain. Messages in the broadcast mailbox appear in every user's inbox. See `Mailbox Set`.
- `Undeletequota`—enables users to undelete messages without LDAP services.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
8 Mailbox Get Undeletequota
* 8 ""
8 OK Completed
```

Getacl

Responds with the access control list (ACL) of the specified mailbox (see Access Control Lists on page 361).

Syntax

```
tag Mailbox Getacl mbname
```

where *mbname* is the name of the mailbox for which you want to see the ACL.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- Users (can access only their own mailboxes)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
5 Mailbox Getacl user.glenn.personal
* 5 user.glenn.personal glenn lrswipcda
5 OK Completed
```

List

Responds with a list of mailboxes that match the specified pattern.

Syntax

tag Mailbox List *pattern start count*

where:

- *pattern* is a mailbox name, which can include these wildcard characters:
  - *—matches zero or more characters of any kind; this includes mailbox hierarchy separators ('.').
  - %—matches zero or more characters up to, but not including, mailbox hierarchy separators; this wildcard is provided for compatibility with the IMAP4 protocol.
- *start* is the first position in the list of matches that you want to see. The empty string ("") implicitly means 0.
- *count* is number of matches that you want to see. The empty string ("") implicitly means all matches. If *count* is greater than the total number of matches, list returns as many matches as possible.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- Users (can access only their own mailboxes)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
6 Mailbox List user.g* "" ""
* 6 () "." user.glenn
* 6 () "." user.glenn.personal
6 OK Completed
```

Msgexpirenow

Message aging is controlled by the `Cos Enable Msgexpiration` command and related LDAP attributes. The `Mailbox Msgexpirenow` activates expiration at any time, and aged messages may be scheduled for automatic expiration. For mailboxes that have `miMailExpirePolicy`, the expiration command prints a line showing domain, mailbox name, and a brief narrative description (even if nothing occurred). The following CLI command schedules message expiration daily at 3 AM for all users in the current domain, or top-level domain if none is current:

```
Schedule Add aging daily 3 "Mailbox Msgexpirenow (users=*) 0"
```

If users with a `miMailExpirePolicy` have messages restored or undeleted, the restored messages may be deleted the next time this expiration command runs. You might expect restored messages to get a new lease on life, a reset, but they do not. Workaround: the administrator could temporarily disable message expiration for users after any restores or undeletes.

Syntax

`tag` `Mailbox Msgexpirenow` *`criteria limit`*

where:

- *`criteria`*—One or more parenthesized attribute/value pairs indicating scope of message expiration. Attributes must be one of those below, followed by an equal sign, then the attribute value. Values may use wildcard patterns like the `Mailbox List` command. The empty string is not a valid attribute/value.
  - `domains`—The domain(s) in which to expire messages. If not specified, expires users or shared mailboxes in the current or top-level domain.
  - `users`—User mailboxes to expire. If none are specified, no mailboxes are expired. Thus, "`(users=)`" is a no-op, "`(users=*)`" expires all mailboxes, and "`(users=*)(domains=*)`" expires all mailboxes in all domains.
  - `sharedmailboxes`—The shared mailboxes to expire. If not specified, no shared mailboxes are expired.
- *`limit`*—Maximum number of user accounts and shared mailboxes in which to search for expired mail messages. May be in the range one (1) to a million. Null string ("") or 0 indicates no limit. Both user and shared mailboxes count towards the limit. Expiration occurs in the following order: top level domain then any delegated domains, then shared mailboxes in alphabetic order, then users in alphabetic order.

Privilege Levels

Administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
13 Mailbox Msgexpirenow "(domains=example.com)(users=*)" ""
* 13 example.com user.juser "9 messages deleted"
* 13 example.com user.fred "0 messages deleted"
* 13 example.com user.fred.trash "57 messages deleted"
* 13 example.com user.wilma "1 message deleted"
13 OK Completed
```

```
14 Schedule Add age daily 3 "t Mailbox Msgexpirenow \"(users=*)(domains=*)\" 0"
14 Ok Completed
```

Publish

Requests the mailbox cache software to publish shared local folders into LDAP. Publishes all shared folders in all domains. This command is required to get initial folder information into the directory server. Although if the system is already using LDAP folders, folder changes get published without an explicit `Publish` command.

When a folder has an ACL placed on it, an implicit publish is done. All that folder's parents are also published with the appropriate type. Any missing parent folders are assumed to be of type `STRUCTURAL`, as are most folders except User.Administrator or User.Quarantineadmin (`LOCAL`) and User.*Name* (`NORMAL`).

Syntax

```
tag Mailbox Publish domains folders
```

where currently *`domains`* must be ***@*** to indicate all domains, and *`folders`* must be "" to indicate all folders.

Privilege Levels

Administrator

Domain Sensitivity

Can be used only in the primary domain.

Example

```
18 Mailbox Publish *@* ""
18 WARNING "tbd"
18 INFO "Published 76 of 77 objects"
18 OK Completed
```

Rename

Changes name of the specified mailbox. To change mailbox name, you must have `delete` permission on the mailbox and `create` permission on its parent mailbox.

You can rename a user mailbox (with `user.` *prefix*) to another user mailbox, and you can rename one shared mailbox (without the `user.` *prefix*) to another, but you cannot rename a shared mailbox to a user mailbox, or vice-versa. Instead, you must create a new mailbox, copy messages, duplicate ACLs, and delete the old mailbox.

Syntax

tag `Mailbox Rename` *oldname newname*

where:

- *oldname* is the name of the mailbox you want to rename.
- *newname* is the new mailbox name.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (can access only their own mailboxes)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
8 Mailbox Rename user.glenn.personal user.glenn.private
8 OK Completed
```

Set

Sets the value of the specified parameter.

Mirapoint provides a more efficient message broadcast facility, which you might find more convenient than `Message Set Broadcast`, packaged as the `E3_broadcast` patch. Check with Mirapoint customer support for availability and documentation.

Syntax

tag `Mailbox Set` *parameter value*

where:

- *parameter* is one of the following:
  - `Broadcast`—a shared mailbox for notifying all user accounts in a domain. The *value* can specify any mailbox name; the last one set takes precedence. If the mailbox does not exist, you must create it before this facility works. To reach all users on a system, set broadcast mailbox(es) for the primary domain and for all delegated domains. Messages placed into the broadcast

mailbox, for example by copy and paste in WebMail, get copied into a user's inbox the next time that user logs in. Broadcast messages are not forwarded; users with `Fwd` who never log in do not receive them. Use with care, because mailbox broadcast necessitates double folder login, which adversely affects system performance. To disable mailbox broadcast, set the mailbox value to null string (`""`), the default.

  - `Undeletequota`—This option enables users to undelete messages without requiring LDAP service. The *`value`* specifies the size in kilobytes (not bytes) for the undelete quota, or "" to disable undelete. If `Undeletequota` is set, every user and shared mailbox on the system gets the same undelete quota. This setting is disabled by default. For more information, see Undelete on page 374.

- *`value`* is the value that you assign to *`parameter`* (see above).

Privilege Levels

Administrator

Domain Sensitivity

The `Undeletequota` parameter returns an error if issued in a delegated domain.

Example

```
6 Mailbox Add announce
6 OK Completed
7 Mailbox Set broadcast announce
7 OK Completed

15 Mailbox Set Undeletequota 1000
15 OK Completed
```

Setacl

Sets the access control list on the specified mailbox.

Syntax

tag `Mailbox Setacl` *mbname user permissions*

where:

- *`mbname`* is the name of the mailbox for which you want to set the ACL.
- *`user`* is the login name of a user whose permissions on the mailbox you want to set. This ACL user must already exist, although the mailbox owner does not need to exist. If a delegated domain is current, the user is interpreted as being in the current domain. The user name `administrators` indicates all users with administrator privilege. Use the @ symbol for LDAP groups.
- *`permissions`* is a string of single-character codes representing access permissions (see Table 4 on page 361) with an optional prefix.

If the string is prefixed by a plus sign (+), the specified permissions are added to the user's permissions on the mailbox.

If the string is prefixed by a minus sign (-), the specified permissions are removed from the user's permissions on the mailbox.

If there's no prefix, the specified permissions replace the user's existing permissions on the mailbox.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (can access only their own mailboxes)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
10 Mailbox Setacl user.glenn.private glenn adilprsw
10 OK Completed
```

```
11 Mailbox Setacl archive.mygroup @ldap:mygroup +lwriscpa
11 OK Completed
```

Sync

Requests the mailbox cache software to synchronize all folders from LDAP and make that information available to other parts of the system. This `Sync` command should be scheduled to provide periodic refreshes of shared folders from LDAP.

Syntax

tag `Mailbox Sync` *domains folders*

where currently *domains* must be ***@*** to indicate all domains, and *folders* must be "" to indicate all folders.

Privilege Levels

Administrator

Domain Sensitivity

Can be used only in the primary domain.

Example

```
19 Mailbox Sync *@* ""
19 OK Completed
```

Undelete

Moves all recently deleted messages from a user's message holding hierarchy, under `deletedmessages` on the Mirapoint server, to the specified mailbox. Requires "a" (administer) access rights. Returns the number of messages moved. If retrieving all specified messages would exceed the user's `mailQuota`, no messages are moved.

In practice, administrators seldom run the `Mailbox Undelete` command. Instead, they set up COS (class of service) so users can undelete their own messages. Deleted message retrieval is allowed if "undelete quota" appears in the LDAP user record or domain-wide LDAP entry. Alternatively the `Mailbox Set Undeletequota` command enables message undelete without LDAP, for everyone, not under COS control.

When messages preserved into a user's undelete hierarchy put that user over their undelete quota, the oldest messages in the user's undelete hierarchy are removed until more than 10% of space becomes free. Undelete quota is a COS value, `miMailUndeleteQuota`. See "The Cos Command" on page 131.

If a deleted message is larger than the undelete quota, it is temporarily allowed in the undelete hierarchy. For example, if a user deletes a 20 MB message but has only a 10 MB undelete quota, all messages except the 20 MB message are removed. The next time that user expunges a message, even a small message, the 20MB message is removed and the undelete hierarchy becomes temporarily underutilized.

Deleted messages cannot be read as email, only undeleted to their original folder. If WebMail users have the "delete to Trash" preference set, which is the default, undeleted messages get restored to the Trash folder, not to the folders from which they were originally deleted. Message undelete is useful primarily with IMAP clients such as Outlook (Express) and Mozilla Thunderbird.

Syntax

```
tag Mailbox Undelete mailbox
```

where *`mailbox`* specifies a place in the `/deletedmessages` hierarchy where messages should be copied to *`mailbox`* in the normal folder hierarchy. The special mailbox name `INBOX` is an abbreviation for the `user.`*`name`* top-level mailbox.

Privilege Levels

- Administrator
- Users (can undelete their own mailboxes)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
11 Mailbox Undelete user.joe
* 11 93
11 Ok Completed

12 Mailbox Undelete user.joe.Sent
12 No Operation would exceed quota
```

The Maildom Command

The `Maildom` command manages local-delivery mail domains on Mirapoint systems.

Mail Domains

A **mail domain** is a Domain Name Service (DNS) domain for which your Mirapoint system accepts mail for local delivery or routing within your organization. Mail domains are useful for identifying alternate domain names for your organization or for routing messages to virtual or delegated domains hosted on other Mirapoint Message Servers. Your DNS database must have MX records referring to your Mirapoint system for each mail domain. Mail domain hosts are allowed to relay if DNS lookup determines that both PTR and "A" records exist, and match.

A message sent to *address@maildom* can only be delivered successfully if there is a mailbox named *address* on the Mirapoint system identified by *maildom*.

For correct determination of local delivery, you should add mail domains for DNS CNAME and "A" records. There is a per-system limit of 500,000 mail domains.

Subcommands

Add

Adds a domain to the list of mail domains for which the SMTP service attempts local delivery.

Syntax

```
tag Maildom Add mail-domain
```

where *mail-domain* is the name of the mail domain you want to add. If this value starts with "`*.`" (as in `*.example.com`), the SMTP service attempts local delivery or routing for all subdomains of the specified domain (but not also for the top-level domain `example.com`). For example, a value of `*.example.com` indicates that local delivery or routing will be attempted for `mail1.example.com`, `mail2.example.com`, and so on.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Maildom Add example.com
2 OK Completed
```

Count

Responds with the number of domains for which the SMTP service on your Mirapoint system attempts local delivery(see Mail Domains on page 377).

Syntax

`tag Maildom Count pattern`

where *pattern* is a string, possibly containing wildcards, matching mail domains. See Using Patterns on page 49.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
3 Maildom Count "*com"
* 3 2
3 OK Completed
```

Delete

Deletes a mail domain.

Syntax

`tag Maildom Delete mail-domain`

where *mail-domain* is the name of the mail domain you want to delete(see Mail Domains on page 377).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Maildom Delete example.com
4 OK Completed
```

List

Responds with a list of domains for which the SMTP service on your Mirapoint system attempts local delivery(see Mail Domains on page 377).

Syntax

`tag` `Maildom List` *`pattern start count`*

where:

- *`pattern`* is a string, possibly containing wildcards, that matches mail domains. See Using Patterns on page 49.
- *`start`* is the number of the first mail domain you want to see. The empty string (`""`) implicitly means `0`.
- *`count`* is the total number of mail domains you want to see. The empty string (`""`) implicitly means all mail domains. If *`count`* is greater than the total number of mail domains, `list` returns as many mail domains as possible.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
5 Maildom List "*com" "" ""
* 5 example.com
* 5 beispiel.com
5 OK Completed
```

36

The Mailq Command

The `Mailq` command lets you view the messages currently waiting in the outgoing mail queue. If necessary, it also lets you selectively delete messages from the queue.

Using Patterns

The `Count`, `Delete`, `List`, and `Reject` subcommands allow you to specify a **pattern** for several message attributes. Patterns are case-insensitive (except for queue IDs) and can contain the `?` (any character) and `*` (zero or more characters) wildcards.

For `Mailq` subcommands with a pattern, *`pattern`* can take the following forms:

- "" (empty string)—equivalent to the wildcard character `*`, meaning all message queue IDs.
- *`value`*—a pattern string, optionally containing wildcard characters, matching the queue IDs to count, delete, or list.
- Any combination of the following key/value pairs, enclosed in double quotes:

  ```
  "(id=val)(age=val)(size=val)(sender=val)(reason=val)(recipients=val)"
  "(id=val)(age>=val)(size>=val)(sender=val)(reason=val)(recipients=val)"
  ```

  Here *`val`* specifies a pattern string, possibly containing wildcard characters, the meaning of which depends on the specified key. The keywords `age` and `size` also accept `>=` and `<=` in addition to equal sign before *`val`* value.

  - `id`—*`value`* represents a (case-sensitive) queue ID.
  - `age`—*`value`* represents a message age in days (the default) or in units specified by the following suffixes. Messages `>=` (greater than or equal to) and `<=` (less than or equal to) the given age are also matched:
    - `w`—weeks
    - `d`—days
    - `h`—hours
    - `m`—minutes
    - `s`—seconds
  - `size`—*`value`* represents a message size in bytes (the default) or in units specified by the following suffixes. Messages `>=` (greater than or equal to) and `<=` (less than or equal to) the given size are also matched:
    - G—gigabytes
    - M—megabytes
    - K—kilobytes
  - `sender`—*`value`* represents the message originator's address.

- `reason`—*`value`* is a status explaining why this message is in the queue. Here are some common reasons:
  - Deferred: Connection refused
  - Deferred: Operation timed out
  - Reply: read error
- `recipients`—*`value`* represents one or more message recipient addresses.

Subcommands

Count

Responds with the number of entries currently in the mail queue.

Syntax

`tag` `Mailq Count` *`pattern`*

where *`pattern`* is as described in Using Patterns on page 49. Only those queue entries matching *all* the specified values are counted.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Mailq Count ""
* 2 15
2 OK Completed
3 Mailq Count *0001*
* 3 8
3 OK Completed
4 Mailq Count "(id=*0001*)(recipients=*7*)"
* 4 1
4 OK Completed
```

Delete

Deletes the specified entries from the mail queue.

Syntax

`tag` `Mailq Delete` *`pattern`*

where *`pattern`* is as described in Using Patterns on page 49. Only those queue entries matching *all* the specified values are deleted.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Mailq Delete AA000178
2 OK Completed
```

Get

Responds with the specified part of the specified queued message.

Syntax

`tag Mailq Get msg-part queue-id`

where:

- *`msg-part`* is one of:
  - `Envelope`—the message envelope
  - `Header`—the message header
- *`queue-id`* identifies the queued message for which you want to see *`msg-part`*.

Response

The response has the format:

`* tag qid data`

where:

- *`qid`* is the queue ID, the unique number that identifies the message in the mail queue
- *`data`* is a literal string containing the requested data (see Literal Strings in Responses on page 49)

For the `Envelope` message part, the output data can contain the following keywords in key-value pairs appended to certain SMTP commands:

- `BODY`—indicates the message body type. Possible values are:
  - 8BITMIME
  - 7BIT
- `ENVID`—the envelope identifier
- `RET`—indicates bounce behavior. Possible values are:

  - `hdrs`—a bounce message containing only the message headers will be generated
  - `full`—a bounce message containing the full message will be generated
- `NOTIFY`—indicates delivery status notification (DSN) behavior. Possible value are:
  - `DELAY`—a DSN will be issued when delivery is delayed
  - `FAILURE`—a DSN will be issued when delivery fails
  - `NEVER`—a DSN will never be issued
  - `SUCCESS`—a DSN will be issued on successful delivery
- `ORCPT`—the originally specified recipient

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Mailq Get Header AA000178
* 3 AA000178 {342}
Return-Path: <fred@fred.example.com>
Received: from localhost (fred.example.com [192.168.0.7])
    by test.example.com (1.0.0/1.0.Beta1) with SMTP id AA000178
    Sat, 27 Feb 1999 00:44:39 -0800 (PST)
Date: Sat, 27 Feb 1999 00:44:39 -0800 (PST)
Message-Id: <199902270844.AA000178@test.example.com>
Subject: Sample data
3 OK Completed
```

List

Responds with a list of mail queue entries.

Syntax

tag `Mailq List` *pattern start count*

where:

- *pattern* is as described in Using Patterns on page 49.
- *start* is the first entry in the mail queue that you want to see. The empty string (`""`) implicitly means `0`.
- *count* is number of mail queue entries that you want to see. The empty string (`""`) implicitly means all entries. If *count* is greater than the total number of entries, `list` returns as many entries as possible.

Response

There is an entry for each queued message; each entry has the format:

```
* tag qid arrtime stime "reason=status" size nrecip "recipient" "sender"
```

where:

- *qid* is the queue ID, a unique number that identifies this message in the queue.
- *arrtime* is the arrival time in the format:

  yyyymmddhhmmss

  where:

  - *yyyy* is the four-digit year
  - *mm* is the two-digit month (01 through 12)
  - *dd* is the two-digit day of the month (01 through 31)
  - *hh* is the two-digit hour (00 through 23)
  - *mm* is the two-digit minute (00 through 59)
  - *ss* is the two-digit second (00 through 59)
- *stime* is the time of the most recent change in message status for *recipient*. This is either the time of the most recent delivery attempt or "-------------" if SMTP service has not yet attempted delivery to *recipient*.
- *reason=status* is the message status for *recipient*. Examples include:

  ```
  "deferred"
  "host not responding"
  "connection timed out"
  ```

  If the SMTP service has not yet attempted delivery to *recipient*, this field is the empty string ("").
- *size* is the total size of the message in bytes.
- *nrecip* is the remaining number of recipients to whom the message has not yet been delivered.
- *recipient* is the email address of the currently pending recipient.
- *sender* is the email address of the originating sender.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Mailq List "" "" ""
* 5 ABP00002 20000605145119 ------------- "" 11 1 kennan@example.com
    demo@test.example.com
* 5 ABP00004 20000605153759 ------------- "" 9 1 rcpt1@example.com
    sndr1@test.example.com
```

```
* 5 ABP00005 20000605153806 -------------- "" 9 1 rcpt2@example.com
    sndr2@test.example.com
* 5 ABP00006 20000605153813 -------------- "" 9 1 rcpt3@example.com
    sndr3@test.example.com
* 5 ABP00007 20000605153820 -------------- "" 9 1 rcpt4@example.com
    sndr4@test.example.com
* 5 ABP00008 20000605153827 -------------- "" 9 1 rcpt5@example.com
    sndr5@test.example.com
* 5 ABP00009 20000605153834 -------------- "" 9 1 rcpt6@example.com
    sndr6@test.example.com
* 5 ABP00010 20000605153841 -------------- "" 9 1 rcpt7@example.com
    sndr7@test.example.com
* 5 ABP00011 20000605153848 -------------- "" 9 1 rcpt8@example.com
    sndr8@test.example.com
* 5 ABP00012 20000605153855 -------------- "" 9 1 rcpt9@example.com
    sndr9@test.example.com
* 5 ABP00013 20000605153902 -------------- "" 9 1 rcpt10@example.com
    sndr10@test.example.com
* 5 ABP00014 20000605153909 -------------- "" 9 1 rcpt11@example.com
    sndr11@test.example.com
* 5 ABP00015 20000605153916 -------------- "" 9 1 rcpt12@example.com
    sndr12@test.example.com
* 5 ABP00016 20000605153923 -------------- "" 9 1 rcpt13@example.com
    sndr13@test.example.com
* 5 ABP00017 20000605153930 -------------- "" 9 1 rcpt14@example.com
    sndr14@test.example.com
5 OK Completed
6 Mailq List *0001* "" ""
* 6 ABP00010 20000605153841 -------------- "" 9 1 rcpt7@example.com
    sndr7@test.example.com
* 6 ABP00011 20000605153848 -------------- "" 9 1 rcpt8@example.com
    sndr8@test.example.com
* 6 ABP00012 20000605153855 -------------- "" 9 1 rcpt9@example.com
    sndr9@test.example.com
* 6 ABP00013 20000605153902 -------------- "" 9 1 rcpt10@example.com
    sndr10@test.example.com
* 6 ABP00014 20000605153909 -------------- "" 9 1 rcpt11@example.com
    sndr11@test.example.com
* 6 ABP00015 20000605153916 -------------- "" 9 1 rcpt12@example.com
    sndr12@test.example.com
* 6 ABP00016 20000605153923 -------------- "" 9 1 rcpt13@example.com
    sndr13@test.example.com
* 6 ABP00017 20000605153930 -------------- "" 9 1 rcpt14@example.com
    sndr14@test.example.com
6 OK Completed
7 Mailq List "(id=*0001*)(recipients=*7*)" "" ""
* 7 ABP00010 20000605153841 -------------- "" 9 1 rcpt7@example.com
    sndr7@test.example.com
7 OK Completed
```

Reject

Deletes the specified entries from the mail queue, generating bounce messages to message senders. It is much like `Mailq Delete` except for the courtesy messages.

Syntax

tag `Mailq Reject` *pattern reason*

where *pattern* is as described in Using Patterns on page 49, and *reason* explains why.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
9 Mailq Reject AA000178 "Mailbox over quota."
9 OK Completed
```

Release

Causes specified messages to be released, if held in the queue by `Mailq Hold`.

Syntax

tag `Mailq Release` *pattern*

where *pattern* is as described in Using Patterns on page 49. Only queue entries matching *all* the specified pattern values are released.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
12 Mailq Release "(recipients=hogg)"
  * 12 8 messages released
12 OK Completed
```

```
13 Mailq Release ""
  * 13 97 messages released
13 OK Completed
```

Retryall

Attempts immediate delivery on all queue entries.

Syntax

tag `Mailq Retryall`

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Mailq Retryall
8 OK Completed
```

Transfer

Causes specified messages to transferred to the named host. The transfer mechanism is SMTP: MAILQ TRANSFER, which forces the next-hop host to be what you specify. When deliver is reattempted, messages are sent to the named host or IP address. If a host name is given instead of an IP address, MAILQ TRANSFER uses MX routing. Note: if the named host refuses to accept mail, it bounces, and you are responsible.

Syntax

```
tag Mailq Transfer pattern host-or-IP
```

where:

- *pattern* is as described in Using Patterns on page 49. Only queue entries matching *all* the specified pattern values are transferred.
- *host-or-IP* is a host name or IP address.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
14 Mailq Transfer "(tries>5)" holdit.example.com
* 14 138 messages marked for transfer
14 OK Completed
```

37 The Message Command

The `Message` command administers welcome messages, providing multiple formats, variable substitution, a policy mechanism, and character set localization.

This command is useful for customizing welcome messages to new users, and spam summaries from Junk Mail Manager.

A facility had a default setting and ties to other system services.

Subcommands

Count

Returns the number of non-default system messages.

Syntax

```
tag Message Count localepattern facilitypattern
```

where *localepattern* and *facilitypattern* must be either the null string ("") or asterisk (*).

Privilege Levels

- Administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

```
2 Message Count "" ""
* 2 0
2 OK Completed
```

Countargs

Returns the number of override variables for a message in *locale* and *facility*.

Syntax

tag `Message Countargs` *locale facility variable*

where *locale* must currently be the null string (""), and *facility* is one of those shown by `Message Listfacilities`, and *variable* must be the null string ("") or asterisk (*).

Privilege Levels

- ◆ Administrator
- ◆ Domain administrator
- ◆ Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

```
17 Message Countargs "" MESSAGE.JUNKMAIL.SUMMARY ""
* 17 1
17 OK Completed
```

Countfacilities

Returns the current number of valid message facilities.

Syntax

tag `Message Countfacilities` *pattern*

where *pattern* must be either the null string ("") or asterisk (*).

Privilege Levels

- ◆ Administrator
- ◆ Domain administrator
- ◆ Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

```
3 Message Countfacilities ""
```

```
* 3 4
3 OK Completed
```

Countpolicies

Returns the number of policies for a facility in a given message locale.

Syntax

tag `Message Countpolicies` *locale facility*

where *locale* must be the null string ("") and *facility* is one of those shown by `Message Listfacilities`.

Privilege Levels

- Administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

```
4 Message Countpolicies "" MESSAGE.SYSTEM.WELCOME
* 4 3
4 OK Completed
```

Get

Returns the currently set message for the specified message locale. If no message has been set in that message locale, returns an error. You can get the default message for a facility by specifying *locale* as null (“ “).

Syntax

tag `Message Get` *locale facility*

where:

- *locale* is one of those accepted by the `Message Set` command, or null string (“ “) to retrieve the default message.
- *facility* is one of those returned by the `Message Listfacilities` command.

Privilege Levels

- Administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

```
5 Message Get "" MESSAGE.SYSTEM.WELCOME
{213}
To: $(to)
From: Administrator@$(host)
Subject: Welcome Message
Mime-Version: 1.0
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 7bit

Welcome to your email account on $(host).

5 OK Completed
```

Getarg

Retrieves the specified override variable for a message in *locale* and *facility*.

Syntax

tag `Message Getarg` *locale facility variable*

where *locale* must currently be the null string (""), *facility* is one of those shown by `Message Listfacilities`, and *variable* is any substitutable tag accepted in a facility message; see `Message Set`.

Privilege Levels

- Administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

```
18 Message Getarg "" MESSAGE.JUNKMAIL.SUMMARY host
* 18 jmm.example.com
18 OK Completed
```

Getpolicy

Returns the current policy in all message locales for the specified facility.

Syntax

tag `Message Getpolicy` *locale facility* where *locale* must be the null string ("") and *facility* is one of those shown by `Message Listfacilities`. Policy may be `Off` or `On`.

Privilege Levels

- Administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

```
6 Message Getpolicy "" MESSAGE.SYSTEM.WELCOME
* 6 OFF
6 OK Completed
```

List

Returns the list of administrator-defined messages by message locale and facility.

Syntax

`tag Message List localepattern facilitypattern start count`

where *localepattern* and *facilitypattern* must be either the null string ("") or asterisk (*); *start* and *count* specify the beginning and duration points.

Privilege Levels

- Administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

```
13 Message List "" "" "" ""
* 13 en_US.iso-8859-1 FILTER.DISCLAIMER.CONFIDENTIAL
* 13 en_US.iso-8859-1 FILTER.NOTIFY.RAPIDAV
* 13 en_US.iso-8859-1 MESSAGE.SYSTEM.WELCOME
12 OK Completed
```

Listargs

Displays the specified override variable for a message in *locale* and *facility*.

Syntax

tag Message Listargs *locale facility variable start count*

where *locale* must currently be the null string (""), and *facility* is one of those shown by Message Listfacilities, and *variable* may be the null string ("") or asterisk (*) constrained by *start* and *count*.

Privilege Levels

- Administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

```
17 Message Listargs “ “ MESSAGE.JUNKMAIL.SUMMARY “ “ “ “ “ “
* 17 host
17 OK Completed
```

Listfacilities

Shows the current list of valid message facilities. The system welcome facility is for new users. Junk Mail Manager requires two facilities, welcome and summary. The XSS security feature, which may rewrite HTML message parts, is patch-enabled.

Syntax

tag Message Listfacilities *pattern start count*

where *pattern* must be either the null string ("") or asterisk (*); *start* and *count* specify the beginning and duration points.

Privilege Levels

- Administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

```
7 Message Listfacilities "" "" ""
* 7 MESSAGE.JUNKMAIL.SUMMARY
* 7 MESSAGE.JUNKMAIL.WELCOME
* 7 MESSAGE.SYSTEM.WELCOME
* 7 MESSAGE.XSS.WARNING
7 OK Completed
```

Listpolicies

Shows the list of policies for a facility in a given message locale.

Syntax

tag `Message Listpolicies` *locale facility*

where *locale* must be the null string ("") and *facility* is one of those shown by `Message Listfacilities`. Policy `On` means the facility is active, `Off` means inactive, and `Inherit` means the value originates from the primary domain.

Privilege Levels

- Administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

```
8 Message Listpolicies "" MESSAGE.SYSTEM.WELCOME
* 8 ON
* 8 OFF
* 8 INHERIT
8 OK Completed
```

Send

Sends the currently set message. If one does not exist, sends the message from the default message locale. If that does not exist, sends the default (English) message. This command bypasses domain filters, and may be used to test messages inside of delegated domains, or as a policy mechanism.

Syntax

tag `Message Send` *locale facility destAddr arguments*

where:

- *locale* is one of those accepted by the `Message Set` command. For facility `MESSAGE.JUNKMAIL.SUMMARY`, if a user has specified message locale preference on the **Junk Mail Manager > Preferences** page, it overrides this *locale*. So the junkmail summary appears in the user's preferred locale, if a summary was set for that *locale*; if not, the summary appears in the default *locale*.

- *facility* is one of those shown by `Message Listfacilities`.
- *destAddr* is the destination email address (user or distribution list) for this message. Multiple addresses may be given, separated by commas. If given as null string, this message is sent to the current user.
- *arguments* is a parenthesized list of variables and values, which can be used to set existing variable substitutions or add new values.

Privilege Levels

- Administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

This command sends a welcome message to Joe User with hostname filled in.

```
9 Message Send en_US.ISO_8859-1 MESSAGE.JUNKMAIL.WELCOME juser "(host=mail)"
* 9 AB000006
9 OK Completed
```

```
14 Message Send en_US.ISO_8859-1 MESSAGE.SYSTEM.WELCOME juser@m2.example.com
       "(host=m2)(password=secret11)"
* 14 AB000028
14 OK Completed
```

Set

For a given message locale, sets the specified facility's text message. Only one locale per facility may be created at a time with this command. If a locale already exists for a facility, even if different from the one given to this command, `Message Set` overwrites it.

Syntax

```
tag Message Set locale facility message
```

where:

- *locale* is one of the following (but omit left-of-dot in the message header):
  - default.utf-8
  - de_DE.iso-8859-1
  - en_US.iso-8859-1
  - es_MX.iso-8859-1
  - fr_FR.iso-8859-1

- it_IT.iso-8859-1
- ja_JP.iso-2022-jp
- ja_JP.shift-jis
- ko_KR.euc-kr
- pt_BR.iso-8859-1
- th_TH.tis-620
- zs_CN.gb2312
- zt_TW.big5

◆ `facility` is one of those listed by the `Message Listfacilities` command, or it can be a three-part facility with this format:
`FILTER.DISCLAIMER.`*`disclaimer_name`*

◆ `message` is text for the facility in this locale. It is specified on multiple lines as a counted string literal. In the Content-Type header, omit *language_Region* and specify only the right hand side, for instance "`text/html; charset=iso-2022-jp`". Here is a list of substitutable tags (variables) that can appear in all facilities:

- `$(action)`—Action taken by filter on the mail.
- `$(attachments)`—List of attachments separated by a delimiter.
- `$(date)`—Date.
- `$(domain)`—Current domain.
- `$(filtername)`—Filtername that triggered the notification.
- `$(host)`—Mail server hostname and domain.
- `$(pass)`—Password created for a user.
- `$(proxyhost)`—Host in the URL for junkmail and summary message.
- `$(sender)`—Original mail sender.
- `$(subject)`—Original mail subject.
- `$(to)`—Recipient in the To line.
- `$(urluser)`—User's name for HTTP.
- `$(user)`—User's login name.

Privilege Levels

◆ Administrator

◆ Domain administrator

◆ Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

```
10 Message Set en_US.iso-8859-1 MESSAGE.SYSTEM.WELCOME {212+}
To: $(to)
From: Administrator@$(host)
Subject: Account Created

Welcome to your new email account on $(host).
Your password has been set to $(password) initially.
```

```
Please change it for security reasons.
10 OK Completed
11 Message Set en_US.iso-8859-1 FILTER.NOTIFY.RAPIDAV {232+}
From: Virus Administrator
Subject: A potentially infected message has been held for examination

A message from $(sender) with subject "$(subject)" has been
determined to be possibly infected. It is being held for examination.
11 OK Completed
12 Message Set en_US.iso-8859-1 FILTER.DISCLAIMER.ORG {265+}
To: $(to)
Subject: Disclaimer
Mime-Version: 1.0
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 7bit

This email is intended only for the addressee shown.
It contains information that is confidential and protected from disclosure.
12 OK Completed
12 Filter Add "(domain=any)" Conf Disclaimer "(message=FILTER.DISCLAIMER.ORG)" allof stop {44+}
:envelopeto in-ldap-group "internal-users"
12 OK Completed
```

Setarg

Overrides variables (substitutable tags) in the message portion of `Message Set`. Once set, override occurs in the specified *locale* and *facility* until unset by specifying *value* as a null string.

Syntax

tag `Message Setarg` *locale facility variable value*

where:

- *locale* must currently be the null string ("").
- *facility* is one of those shown by `Message Listfacilities`.
- *variable* may be any substitutable tag accepted in `Message Set` messages.
- *value* is the override hostname for this *locale* and *facility*.

Privilege Levels

- Administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

```
15 Message Setarg "" MESSAGE.JUNKMAIL.SUMMARY proxyhost jmm.example.com
15 OK Completed
```

```
16 Message Setarg "" MESSAGE.JUNKMAIL.WELCOME proxyhost jmm.example.com
16 OK Completed
```

Setpolicy

Sets policy for the specified facility in a given message locale.

Syntax

tag `Message Setpolicy` *locale facility value*

where *locale* must be the null string ("") and *facility* is one of those shown by `Message Listfacilities`. Policy indicates whether to send messages; *value* may be set `On` or `Off`. The default policy is Off for `MESSAGE.SYSTEM.WELCOME`, but On for `MESSAGE.JUNKMAIL.WELCOME` if Junk Mail Manager is licensed.

Privilege Levels

- Administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Applies to delegated domain, or to primary domain if none is current.

Example

```
11 Message Setpolicy "" MESSAGE.SYSTEM.WELCOME On
11 OK Completed
```

The Model Command

The Model command reports information specific to your Mirapoint system model.

The output of this command consists of cryptic keywords used by the Mirapoint system software and administration client. The output is not fully descriptive.

Subcommands

Get

Responds (somewhat cryptically) with the value of the specified parameter.

Syntax

`tag Model Get parameter`

where parameter is one

- `Apptype`—indicates the application type of the system, usually `MIR`.
- `Chassis`—indicates the hardware chassis type (the model number).
- `Compliance`—indicates the hardware compliance (RoHS for example).
- `CPU`—indicates the processor type and speed (MHz) in the system.
- `Enet`—indicates the number of Ethernet ports on the system.
- `Mem`—indicates the memory configuration (in MB) of the system.
- `Raid`—indicates the RAID configuration used by the system.
- `Serial`—reports the serial number on M400(0) and later systems.
- `Storage`—indicates the type of disk enclosure on the system.

Response

The response line contains a cryptic string that is useful only to the system software, the administration client application, and Mirapoint manufacturing.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
1 Model Get Storage
* 1 E01UW7
1 OK Completed

2 Model Get Mem
* 2 512
2 OK Completed

3 Model Get Storage
* 3 E02FC14
3 OK Completed
```

The Mon Command

The `Mon` command lets you clear system alerts, configure the alert thresholds for certain monitored parameters, and view the status of system alerts.

Subcommands

Clear

Stops (clears) alerts for persistent fault conditions, including but not limited to:

- Motherboard temperature too high (`SYSTEM.TEMP`)
- Motherboard voltage out-of-range (`SYSTEM.VOLTAGE`)
- CPU cooling fan failure (`SYSTEM.FANC`)
- Motherboard cooling fan failure (`SYSTEM.FANMB1` and `SYSTEM.FANMB2`)
- Front-panel keypad (touchpad) and LCD panel failure (`SYSTEM.TOUCH`)
- Main system chassis power supply failure (`SYSTEM.POWER`)

Syntax

tag `Mon Clear`

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Mon Clear
2 OK Completed
```

Disable

Turns off network server monitoring, which pings remote systems to determine response time. This is disabled by default; see `Mon Enable`.

Syntax

tag `Mon Disable` *parameter*

where *parameter* must currently be `Netmonping`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Mon Disable Netmonping
6 OK Completed
```

Enable

Turns on network server monitoring, which pings remote systems to determine response time; see `Mon Getthresh`. Network server monitoring is disabled by default because IP addresses are often unreachable due to ICMP filtering by firewalls.

Syntax

tag `Mon Enable` *parameter*

where *parameter* must currently be `Netmonping`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Mon Enable Netmonping
5 OK Completed
```

Enabled

Check status of network server monitoring.

Syntax

tag `Mon Enabled` *parameter*

where *parameter* must currently be `Netmonping`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Mon Enabled Netmonping
* 4 NO
4 OK Completed
```

Getthresh

Responds with the alert threshold value of the specified system statistic.

Syntax

`tag Mon Getthresh parameter`

where *parameter* is one of the following system statistics:

- `SYSTEM.ADMINC`—The minimum number of administration server connections that triggers a “too many connections” email alert.
- `SYSTEM.DNS*RESP`—DNS server(s) response time to trigger an alert.
- `SYSTEM.IMAPC`—The minimum number of IMAP connections that triggers a “too many connections” email alert.
- `SYSTEM.KERB4*RESP`—Kerberos 4 server(s) response time to trigger an alert.
- `SYSTEM.KERB5*RESP`—Kerberos 5 server(s) response time to trigger an alert.
- `SYSTEM.LDAP*RESP`—LDAP server(s) response time to trigger an alert.
- `SYSTEM.LEGATORESP`—Deprecated.
- `SYSTEM.LMRRESP`—SMTP LMR response time to trigger an alert.
- `SYSTEM.LOAD`—CPU load that triggers a “processor load too high” email alert. Measured as the system run-queue averaged over the past minute. The default threshold is 65.
- `SYSTEM.NIS*RESP`—NIS server(s) response time to trigger an alert.
- `SYSTEM.NTP*RESP`—NTP server(s) response time to trigger an alert.
- `SYSTEM.OMRRESP`—SMTP OMR response time to trigger an alert.
- `SYSTEM.POPC`—The minimum number of POP connections that triggers a “too many connections” email alert.
- `SYSTEM.RADIUS*RESP`—Radius server(s) response time to trigger an alert.
- `SYSTEM.RBL*RESP`—RBL server(s) response time to trigger an alert.
- `SYSTEM.ROUTERRESP`—Default gateway response time to trigger an alert.

- SYSTEM.SMTPC—The minimum number of SMTP connections that triggers a "too many connections" email alert.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
4 Mon Getthresh SYSTEM.ADMINC
* 4 25
4 OK Completed
```

Setthresh

Sets the alert threshold value for the specified system statistic.

Syntax

```
tag Mon Setthresh parameter value
```

where:

- *parameter* is one of the following system statistics (use * to set the threshold for servers, which may be numbered from 1 to 9):
  - SYSTEM.ADMINC—The number of administration server connections that triggers a "too many connections" alert. The default is usually 100.
  - SYSTEM.DNS*RESP—DNS server(s) response time to trigger an alert.
  - SYSTEM.IMAPC—The number of IMAP server connections that triggers a "too many connections" alert. The default is 2000.
  - SYSTEM.KERB4*RESP—Kerberos 4 server(s) response time to trigger alert.
  - SYSTEM.KERB5*RESP—Kerberos 5 server(s) response time to trigger alert.
  - SYSTEM.LDAP*RESP—LDAP server(s) response time to trigger an alert.
  - SYSTEM.LEGATORESP—Deprecated.
  - SYSTEM.LMRRESP—SMTP LMR response time to trigger an alert.
  - SYSTEM.LOAD—CPU load that triggers a "processor load too high" email alert. Measured as the system run-queue averaged over the past minute. The default threshold is 65.
  - SYSTEM.NIS*RESP—NIS server(s) response time to trigger an alert.
  - SYSTEM.NTP*RESP—NTP server(s) response time to trigger an alert.
  - SYSTEM.OMRRESP—SMTP OMR response time to trigger an alert.
  - SYSTEM.POPC—The number of POP server connections that triggers a "too many connections" alert. The default is 200.
  - SYSTEM.RADIUS*RESP—Radius server(s) response time to trigger an alert.
  - SYSTEM.RBL*RESP—RBL server(s) response time to trigger an alert.

    - SYSTEM.ROUTERRESP—Default gateway response time to trigger an alert.
    - SYSTEM.SMTPC—The number of SMTP server connections that triggers a “too many connections” alert. This is a dynamic threshold based on system memory and number of processors.

- *value* is the setting for *parameter*. In the case of RESP parameters, this is the threshold (in milliseconds) that triggers monitoring, or zero (0) to disable monitoring. RESP parameters are usually disabled by default, but when enabled, the default setting is 2000 milliseconds. When a server is down, value may appear as -1, which always triggers monitoring.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Mon Setthresh SYSTEM.ADMINC 25
5 OK Completed
```

Status

Shows uncleared system alerts.

Syntax

```
tag Mon Status
```

The output contains three fields: name of the alert, timestamp showing seconds and time since boot as in the SYSTEM.UPTIME statistic, and a detailed error string.

Privilege Levels

Administrator

Domain Sensitivity

Must be run in the top-level domain.

Example

```
3 Mon Status
* 3 SYSTEM.FANC "1174 19m34s" "Main CPU fan is not functioning"
* 3 SYSTEM.CHASTEMP "1254 20m54s" "System is too hot."
3 OK Completed
```

40

The Mtaverify Command

Mirapoint MailHurdle is an antispam subsytem that takes advantage of the SMTP 451 error code indicating “please retry this recipient later.” Most spam mailers pay no attention to this retry code, and do not try sending again. If a mailer retries the message within a certain timeframe, SMTP delivers it with a 250 success code.

In order to distinguish between mailers that are asked to retry, and mailers whose messages can be delivered immediately, MailHurdle keeps track of triplets. A **triplet** is a combination of the remote mail server’s IP address, the envelope From address, and the envelope Recipient address(es). The second and third elements may be wildcards. Triplets are stored in a database, managed by the `Mtaverify` command. Successful triplets permit immediate delivery.

The MailHurdle facility requires a license, often bundled with an Antispam license. Without a license, the `Mtaverify` command is hidden and cannot be used.

You enable triplet checking with the `Smtp Set Mtaverify` command. You enable whitelisting, blacklisting, and recipient whitelisting by SMTP (during `RCPT TO`) with `Smtp Set` command choices `UceWhitelist`, `UceBlacklist`, and `UceWhitelistTo`. You set up these whitelists and blacklists with the `Uce Addexception` command.

Multiple message-router clients can access one central triplet database after trust relationships have been established with `Trustedhost Add`. Or you can distribute the triplet database across multiple servers by doing `Mtaverify Add` multiple times.

Based on default settings of `AllowRelays` and `InboundOnly`, SMTP-authenticated and local users are not subject to MailHurdle checking of outbound mail. Wildcard triplet elements are set by `AllowEntireIp`, enabled by default.

MailHurdle is currently very effective, but efficacy might diminish as spammers use more sophisticated (retrying) mailers. One disadvantage of MailHurdle is that it imposes a delay on messages with unrecognized triplets. As the triplet database grows, the frequency of delayed messages decreases. Recipients who must receive messages quickly (such as support personnel) can be whitelisted to avoid delays.

Mtaverify Open Issues

Most SMTP services send email and, if they receive a please-retry 451 error code, attempt to redeliver in 30-40 minutes, then back off for longer periods until giving up after some number of retries. `Mtaverify` defaults (initial timout of 5 minutes and initial lifetime of 24 hours) were chosen as a compromise between SMTP services that rapid-fire several retries within minutes, and other SMTP services that wait for 12 hours before the first retry.

If a message has multiple recipients, two of which resolve to the same user address, two copies may get delivered, one before the other, if the first is from a recognized triplet while the second is not. Ordinarily the copies are considered duplicates and SMTP delivers only one.

`Mtaverify` can cause an odd time-warp effect if one message arrives without a valid triplet, soon followed by another message with the same triplet. The later message is delivered immediately, because the first one primed the triplet database, whereas the earlier message is delayed until sender retry, which could be many hours later.

Some SMTP services notify their senders if messages are delayed by the please-retry 451 error code. This may confuse senders and temporarily clog the mail queue if a non-triplet site suddenly sends a lot of messages to an `Mtaverify` server.

There are a number of well-respected mailers that use the "fire and forget" method of sending mail, just like spammers. Some examples are Amazon, Ebay, and Yahoo groups. The `AllowMisbehavingMailers` option is set On by default to allow these mailers to deliver mail.

Some large ISPs use a pool of outbound SMTP servers to deliver from a centralized queue. This can cause triplet variation and unlimited delays due to differences in the sending IP address. The `ReverseMx` option is On by default to avoid this problem.

Distribution lists using VERPs (variable envelope return paths) may get delayed, although `Mtaverify` tries to avoid this (see http://cr.yp.to/proto/verp.txt).

Subcommands

Add

Establishes an active `Mtaverify` server, given its host name or IP address. You can distribute the triplet database across servers by adding multiple `Mtaverify` servers (up to 1024); add the same set of servers on all participating hosts. If one of the multiple servers is down, its portion of the triplet database is missing, but not the portions on other servers, reducing effectiveness proportionally.

Alternatively you can add MailHurdle clients, which use but do not manage the triplet database, by having `Mtaverify Add` point to the active `Mtaverify` server(s).

For MailHurdle clients in a multi-tier environment, run `Mtaverify Add localhost` on the server that manages the triplet database, then `Mtaverify Add` *`MtaServer`* on all MailHurdle clients. Also run `Trustedhost Add` to establish trust relationships: on the *`MtaServer`* only, add all MailHurdle clients to the trusted list.

To spread the triplet database across multiple servers, run `Mtaverify Add` *`MtaHostN`* (not `localhost`) on each server, for that server and all other servers. SMTP service distributes the triplet database across all MailHurdle servers that you have added. Once you create the list, adding another MailHurdle server invalidates existing entries in the triplet database. Run `Trustedhost Add` for each participating remote MailHurdle server. To distribute MailHurdle across two hosts:

```
mh1.example.com> Mtaverify Add mh1.example.com
mh1.example.com> Mtaverify Add mh2.example.com
mh1.example.com> Trustedhost Add Mtaverify mh2.example.com ""
mh1.example.com> Smtp Set Mtaverify On

mh2.example.com> Mtaverify Add mh1.example.com
mh2.example.com> Mtaverify Add mh2.example.com
mh2.example.com> Trustedhost Add Mtaverify mh1.example.com ""
mh2.example.com> Smtp Set Mtaverify On
```

Syntax

tag `Mtaverify Add` *host-spec*

where *host-spec* is the host name or IP address of the server that should keep the triplet database for use by MailHurdle antispam.

Privilege Levels

Administrator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
2 Smtp Set Mtaverify On
2 OK Completed

3 Mtaverify Add host1.example.com
3 OK Completed
17 Mtaverify Add host2.example.com
17 OK Completed
18 Mtaverify List "" "" ""
* 18 host2.example.com
* 18 host1.example.com
18 OK Completed
```

Addmisbehavingmailer

Adds an IP address to the list of mailers exempted from `Mtaverify` processing; see `Mtaverify Set AllowMisbehavingMailers`. The `Uce` rulegroup **Mtaverify** already contains a list of misbehaving mailers; this is in addition to that.

`Addmisbehavingmailer` is similar to whitelisting, but is specified by IP address (or range of addresses) rather than by email address or domain name.

Syntax

tag `Mtaverify Addmisbehavingmailer` *IPaddr/netmask*

where *IPaddr* is a network address in dotted-quad notation, and *netmask* is the network mask. To specify a single host with misbehaving mailer, use netmask `/32`. To specify an entire class C network of misbehaving mailers, use netmask `/24`.

Privilege Levels

Administrator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
11 Mtaverify Addmisbehavingmailer 1.2.3.4/32
11 OK Completed

12 Mtaverify Addmisbehavingmailer 2.3.4.0/24
12 OK Completed
```

Count

Counts the number of active `Mtaverify` servers.

Syntax

`tag Mtaverify Count pattern`

where *pattern* must be "" or *, both of which match everything.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
4 Mtaverify Count ""
* 4 1
4 OK Completed
```

Countmisbehavingmailer

Counts the number of addresses currently set by `Addmisbehavingmailer`.

Syntax

`tag Mtaverify Countmisbehavingmailer pattern`

where *pattern* must be "" or *, both of which match everything.

Privilege Levels

Administrator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
13 Mtaverify Countmisbehavingmailer *
* 13 2
13 OK Completed
```

Delete

Removes an active `Mtaverify` server. The triplet database remains in place, although it no longer gets updated.

Syntax

```
tag Mtaverify Delete host-spec
```

where *host-spec* is the host name or IP address of the server that was keeping the triplet database for MailHurdle antispam.

Privilege Levels

Administrator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
10 Mtaverify Delete localhost
10 OK Completed
```

Deletemisbehavingmailer

Deletes an IP address (or range of addresses) from the list of mailers previously set by `Mtaverify Addmisbehavingmailer`.

This command must be followed by the `Mtaverify Flush *` command. Otherwise, deletions are not recognized by the system until SMTP service is restarted.

Syntax

```
tag Mtaverify Deletemisbehavingmailer IPaddr/netmask
```

where *IPaddr* is a network address in dotted-quad notation, and *netmask* is the network mask. To specify a single host with misbehaving mailer, use netmask `/32`. To specify an entire class C network of misbehaving mailers, use netmask `/24`.

Privilege Levels

Administrator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
15 Mtaverify Deletemisbehavingmailer 1.2.3.4/32
15 OK Completed
```

Flush

Empties all or part of the triplet database, if the local host is an `Mtaverify` server.

Syntax

tag `Mtaverify Flush` *type*

where *type* may be "" or * to flush all triplets, or `Expired` to flush only triplets that have exceeded the `InitialEntryLifetime` or `AllowedEntryLifetime` timeouts (see `Mtaverify Set` for an explanation of timeouts).

Privilege Levels

Administrator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
5 Mtaverify Flush Expired
5 OK Completed
```

Get

Shows parameters of `Mtaverify` operation.

Syntax

tag `Mtaverify Get` *parameter*

where *parameter* is one of those specified under `Mtaverify Set`.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
8 Mtaverify Get InitialTimeout
* 8 1h
8 OK Completed
```

List

Displays information about the active `Mtaverify` servers.

Syntax

`tag` `Mtaverify List` *`pattern start count`*

where:

- *`pattern`* must be "" or *, both of which match everything.
- *`start`* is the number of the first item, and *`count`* is the number of items to list.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
4 Mtaverify List "" "" ""
* 4 host.example.com
* 4 localhost
4 OK Completed
```

Listmisbehavingmailer

Displays the addresses currently set by `Mtaverify Addmisbehavingmailer`.

Syntax

tag Mtaverify Listmisbehavingmailer *pattern start count*

where *pattern* must be "" or *, both of which match everything; *start* is the number of the first item, and *count* is the number of items to list.

Privilege Levels

Administrator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
14 Mtaverify Listmisbehavingmailer "" * *
* 14 "1.2.3.4/32"
* 14 "2.3.4.0/24"
14 OK Completed
```

Set

Controls parameters of Mtaverify operation.

After over a year of experience in the field, the following changes to default settings are often recommended at initial installation time. After that point, changing these settings would wipe out the previous MailHurdle triplet database. These alterations seem to maximize catch rate while minimizing delayed email:

```
Mtaverify Set AllowEntireIP off
Mtaverify Set Ignorerecipients on
Smtp Set UceSuspectlist on
Uce Setoption Suspectlistthreshold (domain=any) 50
Mtaverify Set Checksuspectlist on
```

Syntax

tag Mtaverify Set *parameter setting*

where:

- *parameter* is one of the following:
  - AllowedEntryLifetime——Lifetime of triplets for which mail was allowed to pass. During this time period, all mail from a particular triplet is allowed; any such mail resets the entry's lifetime. *Setting* is a timeframe (see below). Default active entry lifetime is 36 days. Also called Active in the GUI.
  - AllowEntireIP——If On, allow all triplets from a given IP address once one triplet from that IP address has been successfully allowed. Default is On. This basically amounts to ignoring the sender portion of the triplet. The exempted IP addresses expire with the normal timeout values specified by AllowedEntryLifetime. This setting must made on *both* servers and clients. If enabled, triplets will be hashed onto multiple servers based on their IP addresses rather than the entire triplet.

- `AllowMisbehavingMailers`——If `On`, allows a fixed list of known good SMTP servers to be exempted from `Mtaverify` processing. Default is `On`.
- `AllowNullFrom`——If `On`, this setting allows mail with an envelope sender of "<>" to be passed through immediately, without the usual timeouts. Default is `On`. This setting is useful for communicating with SMTP services that verify senders by issuing a `RCPT-TO` request with null envelope sender. Some SMTP services say envelope sender is Postmaster. For those services, add `postmaster` to your whitelist and run `Smtp Set Ucewhitelist On`.
- `AllowRelays`——Controls whether server addresses or envelope senders that would normally be allowed to relay get `Mtaverify` treatment or not. Envelope recipients are normally allowed to relay. Default is `On`. If set `Off`, servers in the system's relay list are required to retry before being allowed to pass mail. The `AllowRelays` exemption also applies to authenticated users, login-before-SMTP, and mail from localhost.
- `Checksuspectlist`——Enables or disables honoring the suspect list. Off by default. When turned On, MailHurdle imposes a delay only for IP addresses that are on the suspect list. When enabled by `Smtp Set UceSuspectlist`, the suspect list is automatically maintained by Antispam scanning. You can manually add IP addresses to the suspect list with `Uce Addexception`, and remove them from it with `Uce Deleteexception`.
- `Exemptonspf`——*Mode* replaces *Setting* and can be one of the following:
  - `None`——(the default) gives the default MailHurdle behavior.
  - `Spfpass`——MailHurdle will exempt senders who pass an SPF enforcement check.
  - `Spfnotfail`——MailHurdle will exempt senders who pass an SPF enforcement check, senders who do not have an SPF record, and senders who receive a NEUTRAL from an SPF enforcement check.
- `Ignorerecipients`——Enable or disable inclusion of the mail recipient(s) during MailHurdle processing. Off by default. When turned On, `Mtaverify` considers only the source IP address and the envelope From address. New triplets are treated like any other, the third field is just ignored. Old triplets containing unique recipient(s) in the third field will age out of the database. Turning this option On improves performance, decreases size of the triplet database, and prevents additional delay when people are added to a Cc list, all without significantly reducing catch rate.
- `InboundOnly`——If `On`, applies `Mtaverify` timeouts to inbound mail only, as determined by a current local check for domain filters. If `Off`, outbound mail is also verified and subject to timeout. Default is `On`. Especially when combined with `Smtp Set Recipientcheck`, this feature can greatly reduce the number of triplets that the `Mtaverify` server must process.
- `InitialEntryLifetime`——Lifetime of triplets that are not yet retried, and have not been passed. After this period, the triplet is purged from the database, so it will have to start again with `InitialTimeout` before being allowed to send mail. If a message gets resent while in this state, its triplet is passed, and is then subject to rules described by `AllowedEntryLifetime`. *`Setting`* is a timeframe (see below) which may not be less than `InitialTimeout`. Default entry lifetime is 12 hours. Also called InitialActive in the GUI.
- `InitialTimeout`——Amount of time that mail from a previously unknown triplet initially gets delayed (asked to retry). After this timeout period,

SMTP begins accepting mail from the triplet. *Setting* is a timeframe (see below). Default timeout is 5 minutes. Also called InitialDeny in the GUI.

- `ReverseMx`——Controls how the incoming IP address is turned into the hostname that is stored in the triplet database. The system performs a reverse PTR lookup via DNS to find the hostname. By default, it takes the most significant parts of the fully-qualified domain name (FQDN) and stores them in the triplet database as the source hostname. (Despite its name, ReverseMX does not look up MX records; the default number of significant parts is determined by the suffix , such as .com or .jp.) If `Off`, the hostname is used as is. If `On`, the most significant parts of the hostname are stored. If `Complete`, multiple DNS lookups are done to determine the number of significant parts in the hostname, and those parts are stored.

- *setting* could be `On` or `Off`, or it could be a timeframe: a number followed by a letter, such as `D` for days, `H` for hours, or `M` for minutes. Timeouts and lifetimes are not exact. The actual timeout will occur between the time specified and one time unit later. For instance, if you specify `5m`, timeout occurs 5-6 minutes after. If you specify `2h`, timeout occurs 2-3 hours later. Timeframe specifications are limited to 59 seconds, 59 minutes, 23 hours, and 60 days.

Privilege Levels

Administrator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
6 Mtaverify Set InitialTimeout 1h
6 OK Completed

7 Mtaverify Set InboundOnly On
7 OK Completed
```

Test

Tests the given triplet against the current state of the database. Whereas SMTP checks against the triplet database may change the state of a triplet, this test does not.

Syntax

tag `Mtaverify Test` *IPaddress sender recipient*

where mail server *IPaddress*, envelope *sender* address, and envelope *recipient* address constitute the triplet to be tested. If the `Mtaverify` server is not localhost, the test request is sent to a remote server.

Response

The tagged response displays the following fields:

- Normalized IP address.
- Normalized sender address.
- Normalized recipient address.
- Triplet state, one of the following:
  - `UNKNOWN`——The requested triplet is not yet in the triplet database.
  - `INITIAL-DENY`——This is the first time we have seen this triple. Request a mail retry (code 451), and start `InitialTimeout`. Once the timeout elapses, move to `INITIAL-ACTIVE` state and wait for the triplet to retry.
  - `INITIAL-ACTIVE`——We are waiting for the triplet to retry, because we want to deliver the mail. If we see the triplet again, advance it to `ACTIVE` state, notify success to the sender (code 250) and deliver.
  - `INITIAL-EXPIRED`——We finally gave up waiting for a retry of this triplet. If we see this triplet again, it goes back into `INITIAL-DENY` state.
  - `ACTIVE`——We like this triplet, it is current, and we deliver its mail.
  - `EXPIRED`——The triplet has expired. It was `ACTIVE` for a while, and we delivered mail for it, but it has not sent mail in a long time. If we see this triplet again, it goes back into `INITIAL-DENY` state.
  - `BYPASS`——Some combination of mail server settings (such as sender or recipient whitelist), or the current state of SMTP service, indicates that we are going to accept this triplet and deliver mail immediately, without even contacting the `Mtaverify` server.
  - `SERVER-FAILURE`——`Mtaverify` server returned a general processing error.
  - `NETWORK-ERROR`——Trouble communicating with the `Mtaverify` server.
- Triplet state expiration time, if applicable.
- Bypass reason, if applicable.
- General network error string, if applicable.
- Status for each server that owns the triplet if more than one server has been specified.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
9 Mtaverify Test "10.0.11.15" "friend@corp.net" "juser@example.com"
* 9 "10.0.11.15"
* 9 "friend@corp.net"
* 9 "juser"
* 9 "State ACTIVE"
* 9 "Expires 33d20h47m52s"
* 9 "Reason NA"
* 9 "Error NA"
* 9 "Server 10.1.1.2 up"
* 9 "Server 10.1.1.3 up"
* 9 "Server 10.1.1.4 down"
9 OK Completed
```

The Ndmp Command

The `Ndmp` command controls behavior of the Network Data Management Protocol (NDMP) backup service.

Index history generation is required to permit selective restore from image backup. It is controlled by the `HIST=Y` setting in Veritas NetBackup and Legato NetWorker, by the `TOC=Y` setting in Tivoli Storage Manager, and by the "Save File Information" checkbox in BakBone NetVault. The DMA connection may timeout while waiting for index history generation. Consult the documentation for your specific DMA to get details on increasing the timeout.

As you gain experience with the time required for image backups, schedule them in a sufficiently long backup window.

Subcommands

Clear

Truncates the current `Logprotocol` files, but does not affect settings controlled by the `Ndmp Set Logprotocol` command.

Syntax

```
tag Ndmp Clear Logprotocol
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
11 Ndmp Clear Logprotocol
11 OK Completed
```

DeleteTs

Deletes a timestamp for NDMP-based backup. This removes the Mirapoint system's backup record, but does not affect saved data on the NDMP client. Timestamps are reported using the same time and date format as the `Ntp Get Date` command.

Deleting a timestamp from the Mirapoint system has no effect on which level the Data Management Application (DMA) requests when it does a backup. Nor does deleting a timestamp affect which level backup the Mirapoint NDMP services report. The level of backup performed, logged, or otherwise reported, is always that requested by the DMA.

When a backup is performed, the timestamp used to determine which files to save is the most recent lower-level backup. If level 0 through 4 backups were performed, in order, and the next backup is a level 5, the timestamp for the level 4 backup is used (most recent lower-level backup). If you delete the level 4 timestamp with `Deletets`, when the level 5 is requested, the timestamp for level 3 is used. If you delete the level 0 timestamp with `Deletets`, performing a level 5 backup will be based off the level 4 timestamp because it remains the most recent lower-level backup.

Of course, deleting timestamps could affect how much data gets backed up. If this command is used to delete all timestamps and the DMA then requests a level 4 backup, the timestamp used would be that of the epoch (Jan 1, 00:00 1970 UTC), so all data in the message store is selected for backup.

Syntax

`tag` `Ndmp Deletets` *`backupType backupLevel`*

where:

- *`backupType`*—May be `Image` or `File` (the words `Message`, `Tar`, and `Dump` are accepted as synonyms for `File`).
- *`backupLevel`*—A number between 0 (zero) and 255, or the keyword `All` to indicate all possible levels.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
6 Ndmp Getts file 0
* 6 level 0 "Feb 23 09:49:28 2001"
6 OK Completed
7 Ndmp Deletets file 0
7 OK Completed
8 Ndmp Getts file 0
```

```
8 OK Completed
```

Estimatesize

Reports the estimated size (in bytes) of an NDMP backup of a given type and level.

Syntax

tag `Ndmp Estimatesize` *backupType backupLevel*

where:

- *backupType*—May be `Image` or `File` (the words `Message`, `Tar`, and `Dump` are accepted as synonyms for `File`).
- *backupLevel*—A number between 0 (zero) and 255, or the keyword `All` to indicate all possible levels. The most recent lower-level timestamp is used to estimate the size of a backup. For instance, if you specify level 1, timestamp is taken from the last level 0 (full) backup.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
9 Ndmp Estimatesize image 0
* 9 1234773229
9 OK Completed
```

Export

Exports the log file. If the argument is empty string "" it means to use standard administration protocol syntax. If the argument is a URL, data goes over the net.

Syntax

tag `Ndmp Export Logprotocol` *destination*

where *destination* may be "" to get the log file as a string literal, or a valid URL to have log data placed in an FTP or HTTP location.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

Assuming anonymous FTP is allowed, then assuming it is not.

```
13 Ndmp Export Logprotocol ftp://archive/pub/logNDMP-001
13 OK Completed

14 Ndmp Export Logprotocol ftp://user:passwd@archive/logNDMP-002
14 OK Completed
```

Get

Responds with the value of the specified NDMP parameter.

Syntax

tag `Ndmp Get` *parameter*

where *parameter* is one of those documented under `Ndmp Set`, or the following:

- `Softwareversion`—Returns the version number of the NDMP server running on the current system. This version number is useful for long-term validation of Mirapoint NDMP service with an approved DMA.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
3 Ndmp Get Softwareversion
* 3 Mirapoint NDMP software version 2.1.50
3 OK Completed
```

Getts

Reports the currently stored backup timestamps for all NDMP-based backups. Timestamps are reported using the same time and date format as the `Ntp Get Date` command. There is no provision for recording timestamps of the protocol-based `Backup` command.

Syntax

tag `Ndmp Getts` *backupType backupLevel*

where:

- *backupType*—May be `Image` or `File` (the words `Message`, `Tar`, and `Dump` are accepted as synonyms for `File`).
- *backupLevel*—A number between 0 (zero) and 255, or the keyword `All` to indicate all possible levels. The Data Management Application (DMA) usually ignores numbers above 9.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
4 Ndmp Getts image 0
* 4 level 0 “Feb 23 09:49:28 PDT 2001”
4 OK Completed

5 Ndmp Getts file all
* 5 level 0 “Aug 11 09:47:37 PDT 2001”
* 5 level 4 “Feb 18 09:47:47 PDT 2001”
* 5 level 9 “Feb 19 09:47:55 PDT 2001”
5 OK Completed
```

Merge

The `Merge` command supports selective mailbox restore from image backup. You can display merge status, abort the merge, or clear status from the last merge.

Syntax

tag `Ndmp Merge` *action*

where *action* is one of:

- `Abort`—Stop the merge. During a “Selective Folder Restore from Image” data get staged in a temporary directory hierarchy by the NDMP data service (NDS). After NDS completes the transfer of data, it starts to merge the restored data into its original location and notifies the Data Management Application (DMA) that the restore has completed (it must do this since most DMAs wait only a short time after detecting lack of data movement before calling the restore failed). If you decide not to continue the merge, this command stops it. Once stopped, there is no restarting the merge.
- `Clean`—Deletes merge-file data from system disks. Normally `Merge` cleans up after itself, but `Clean` might be useful if space is tight after `Ndmp Merge Abort`. This command fails if a merge is in progress.

- `Clear`—Removes merge status information. Because there is no good way of knowing when to discard status, the administrator is responsible for doing so. This command fails if a merge is in progress.
- `Status`—As mentioned above under `Abort`, after backup data are copied to the system, files must be merged into their original location. This command tracks the status of the merge. After completion of a merge, the entire status file is mailed to the `backup-status` DL.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
12 Ndmp Merge Clear
12 OK Completed
```

Set

Sets the value of the specified NDMP parameter.

Syntax

tag `Ndmp Set` *parameter value*

where *parameter* is one of the following, and *value* is the setting for *parameter*.

- `Dironly`—Allows for the option of recovering by directory only (mailboxes or folders) instead of individual files. Possible settings for *value* are:
  - `No`—Default setting. Specifies that the ndmp image backup file history will contain all files and directories in the backup.
  - `Yes`—Specifies that the ndmp image backup file history will contain only file history for the directories in the backup. The DMA will only have directories to select from; however, all files and directories in the selected directory or directories and below will be recovered.
- `Dma`—Specifies the Data Management Application (DMA) driving NDMP service so that the Mirapoint system can behave accordingly. Possible settings for *value* are:
  - `Bakbone`—BakBone NetVault for NDMP.
  - `Default`—A recent version of NDMP.
  - `Legato`—Legato NetWorker for NDMP.
  - `Tivoli`—Tivoli Storage Manager for NDMP backup.
  - `Veritas`—Veritas NetBackup for NDMP.
- `Logprotocol`—Record for the administrator details of all NDMP protocol requests and responses. This setting takes effect on the next connection from the

DMA. Since some protocol methods are called millions of times, different `value` levels of protocol logging are provided:

- ❖ `High`—Causes all NDMP protocol requests and replies to be logged.
- ❖ `Low`—Does not log the following: `NDMP_FH_ADD_DIR`, `NDMP_FH_ADD_FILE`, `NDMP_FH_ADD_NODE`, `NDMP_FH_ADD_UNIX_DIR`, `NDMP_FH_ADD_UNIX_NODE`, `NDMP_FH_ADD_UNIX_PATH`, `NDMP_LOG_DEBUG`, `NDMP_LOG_FILE`, `NDMP_LOG_LOG`, and `NDMP_LOG_MESSAGE`.
- ❖ `Off`—Disable logging. This is the default.

- ◆ `Port`—The TCP port number on which NDMP service listens for requests. This can be any `value` between 0 and 65535. The default NDMP port is 10000. No attempt is made to check whether the port is being used by another service. There is usually no reason to reset port unless requested by support personnel (the ability to change port is required by some NDMP DMA vendors). When the port number is actually changed, this command prompts you to stop and restart NDMP service.
- ◆ `PreferredIp`—The Mirapoint system onto which backup traffic should flow. In NDMP protocol version 4 and higher, the Mirapoint system sends the DMA a list of IP addresses on which it will accept traffic; the preferred IP, if set, will be placed at the beginning of the list. Changes to the value take effect on the next new NDMP connection. Specify the value as an IP address in Internet standard dotted quad notation. Reserved addresses such as 127.0.0.1 are not accepted. This setting has no effect if the NDMP protocol version is set to 2 or 3.
- ◆ `Version`—Sets the maximum and also the default NDMP protocol version used by NDMP service. This is the protocol version returned to the DMA when a connection is first made and also the highest protocol version that the DMA may negotiate. A DMA can negotiate the use of any NDMP protocol version from 2 (the lowest version supported) to the version set by this command. Valid version numbers are 2, 3 (the usual default), and 4 (default for BakBone). The new value takes effect upon the next connection to NDMP service. Version 2 limits you to one IP address per system.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Ndmp Set Port 10000
2 OK Completed

15 Ndmp Set Preferredip 127.0.0.1
15 NO Invalid address

16 Ndmp Set Preferredip 10.2.3.4
16 OK Completed

17 Ndmp Set Preferredip port1
17 NO Invalid address
```

42 The Netif Command

The `Netif` command manages network interfaces on a Mirapoint system, including IP aliases, static routes, trusted IP addresses, network bindings, NIC failover, media, and security settings.

To change the IP address of a system's primary network interface, you must use the serial console menu, its VGA equivalent, or the LCD keypad if available. You cannot set the primary IP address using the `Netif` command.

Subcommands

Addalias

Adds an IP address alias (alternate IP address) to identify the specified network interface port.

Syntax

```
tag Netif Addalias portnum IPaddr/mask-bits
```

where:

- `port`*`num`* identifies the Ethernet port, as labeled on the system back panel. The possible values are `port0`, `port1`, `port2` and so on. This port must currently be bound (see `Netif Bind` on page 432).
- *`IPaddr`* is the alternate IP address you want to assign to the port.
- *`mask-bits`* is the number of bits in the network mask for this alias; for example, if *`mask-bits`* is `24`, the network mask is 255.255.255.0. Use mask bits `/16` for class B networks, `/24` for small subnets, and `/23` for larger subnets containing up to 512 addresses. Use `/32` if the address is on the same subnet as the first network address for this interface (as a alternative to DNS CNAME aliasing). Also use `/32` to specify single-host routing to a network that is not physically connected to the system interface port, for example 167.3.2.9/32 to participate in remote network 167.3.2.0/24.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Netif Addalias port0 192.168.2.2/24
2 OK Completed
```

Addlogical

Adds a logical network interface to the system for failover. If an active network interface fails, the logical standby interface is able to take over without affecting high-level system software. Newly created logical ports are empty (unassigned) until you bind them to a physical port; see `Bindlogical`. You cannot create more logical ports than physical ports existing on the system. Physical interfaces are designated `Port`*N* while logical failover interfaces are designated `Logical`*N*.

After `Addlogical`, it is typical to run `Bindlogical`.

Syntax

tag `Netif Addlogical` *logicalPort type*

where:

- *logicalPort* may be either a valid logical port or "" to select the first empty logical port, initially Logical0. Logical port names are in the format `logical`*N*, where *N* is the number of real interfaces on the machine.
- *type* must currently be `Failover`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
14 Netif Addlogical logical0 Failover
* 14 logical0 created
14 OK Completed
```

Addroute

Adds a static route into the network routing table. The Mirapoint system uses the specified gateway and destination address, instead of dynamic routes advertised by the normal gateway (IP router), to route packets. The effect on routes is immediate.

No more than 64 static routes can be added. The default route is not affected.

Syntax

```
tag Netif Addroute dest-IPaddr/mask-bits gateway-IPaddr options
```

where:

- `dest-IPaddr` is the destination address for this static network route. Destination address may not be 0.0.0.0 or any local IP address on the system.
- `mask-bits` is the number of network mask bits for this route. For example, if `mask-bits` is `24`, the network mask is `255.255.255.0`. If `mask-bits` is 32, it indicates a host route; anything other than 32 is a net route.
- `gateway-IPaddr` is the IP address of the gateway (router) for this static route.
- `options` must be "" and is ignored.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

This command sets up a static route to backup server 100.200.0.8 through gateway 10.0.0.2, helping reduce traffic on 10.0.0.1, the default gateway for dynamic routing.

```
2 Netif Addroute 10.200.0.8/32 10.0.0.2 ""
2 OK Completed
```

Addtrustedip

Specifies an IP address that is exempt from maximum connect-rate processing (see `Netif Set LimitTcpConnectRate`). This list is empty by default. Use this command repeatedly to add more trusted hosts, up to a maximum of 16.

Syntax

```
tag Netif Addtrustedip IPaddr
```

where `IPaddr` is the numeric IP address of a trusted host.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
11 Netif Addtrustedip 10.0.44.44
11 OK completed
```

Bind

Binds an Internet Protocol (IP) address to the specified Ethernet (logical) port.

Mirapoint series 4.5 and 5 systems have a built-in Gigabit Ethernet interface on port 1. Earlier systems had optional Gigabit Ethernet cards, usually on higher ports. To employ this faster interface, designate an IP address and create appropriate DNS entries for a host name and mail on this address. Then use `Netif Bind` to associate the Gigabit Ethernet interface with this host. Route all email traffic through this fast-interface host. For administration, slower port 0 should remain open and connected under the original host name.

The `Netif Bind` and `Unbind` commands cannot be invoked on the primary port. The IP address of the primary port can be changed only from the serial console menu, its VGA equivalent, or the LCD keypad if available.

Syntax

tag `Netif Bind port`*num IPaddr/mask-bits*

where:

- `port`*num* identifies the Ethernet port to which you want to bind the IP address, as labeled on the system back panel. Possible values are `port0`, `port1`, `port2` and so on (or `logical0` and so on).
- *IPaddr/mask-bits* combination where *IPaddr* is the IP address you want to assign the port and *mask-bits* indicates the number of bits in a network mask; for example, if *mask-bits* is 24, the network mask is 255.255.255.0.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Netif Bind port1 10.0.3.3/8
2 OK completed

22 Netif Bind logical0 10.0.11.18/16
22 OK completed
```

Bindings

Responds with a list of all the system's network interfaces and the IP addresses bound to them.

Syntax

tag `Netif Bindings`

Response

Each response line has the format:

`*` *tag* `port`*num IPaddr*/*mask-bits MACaddr speed:duplex maxSpeed*

where:

- `port`*num* identifies the Ethernet port, as labeled on the system back panel. The possible values are `port0`, `port1`, `port2` and so on (or `logical0` and so on).
- *IPaddr* is the IP address assigned to the port.
- *mask-bits* is the number of bits in a network mask; for example, if *mask-bits* is `24`, the network mask is `255.255.255.0`.
- *MACaddr* is the unique media access control (MAC) address of the Ethernet port.
- *speed:duplex* is the current negotiated media settings, for instance `100:FULL` for 100 Mbps full duplex, or `AUTO:AUTO` for automatic.
- *maxSpeed* is the maximum bit rate for this Ethernet interface.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Netif Bindings
* 3 port0  10.12.0.9/20   00:d0:b7:b7:5e:04  AUTO:AUTO
* 3 port1  10.99.99.98/24 00:03:47:25:40:e3  AUTO:AUTO
* 3 port2  unassigned/ 0  00:90:27:3a:ba:f1
3 OK completed
```

Bindlogical

Associates physical ports with a logical interface previously created by `Addlogical`.

The first physical port you bind to a logical interface should have an IP address assigned to it (see `Netif Bind`), although is not an error if unassigned. The second physical port you bind to a logical interface must be unassigned. If the first physical port is assigned to an address, the logical port inherits the network settings of that port. (The opposite occurs when the last physical port is unbound from a logical interface: the physical port inherits the network settings of the logical port.)

The primary Ethernet port should not be changed while it is bound to a logical port. Doing so alters settings for the system primary logical port (serial console, LCD).

Currently only two ports can be bound to a logical interface. The logical interface assumes the hardware address of the first interface assigned to it. The first interface bound must be the last interface unbound; see `Unbindlogical`.

Syntax

tag `Netif Bindlogical` *logicalPort physicalPort*

where *logicalPort* is the name previously created by `Netif Addlogical`, and *physicalPort* is the port name, for example `port0`, `port1`, or `port2`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
15 Netif Bindings
* 15 Port0 10.0.11.8/16 00:d0:b7:a9:52:f8 AUTO:AUTO(100:FULL)
* 15 Port1 unassigned/0 00:d0:b7:b9:f9:6a AUTO:AUTO(100:FULL)
* 15 Port2 unassigned/0 00:d0:b7:b9:f9:6b AUTO:AUTO(100:FULL)
15 OK Completed

16 Netif Bindlogical logical0 port0
16 OK

17 Netif Bindlogical logical0 port1
17 OK

18 Netif Bindings
* 18 Logical0 10.0.11.8/16 00:d0:b7:a9:52:f8 ""
* 18 Port0 linked logical0 AUTO:AUTO(100:FULL)
* 18 Port1 linked logical0 AUTO:AUTO(no-carrier)
* 18 Port2 unassigned/0 00:d0:b7:b9:f9:6b AUTO:AUTO(100:FULL)
18 OK Completed
```

Countaliases

Responds with the number of IP address aliases (alternate IP addresses) associated with the specified network interface port.

Syntax

tag `Netif Countaliases port`*num pattern*

where

- `port`*num* identifies the Ethernet port, as labeled on the system back panel. The possible values are `port0`, `port1`, `port2` and so on. This port must currently be bound (see `Netif Bind` on page 432).
- *pattern* is currently ignored.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Netif Countaliases port0 ""
* 3 2
3 OK Completed
```

Countlogical

Responds with the number of logical network interfaces on the system.

Syntax

`tag Netif Countlogical pattern`

where *pattern* may be * or "" to count all logical interfaces.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
19 Netif Countlogical ""
* 19 1
19 OK Completed
```

Countmedia

Responds with the number of possible media settings (see `Netif Set`).

Syntax

`tag Netif Countmedia pattern`

where *pattern* can be a port number, or "" to count all ports.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Netif Countmedia ""
* 4 5
4 OK Completed
```

Countroutes

Returns the number of static routes in the network routing table. This number includes the local route 0.0.0.0, so is always one or greater.

Syntax

tag `Netif Countroutes` *pattern*

where *pattern* must be "" or * to count all routes.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Netif Countroutes *
* 3 2
3 OK Completed
```

Counttrustedip

Count the IP address(es) exempt from denial-of-service processing.

Syntax

tag `Netif Counttrustedip` *pattern*

where *pattern* must be either * or "" match all trusted hosts.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
13 Netif Counttrustedip
* 13 1
13 OK completed
```

Deletealias

Deletes an IP address alias (alternate IP address) from the specified network interface port.

Syntax

`tag Netif Deletealias portnum IPaddr`

where:

- `portnum` identifies the Ethernet port, as labeled on the system back panel. The possible values are `port0`, `port1`, `port2` and so on. This port must currently be bound (see `Netif Bind` on page 432).
- `IPaddr` is the IP address alias you want to delete from the port.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Netif Deletealias port0 192.168.2.2/24
5 OK Completed
```

Deletelogical

Removes a logical network interface from the system. Failover to this logical port no longer occurs. You cannot delete a bound logical port; see `Unbindlogical`.

Syntax

`tag Netif Deletelogical logicalPort`

where `logicalPort` is the name previously created by `Netif Addlogical`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
26 Netif Deletelogical logical0
26 OK Completed
```

Deleteroute

Deletes a static route from the network routing table. The Mirapoint system then uses the default gateway to supply a dynamic route for this destination.

Syntax

tag `Netif Deleteroute` *destination-IPaddr/mask-bits*

where:

- *destination-IPaddr* must match the destination address for a static route previously set with `Netif Addroute`.
- *mask-bits* is the number of network mask bits for this route, as in `Addroute`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Netif Deleteroute 10.200.0.8/32
5 OK Completed
```

Deletetrustedip

Removes a host's exemption from denial-of-service processing (see `Netif Set`).

Syntax

tag `Netif Deletetrustedip` *IPaddr*

where *IPaddr* is the numeric IP address of a trusted host.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
11 Netif Deletetrustedip 10.0.44.44
11 OK completed
```

Get

Responds with the value of the specified parameter.

Syntax

tag `Netif Get` *parameter* `port`*num*

where:

- *parameter* is one of those specified under `Netif Set`.
- `port`*num* must be the null string ("") for most parameters, but for the `Media` parameter it identifies the Ethernet port, as labeled on the system back panel. The possible values are `port0`, `port1`, `port2` and so on.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Netif Get Media port0
* 5 AUTO:AUTO
5 OK Completed
```

Getdomain

Retrieves the default domain name associated with an unqualified address coming from a known IP address. `Conf Enable DerivedomainIP` enables this feature.

Syntax

tag `Netif Getdomain` *IPaddr*

where *IPaddr* is an available interface or alias address.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
28 Netif GetDomain 10.0.3.8
* 28 "test.example.com"
28 OK
```

Getlogical

Retrieves various parameters for logical network interfaces.

Syntax

tag `Netif Getlogical` *parameter logicalPort*

where:

- *parameter* is one of those described under `Netif Setlogical`.
- *logicalPort* is the logical interface name as created by `Netif Addlogical`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
21 Netif Setlogical Linkmon logical0
* 21 500
21 OK Completed
```

Listaliases

Responds with a list of IP address aliases (alternate IP addresses) for the specified network interface port.

Syntax

tag `Netif Listaliases port`*num pattern start count*

where:

- `port`*num* identifies the Ethernet port, as labeled on the system back panel. The possible values are `port0`, `port1`, `port2` and so on. This port must currently be bound (see `Netif Bind` on page 432).
- *pattern* is currently ignored.
- *start* is the first position in the list of IP address aliases that you want to see. The empty string (`""`) implicitly means `0`.
- *count* is number of lines from the list of IP address aliases that you want to see. The empty string (`""`) implicitly means all aliases. If *count* is greater than the

total number of IP address aliases for the specified port, `list` returns as many aliases as possible.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Netif Listaliases port0 "" "" ""
* 4 192.168.2.1/24
* 4 192.168.2.2/24
4 OK Completed
```

Listlogical

Displays information about all logical interfaces currently on the system.

Syntax

`tag` `Netif Listlogical` *`pattern start count`*

where *`pattern`* must be either * or "" to match all logical network interfaces; *`start`* and *`count`* specify beginning and duration.

Fields returned by `Listlogical` are: logical interface name, logical interface type, physical ports in the logical interface, active physical ports in logical interface, and a type-specific string, which for the `Failover` type, indicates this status information:

- `Uninitialized`—Interface is not yet configured.
- `Empty`—Interface is configured, but has no physical ports assigned to it.
- `Partial`—Interface is configured with a single alive network port.
- `Optimal`—Interface is configured and all network ports are alive.
- `Degraded`—Interface is configured and at least one network port is down.
- `Failed`—Interface is configured and all network ports are down.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
20 Netif Listlogical "" "" ""
```

```
* 20 logical0 “Failover” "port1 port2" "port1" "Optimal"
20 OK Completed
```

Listmedia

Responds with a list of possible media settings (see `Netif Set`).

Syntax

`tag Netif Listmedia pattern start count`

where:

- *pattern* can be a port number, or "" to list all ports.
- *start* indicates the first media setting you want to see. The empty string ("") implicitly means `0`.
- *count* indicates the number of media settings you want to see. See the example below. The empty string ("") implicitly means all media settings. If *count* is greater than the total number of media settings, `list` returns as many media settings as possible.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Netif Listmedia "" "" ""
* 6 AUTO:AUTO
* 6 100:FULL
* 6 100:HALF
* 6 10:FULL
* 6 10:HALF
* 6 1000:FULL
* 6 1000:HALF
6 OK Completed
```

Listroutes

List static routes in the network routing table. `Listroutes` always returns an entry for the (dynamic) default route, which shows the IP address of the default router.

Syntax

`tag Netif Listroutes pattern start count`

where:

- *pattern* may be "" or * to show all routes.
- *start* indicates the first route to list. The empty string ("") means 0.
- *count* indicates the number of routes to list. The empty string ("") means all.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Netif Listroutes * "" ""
* 4 default/0 10.0.0.1 ""
* 4 10.200.0.8/32 10.0.0.2 ""
4 OK Completed
```

Listtrustedip

Shows the IP address(es) exempt from denial-of-service processing.

Syntax

```
tag Netif Listtrustedip pattern start count
```

where *pattern* must be either * or "" match all trusted hosts; *start* and *count* specify beginning and duration.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
12 Netif Listtrustedip
* 12 10.0.44.44
12 OK completed
```

Set

Sets the value of a specified network interface parameter.

Syntax

```
tag Netif Set parameter portnum setting
```

where:

- *parameter* is one of:
  - BlackholeDuration—Specifies the number of seconds that an offending TCP-hog host remains in the LimitTcpConnectRate blackhole, after the MaxTcpConnectRate threshold has been crossed. Default is 900 seconds, lowest setting is 1, and highest is 2^31 (over 2 billion).
  - LimitTcpConnectCount—Setting this ON enables checking the count of TCP connections to help avoid denial-of-service attacks. Default is Off. When enabled, transmitted packets and incoming connections are dropped from any hosts exceeding the MaxTcpConnectCount limit. You can make hosts immune from the limit with Netif AddTrustedIP.
  - LimitTcpConnectRate—Setting this ON enables checking the rate of TCP connections to help avoid denial-of-service attacks. Default is Off. When enabled, transmitted packets and incoming connections are dropped if the MaxTcpConnectRate limit is exceeded. You can make hosts immune from the limit with Netif AddTrustedIP.
  - MaxTcpConnectCount—Specifies the maximum number of connections allowed for each peer address when LimitTcpConnectCount is on. After this value is exceeded, new connections are dropped until the count falls back below the maximum. Due to concurrency, a few more connections than the maximum may be allowed before throttling is enabled. Default is 50, lowest setting is 1, and highest is 2^31 (over 2 billion). You can make hosts immune from the limit with Netif AddTrustedIP.
  - MaxTcpConnectRate—Specifies the maximum number of connections allowed every 10 seconds when LimitTcpConnectRate is on. After this value is exceeded, new connections from that host are dropped for a period defined by BlackholeDuration. Default is 400, lowest setting is 1, and highest is 2^31 (over 2 billion). You can make hosts immune from the limit with Netif AddTrustedIP.
  - Media—Controls the speed and duplex settings of the specified port. Because altering this parameter terminates any open POP, IMAP, and SMTP connections, it is best to issue it when system activity is low.
  - Primaryport—Sets the primary port for this system. Value is anything reported by Netif Bindings (for example, Port0, Port1, Port2). Modifying the primary port also modifies the default route on a system. You must set Primaryport on the serial console CLI; it returns an error if attempted over a network connection. Furthermore you must use the serial console menu (or LCD keypad if available) to change the IP address of the primary port. Default value: Port0.
- For the LimitTcpConnectRate, MaxTcpConnectCount, and BlackholeDuration parameters, port*num* must be the null string (""). For Primaryport, port*num* comes in the next argument and this one must also be the null string. For the Media parameter only, port*num* identifies the Ethernet port, as labeled on the back panel of the system. Possible values are port0, port1, port2 and so on.
- *setting* should be port*num* for the Primaryport parameter, On or Off for parameters LimitTcpConnectCount and LimitTcpConnectRate, a numerical maximum for parameters MaxTcpConnectCount and MaxTcpConnectRate, or *speed:duplex* for the Media parameter, as follows:

- *speed* is one of:
  - `Auto`—Allows the MAC (Media Access Control) hardware to automatically select its speed,
  - 10—Forces MAC hardware to operate at 10 Megabits per second.
  - 100—Forces MAC hardware to operate at 100 Megabits per second.
  - 1000—Forces the MAC hardware to operate at 1 Gigabit per second.
- *duplex* is one of:
  - `Auto`—Allows the MAC hardware to automatically select full- or half-duplex operation.
  - `Full`—Forces the MAC hardware to operate in full-duplex mode.
  - `Half`—Forces the MAC hardware to operate in half-duplex mode.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
7 Netif Set LimitTcpConnectRate "" On
7 OK Completed
```

```
8 Netif Set MaxTcpConnectRate "" 800
8 OK Completed
```

```
9 Netif Set Media port0 Auto:Auto
9 OK Completed
```

Setdomain

Specifies the default domain name for any unqualified email address of users connected to an IP interface on this system. Both the domain name and IP interface (or alias address) must exist. This mapping is checked at login time to qualify unqualified logins for IMAP, POP, and WebMail users, if possible. You enable this feature with the `Conf Enable DerivedomainIP` command.

Syntax

```
tag Netif Setdomain IPaddr domainName
```

where:

- *IPaddr* is an available interface or alias address.
- *domainName* is a domain on the system or local network. Setting null string ("") deletes the *IPaddr*-to-*domainName* mapping for that interface.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
27 Netif Setdomain 10.0.3.8 test.example.com
27 OK
```

Setlogical

Configures various parameters for logical network interfaces.

Syntax

tag `Netif Setlogical` *parameter logicalPort value*

where:

- *parameter* is one of:
  - `Arpmon`—Sets ARP monitoring frequency in milliseconds. A value of 0 disables ARP monitoring. Default is disabled. This is not compatible with link monitoring, so `Linkmon` must be explicitly disabled before activating `Arpmon`. A link is considered down if a second ARP request times out.
  - `Arpmonaddr`—Sets the IP address to monitor with `Arpmon`. There is no default. An invalid or unset IP address disables ARP monitoring. Currently only supports one IP address, so a comma-separated list is an error.
  - `Failover`—For logical ports of Failover type, initiates an immediate NIC failover; *value* must be "" (null string). If `Mode` is `Activefailback` and all ports are still alive, this will likely result in two immediate failover events back-to-back. To determine which port is active at any given moment, use the `Listlogical` command.
  - `Linkdowndelay`—Sets the delay to *value* (in milliseconds) for a port to consider its network link down. Defaults to 0.
  - `Linkmon`—Sets the Ethernet link status monitoring frequency to *value* (in milliseconds). Defaults to 500. A value of 0 disables link monitoring.
  - `Linkupdelay`—Sets the delay to *value* (in milliseconds) for a port to consider its network link up. Defaults to 0.
  - `Mode`—For logical ports of Failover type, sets the behavior mode; *value* must be one of the following:
    - `Activebackup`—Switches to any other available network port when the active port fails. This is the default mode.
    - `Activefailback`—Switches to another available network port when the active port fails, and switches back when that port is alive again.
- *logicalPort* is the logical interface name as created by `Netif Addlogical`.
- *value* is as described under *parameter*.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
23 Netif Setlogical Mode logical0 Activefailback
23 OK Completed

24 Netif Setlogical Linkdowndelay logical0 100
24 OK Completed
```

Unbind

Unbinds the IP address from the specified Ethernet port. When the command completes, the system is no longer accessible through the specified port.

Syntax

tag `Netif Unbind port`*num*

where `port`*num* identifies the Ethernet port, as labeled on the system back panel. Possible values are `port0`, `port1`, `port2` and so on.

This command cannot unbind the primary port. To change the IP address of port0, use the serial console menu. Also, you cannot unbind a port that has domain aliases associated with it; first use `Netif Deletealias` (see page 437) to delete all aliases.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
10 Netif Unbind port1
10 OK completed
```

Unbindlogical

Unbinds physical ports from a logical network interface. After all physical ports are unbound, the logical network interface can be removed with `Deletelogical`. You should unbind ports in the opposite of binding order; see `Bindlogical`.

When the last physical port is unbound from its logical interface, the physical port inherits network settings from the logical network interface. (The opposite occurs when the first physical port is assigned to an address: the logical interface inherits network settings from that physical port.)

Syntax

```
tag Netif Unbindlogical logicalPort physicalPort
```

where *logicalPort* is the name previously created by `Netif Addlogical`, and *physicalPort* is the port name, for example `port0`, `port1`, or `port2`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
25 Netif Unbindlogical logical0 port1
25 OK Completed
```

The Ntlm Command

The Ntlm command configures a Mirapoint system for NTLM (NT LAN manager) authentication of user logins. This is useful for adding email as a network service after single sign on to the Microsoft network.

Outlook users should enable the "SPA" setting (secure password authentication) to use NTLM as their authentication method if the capability is offered. Otherwise IMAP and POP authentication is done with Plaintext:Local or Plaintext:LDAP. To set SPA in Outlook 2003, select **Tools > E-Mail Accounts (Wizard) > Change** and click the checkbox **Log on using Secure Password Authentication**.

You can use LDAP to map NT user to LoginID, and NT domain to mail domain. See the user:Ntuserid query under LDAP Server Specifications on page 299.

NTLM is a Microsoft authentication protocol used with the SMB (CIFS) protocol. During protocol negotiation, its internal name is "nt lm 0.12" although the number 0.12 is unexplained. NTLM was followed by version two NTLMv2, at which time the original was renamed NTLMv1. There is no official protocol documentation, but it was reverse-engineered by the SAMBA team. The cryptographic calculations are documented by RFC 2433 for v1 and RFC 2759 for v2.

The protocol uses a challenge-response sequence requiring transmission of three messages between a client wishing to authenticate and the server requesting authentication. The client first sends to the server a type-1 message containing a set of flags specifying features supported or requested, such as encryption key sizes, or a request for mutual authentication. The server responds with a type-2 message containing a similar set of flags supported or required by the server, thus enabling an authentication agreement. More importantly, the server sends an 8-byte random challenge. The client combines the challenge from the type-2 message with the user's credentials and calculates the response. Calculation methods differ based on the NTLM authentication parameters negotiated previously, but in general they apply MD4/MD5 hashing algorithms and DES encryption to compute the response. The client sends this response to the server in a type-3 message.

Subcommands

Add

Adds a domain controller host and port to the list of domain controllers for a given NT domain. The port defaults to 139 but you can specify differently.

You can add up to 256 NT domains, with up to 256 domain controllers for each.

Syntax

tag Ntlm Add *ntdomain host:port*

where:

- *ntdomain* is the NT domain name.
- *host* is the (case-insensitive) name of a domain controller for that *ntdomain*. The *host* may be followed by an optional port number, specified after colon (*host*:*port*). If not specified, *port* defaults to 139.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Ntlm Add EXAMPLE dc.example.com
2 OK Completed
```

Count

Responds with the number of domain controllers in a given NT domain.

Syntax

tag Ntlm Count *ntdomain pattern*

where:

- *ntdomain* is the NT domain name.
- *pattern* may be * or null string (" ") to match all domain controllers.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
3 Ntlm Count EXAMPLE ""
* 3 1
3 OK Completed
```

Delete

Deletes the specified domain controller host (with optional port) from the list of domain controllers for a given NT domain.

Syntax

```
tag Ntlm Delete ntdomain host:port
```

where *host* is the domain controller you want to delete.

where:

- *ntdomain* is the NT domain name.
- *host* is the (case-insensitive) name of a domain controller for that *ntdomain*. The *host* may be followed by a port number, specified after colon (*host*:*port*). If not specified, *port* defaults to 139.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Ntlm Delete EXAMPLE dc.example.com
5 OK Completed
```

Export

Produces a list of *dchost:port* for every configured NT domain on the system, and the default *ntdomain* if set.

Syntax

```
tag Ntlm Export
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
7 Ntlm Export
* 7 domain=EXAMPLE
* 7 EXAMPLE dc.example.com
```

```
7 OK Completed
```

Getdefaultdomain

Returns the default NT domain established by `Setdefaultdomain`.

Syntax

tag `Ntlm Getdefaultdomain`

The default NT domain is used when an Outlook (or similar) client fails to provide a domain name in the NTLM type-1 authentication sequence.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
6 Ntlm Getdefaultdomain
* 6 EXAMPLE
6 OK Completed
```

Import

Inputs a list of *dchost:port* for NT domains provided in the import configuration. The import data must have the same format as given by `Ntlm Export`.

Syntax

tag `Ntlm Import` *URL value*

where:

- *URL* provides the location of a configuration file on the web. If *URL* is not given, this command expects the import data to be provided interactively (in the CLI) or as a counted literal string (in the protocol).
- *value* is a placeholder for possible future use.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Ntlm Import ftp://confighost.example.com/ntlm/domains ““
8 OK Completed
```

List

Responds with a list of all the domain controllers in a given NT domain.

Syntax

tag `Ntlm List` *ntdomain pattern start count*

where:

- *ntdomain* is the NT domain name.
- *pattern* may be either * or null string (“ “) to match all domain controllers.
- *start* is the first item to list.
- *count* is the number successive items to list.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
4 Ntlm List EXAMPLE "" "" ""
* 4 dc.example.com
4 OK Completed
```

Listdomain

Responds with a list of NT domains configured on this server for which at least one domain controller (host:port) was added.

Syntax

tag `Ntlm Listdomain` *pattern start count*

where:

- *pattern* is the name of an NT domain name, or * to match all domains.
- *start* is the first item to list.
- *count* is the number successive items to list.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
6 Ntlm List TEST "" ""
* 6 EXAMPLE
6 OK Completed
```

Setdefaultdomain

Establishes the default NT domain, used when an Outlook (or similar) client fails to provide a domain name in the NTLM type-1 authentication sequence.

Syntax

```
tag Ntlm Setdefaultdomain ntdomain
```

where *ntdomain* is the intended default NT domain name.

If only one NT domain is configured on the server, *ntdomain* may be omitted and that NT domain becomes the default. If more than one NT domain is configured on the server and *ntdomain* is omitted, an error results.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Ntlm Setdefaultdomain EXAMPLE
5 OK Completed
```

44

The Ntp Command

The `Ntp` command controls your Mirapoint system's interactions with Network Time Protocol (NTP) servers.

This command also supports the time protocol defined by RFC 868, used by the UNIX `rdate` command.

Date and Time Strings

The format of date and time strings reported by the `ntp` command is:

`month day-of-month hh``:`*`mm`*`:`*`ss timezone year`*

where:

- *`month`* is the name of the month, spelled out fully, such as "`September`".
- *`day-of-month`* is a number between `1` and `31`.
- *`hh`* is the two-digit hour using 24-hour clock, such as `03` or `17`.
- *`mm`* is the two-digit minute.
- *`ss`* is the two-digit second.
- *`timezone`* is a time zone abbreviation, such as `PDT`, for Pacific Daylight Time, or `EST`, for Eastern Standard Time, corresponding to the local time zone. This is an abbreviation of the full time zone name, described in Time Zones, below.
- *`year`* is the four-digit year, such as `1998`.

Time Zones

The Mirapoint system uses time zone names with this format:

`region``|`*`place`*

where:

- *`region`* is a large-scale geographic area, such as America or Pacific.
- *`place`* is a specific location within the region, such as a city, country, or island.

`America|Los_Angeles` is an example of a valid time zone name. Select a time zone in your local country and region, to stay current with possible government changes to Daylight Saving Time (Summer Time).

Subcommands

Add

Adds a host name to the list of NTP servers your Mirapoint system uses to regulate its clock.

Syntax

`tag Ntp Add ntphost`

where *ntphost* is the name of the NTP server you want to add.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Ntp Add ntp.example.com
2 OK Completed
```

Count

Responds with the number of NTP servers your Mirapoint system uses to regulate its clock.

Syntax

`tag Ntp Count pattern`

where *pattern* is currently ignored.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
2 Ntp Count ""
* 2 1
2 OK Completed
```

Delete

Deletes a host name from the list of NTP servers your Mirapoint system uses to regulate its clock.

Syntax

tag Ntp Delete *ntphost*

where *ntphost* is the name of the NTP server you want to delete.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Ntp Delete ntp.example.com
8 OK Completed
```

Get

Responds with the value of the specified parameter.

Syntax

tag Ntp Get *parameter*

where *parameter* is one of:

- `Date`—the current date and time (see Date and Time Strings on page 455).
- `Knownzones`—a list of all time zone names known to your Mirapoint system (see Time Zones on page 455).
- `Zone`—the current time zone.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
4 Ntp Get Zone
```

```
* 4 Etc|GMT
4 OK Completed
```

List

Responds with the list of NTP server host names that your Mirapoint system uses to regulate its clock.

Syntax

tag `Ntp List` *pattern start count*

where:

- *pattern* is currently ignored.
- *start* is the first position in the list of servers that you want to see. The empty string (`""`) implicitly means `0`.
- *count* is number of lines from the list of servers that you want to see. The empty string (`""`) implicitly means all servers. If *count* is greater than the total number of servers, `List` returns as many servers as possible.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
5 Ntp List "" "" ""
* 5 ntp.example.com
5 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

tag `Ntp Set` *parameter value*

where:

- *parameter* is one of:
  - `Date`—the current date and time. The value is of the form:

    yyyymmddhhmmss

    where:

- *yyyy* is the four digit year, such as `1999`.
- *mm* is the two-digit month, such as `06`.
- *dd* is the two-digit day of the month, such as `09`.
- *hh* is the two-digit hour using 24-hour clock, such as `02` or `16`.
- *mm* is the two-digit minute, such as `08`.
- *ss* is the two-digit second, such as `05`.

❖ `Zone`—name of the prevailing time zone for this system. Zone name is case-sensitive and must appear in the list returned by `Ntp Get Knownzones`, with capitals as listed. Always use modern timezone names, in the form Continent/City (sometimes Continent/Region/City) instead of deprecated names such as US/Pacific. Always select a time zone in your own country, and region if applicable, to stay current with possible government changes to Daylight Saving Time (Summer Time).

◆ *value* is the value you want to assign to *parameter*.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Ntp Set Zone "America|Los_Angeles"
6 OK Completed
```

Synchronize

Used during first-time setup or in extreme circumstances to synchronize your system's clock with the specified NTP server. Under normal circumstances, the NTP software synchronizes itself.

If the specified host is not running an NTP server, `Ntp Synchronize` attempts to use the time protocol (see RFC 868), used by the UNIX `rdate` command.

This command does nothing if your system clock is already synchronized with the specified NTP or `rdate` server.

Syntax

```
tag Ntp Synchronize ntphost
```

where *ntphost* is the host name or IP address of the NTP server with which you want to synchronize your Mirapoint system.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
7 Ntp Synchronize ntp.example.com
7 OK Completed
```

45 The Patternlist Command

The `Patternlist` command should eventually replace the `Wordlist` command. See filter Rules on page 240 for information about using patternlists with the `Filter` command. For release 3.10, features of this command are described under `Patternlist` Import on page 463.

Subcommands

Count

Returns the number of patternlists on the system.

Syntax

```
tag Patternlist Count pattern
```

where *pattern* must be "" or * to match all patternlists.

Privilege Levels

- ◆ Administrator
- ◆ Helpdesk administrator
- ◆ Domain administrator
- ◆ Backup operator

Domain Sensitivity

Patternlists are specific to either a delegated domain or the top-level domain.

Example

```
5 Patternlist Count ""
* 5 2
5 OK Completed
```

Delete

Erases the specified patternlist from the system.

Syntax

tag `Patternlist Delete` *listname*

where *listname* is a patternlist as specified by `Patternlist Import`.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

Patternlists are specific to either a delegated domain or the top-level domain.

Example

```
7 Patternlist Delete company
7 OK Completed
```

Export

Outputs the specified patternlist as a string literal.

Syntax

tag `Patternlist Export` *listname*

where *listname* is a patternlist as specified by `Patternlist Import`.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Patternlists are specific to either a delegated domain or the top-level domain.

Example

```
4 Patternlist Export company
{19}
Beispiel
Ejemplo

4 OK Completed
```

Import

Creates a new patternlist given a string literal or URL. If a patternlist of that name already exists, it is replaced. Patternlist has the following features (as of 3.10):

1. The filter predicate "`matches-patternlist`" supports email addresses only.
2. All characters match themselves, except reserved characters colon, slash, and backslash (`:/\`) which must be backslash-escaped, except as noted below.
3. In both pattern and input: parenthetical comments, unquoted whitespace, duplicate quotes, and angle brackets (<>) are removed. Multiple spaces are compressed to one. Upper case is mapped to lower case. All these modifications are according to SMTP convention.
4. The final @ separates LHS (left-hand-side) from RHS (right-hand-side).
5. Subfolders (user+folder@example.com) are ignored when matching LHS, unless a subfolder is specified in the pattern, in which case both user and folder name must match.

You are limited to 10 patternlists per domain, and each list must be under 1 MB. To determine the size of a patternlist, you can `Export` it.

Syntax

`tag` `Patternlist Import` *`listname patternlist`*

where:

- *`listname`* is the (case-sensitive) name of a patternlist. Names beginning with the prefixes "`System:`" and "`SR:`" are reserved for use by Mirapoint. Any *`listname`* beginning with "(" is interpreted as a set of optional arguments for referring to lists and domains outside the current domain, including:
  - `(domain=`*`domainname`*`)`—where *`domainname`* is a delegated domain.
  - `(name=`*`listname`*`)`—where *`listname`* designates the patternlist name.
- *`patternlist`* is a string literal or a URL. If *`patternlist`* is a URL, it must begin with `ftp://` or `http://`, it must not contain newline, and it must be followed by nothing else.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

Patternlists are specific to either a delegated domain or the top-level domain.

Example

```
2 Patternlist Import obscene http://not.fcc.us.gov/bannedpatternlist
```

```
2 OK Completed
```

```
3 Patternlist Import confidential {12+}
CMOS
DRAM
```

```
3 OK Completed
```

List

Returns names of patternlists on the system.

Syntax

tag Patternlist List *pattern start count*

where:

- *pattern* must be "" or * to match all patternlists.
- *start* is the first patternlist to return.
- *count* is number of patternlists to return.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Patternlists are specific to either a delegated domain or the top-level domain.

Example

```
6 Patternlist List "" "" ""
* 6 company
* 6 obscene
6 OK Completed
```

46

The Pop Command

The `Pop` command configures behavior of the POP mail reading service.

POP service requires a license, usually based on the number of users supported.

If messages were deleted but not yet expunged by IMAP or Webmail, POP clients download them anyway.

POP Service Modes

The POP service can operate in either of two modes: **normal mode** or **proxy mode**, as described in the following sections. Use `Pop Set Mode` to set the POP service mode.

Normal Mode

In normal mode, the POP service accepts POP connections providing access only to mailboxes on the local host.

Proxy Mode

In this mode, POP service acts as a POP proxy. When a user connects to a POP proxy, the proxy uses an LDAP database to determine the mail host where the user's email is stored. It then logs into the POP service on that mail host and passes POP commands and responses between the user's mail client and POP service on that mail host for the duration of the user's POP connection.

For the POP proxy to work correctly, you must run the `Ldap Setquery` command to define the `user:Mailhost` and `user:Loginid` query specifications correctly for your LDAP database(see LDAP Query Specifications on page 300).

You can use the POP service in proxy mode only if an Mail Routing license is installed on the system. Contact your Mirapoint sales representative for details.

Subcommands

Get

Responds with the value of the specified parameter.

Syntax

tag `Pop Get` *parameter*

where *parameter* is one of:

- `Auth`—The authentication methods available for POP login (see `Pop Set`).
- `Mode`—The mode in which POP service is currently operating (see `Pop Set`).
- `Minpoll`—The minimum interval in minutes that must elapse between POP connections by a particular user. Users may reconnect to POP service within this period, but may not access new messages until the time elapses. This parameter helps prevent excessive polling by POP clients, which can degrade performance. For empty mailboxes, the `Minpoll` restriction is lifted, it being unneeded.
- `Security`—The security schemes allowed for POP connections (see `Pop Set`).
- `Timeout`—The POP service idle timeout in minutes. The POP service closes any connection that remains idle for this period.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

The `Minpoll` parameter is specific to a delegated domain. The others are not.

Example

```
3 Pop Get Minpoll
* 3 5
3 OK

4 Pop Get Security
* 4 "ssl cleartext"
4 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

tag `Pop Set` *parameter value*

where:

- *parameter* is one of:
  - `Auth`—Controls what authentication mechanisms are available in POP. The *value* would be set to one or more of the following:
    - `APOP`—Authenticated POP, as described in section 7 of RFC 1725. When enabled a **nonce** string is appended to the initial server greeting,

which string the client hashes with the user's password and then uses with the APOP command to authenticate. The administrator controls who may use APOP with the `Auth` command; see `Auth Set`.

  - `Kerberos_v4`—Kerberos 4 authentication.
  - `Kerberos_v5`—Kerberos 5 authentication.
  - `Plaintext`—Login authentication with local or LDAP plain text.

- `Mode`—Proxy mode in which the POP service is currently operating. The value of this parameter must be one of:
  - `Ldapproxy`—the POP service acts as a POP proxy (see POP Service Modes on page 465).
  - `Normal`—the POP service accepts normal POP connections, providing access to mailboxes on the local host.
- `Minpoll`—The minimum interval in minutes that must elapse between POP connections by a particular user. Users may reconnect to POP service during this period, but may not access any new messages until time elapses. Each attempt made to access POP during this interval causes the system to log a `POP.USER.MINPOLL` event. With an empty mailbox however, a user can poll for new messages without generating a `POP.USER.MINPOLL` event. An interval of 0 (zero, the default) disables this feature. The polling interval for a delegated domain is inherited from the primary domain.
- `Security`—The security schemes allowed for POP connections. The value of this parameter must be a quoted, space-separated list of any of the following data-privacy schemes:
  - `Cleartext`—No encryption of data for incoming POP connections.
  - `Cleartextout`—No data encryption for outgoing POP connections. Use this setting together with `Ssl` on message proxies to specify encrypted incoming connections from mail clients and cleartext proxy connections to message servers.
  - `Secureauthonly`—When set, prohibits plaintext passwords on a clear-text connection. `Secureauthonly` requires that `Cleartext` be set. If `Secureauthonly` is set alone, then `Cleartext` is set automatically. If `Secureauthonly` is set with other values but not including `Cleartext`, an error message results.
  - `Ssl`—Secure Sockets Layer (SSL, versions 2 and 3) and Transport Layer Security (TLS, version 1) encryption of data for incoming POP connections. Use this setting on both message proxies and servers.
  - `Sslout`—SSL for outbound POP connections. Use this setting on a message proxy to encrypt POP proxy connections to message servers. When `Cleartextout` is also set, stays with what the client used when connecting to the back-end (cleartext stays cleartext, SSL stays SSL).
  - `Starttls`—If set, TLS is allowed and supported on incoming POP connections. Requires SSL license, weak or strong. The server advertises STARTTLS capability to notify clients that it is supported.
  - `Starttlsout`—If set, the POP proxy tries to use TLS when sending messages to remote mailhosts. Requires SSL license, weak or strong.

`Starttls` applies to both proxy and server, while `Starttlsout` applies only to the proxy.

   - ❖ `Timeout`—The POP service idle timeout in minutes. The POP service closes any connection that remains idle for this period. The POP3 standard (RFC 1939) requires a setting of at least ten minutes.

- *`value`* is the value you want to assign to *`parameter`*.

Privilege Levels

Administrator

Domain Sensitivity

The `Minpoll` parameter is specific to a delegated domain. It is inherited at creation time, and can be changed independently. Other parameters are not domain specific.

Example

```
5 Pop Set Minpoll 5
5 OK

6 Pop Set Security "Cleartext Ssl"
6 OK Completed

7 Pop Set Auth "APOP Plaintext"
7 OK Completed

8 Pop Set Security "Starttls Starttlsout Secureauthonly"
8 OK Completed
```

47

The Quota Command

The `Quota` command manages quotas on mailboxes and overquota messages.

Subcommands

Get

Responds with the value of the specified parameter.

Syntax

tag `Quota Get` *parameter mailbox*

where:

- *parameter* must currently be `Storage`, referring to the quota of the specified mailbox. If there is a quota on the mailbox, the response line shows the number of kilobytes used by the mailbox followed by the quota in kilobytes. If there is no quota on the mailbox, the command returns a `NO` response.
- *mailbox* is the name of the mailbox for which you want quota information.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- User (can get only his or her own quota)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

In this example, `user.demo` is using 3 KB of space with a quota of 5120 KB.

```
3 Quota Get Storage user.demo
```

```
* 3 user.demo 3 5120
3 OK Completed
```

Getpolicy

Retrieves the value of the specified quota policy parameter.

Syntax

tag Quota Getpolicy *parameter*

where *parameter* is one of the following:

- Overquota—Action to take when users' mailbox exceeds their storage quota.
- Overquotamessage—The message sent to users who exceed mailbox quota.
- Sendoverquotamessage—Whether to send overquota messages in a domain.
- Defaultsendoverquotamessage—System default for overquota messages.

Privilege Levels

- Administrator
- Helpdesk administrator (except Overquota parameter)
- Domain administrator (Overquotamessage and Sendoverquotamessage only)

Domain Sensitivity

Overquotamessage and Sendoverquotamessage work in delegated domains. In the primary domain they affect the system-wide default message; in delegated domains they change the domain-specific message. Other parameters are allowed only when no delegated domain is current.

Example

```
6 Quota Getpolicy Overquota
* 6 QUEUE
6 OK Completed

7 Quota Getpolicy Overquotamessage
* 7 {112}
Subject: over quota

Please delete some messages to free up space
so you can continue receiving new email.

7 OK Completed
```

Set

Sets the local disk quota for mailbox storage.

Quota can be managed from LDAP using the attribute miMailQuota or mailQuota. When a user account and mailbox is provisioned from LDAP, local disk quota is set using the assigned LDAP value. When the COS `Quota` option is enabled, the system rechecks quota from LDAP if an overquota condition occurs.

Syntax

tag `Quota Set` *parameter mailbox value*

where:

- *parameter* must currently be `Storage`, referring to the quota of the specified mailbox in kilobytes. The special value `-1` tells `Quota Set` to remove the existing quota from the mailbox.
- *mailbox* is the name of the mailbox on which you want to set a quota.
- *value* is the value you want to assign to *parameter*.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
2 Quota Set Storage user.demo 5120
2 OK Completed
```

Setpolicy

Sets the value of the specified quota policy parameter.

Syntax

tag `Quota Setpolicy` *parameter value*

where *parameter* combines with *value* as follows:

- `Overquota`—Controls email policy when a user's mailbox usage exceeds the storage quota; *value* may be one of:
  - `Queue`—Queue messages addressed to an over-quota user mailbox for later delivery. Redelivery is attempted until the user has deleted enough messages to allow delivery. This is the default overquota policy.
  - `Reject`—Reject messages addressed to over-quota user mailboxes.
  - `Allowinbox`—Allow messages to be added to a user inbox even if the inbox is over quota. If a message is sent to an overquota subfolder, the message gets delivered to the inbox instead. The size of the inbox, although unlimited, counts against the user's quota. Overquota warning messages are

generated for the inbox, but overquota warning messages for subfolders are placed in the subfolder. This policy affects only the `user.*` hierarchy.

- `Overquotamessage`—The message sent to over-quota users; *`value`* must be a string literal specifying the message, or the word `Default` to restore the original overquota message. In the primary domain, this changes the system-wide message, whereas in delegated domains, it changes the domain-specific message.
- `Sendoverquotamessage`—Whether to send overquota messages in a domain; *`value`* may be one of:
  - `Yes`—Send overquota messages (the default).
  - `No`—Do not send overquota messages.
  - `Default`—Inherit overquota message behavior from system (see below).
- `Defaultsendoverquotamessage`—System default for overquota messages; *`value`* may be one of:
  - `Yes`—For domains without a specific setting, or those set to `Default`, send overquota messages (system default).
  - `No`—For domains without a specific setting, or domains set to `Default`, do not send overquota messages.

Privilege Levels

- Administrator
- Helpdesk administrator (except `Overquota` parameter)
- Domain administrator (`Overquotamessage` and `Sendoverquotamessage` only)

Domain Sensitivity

`Overquotamessage` and `Sendoverquotamessage` affect, and can be set from, delegated domains. Other parameters are allowed only in the primary domain.

Example

```
4 Quota Setpolicy Overquota Queue
4 OK Completed

5 Quota Setpolicy Overquotamessage {112+}
Subject: over quota

Please delete some messages to free up space
so you can continue receiving new email.

5 OK Completed
```

The Radius Command

The `Radius` command lets you configure the Mirapoint system to use Radius authentication for login verification of users.

A Radius server must exist on the network. To set up Radius authentication on the Mirapoint server, you add the Radius server to the system list, and set the secret key using the `Radius Set Secret` command. On the Radius server, grant access to the Mirapoint system for authentication service. Only one Radius server is allowed.

Subcommands

Add

Adds the specified host as the Mirapoint system's (only) Radius server.

Syntax

```
tag Radius Add host
```

where *`host`* is the fully qualified domain name, or dotted IP address, of a Radius server. The *`host`* may include a port number, specified after colon (*`host`*`:`*`port`*).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Radius Add radius.example.com
2 OK Completed
```

Count

Responds with the number of Radius servers on the Mirapoint system (always < 2).

Syntax

tag Radius Count *pattern*

where *pattern* is currently ignored.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
3 Radius Count ""
* 3 1
3 OK Completed
```

Delete

Deletes the specified host as the Mirapoint system's Radius server.

None

Syntax

tag Radius Delete *host*

where *host* is the Radius server you want to delete.

Privilege Levels

Administrator

Domain Sensitivity

Example

```
5 Radius Delete radius.example.com
5 OK Completed
```

Get

Responds with the value of the specified parameter.

Syntax

tag Radius Get *parameter*

where *parameter* is one of:

- `Secret`—the secret key shared between the Radius server and your Mirapoint system, shown encoded (see `Radius Set`).
- `Sendusernameastyped`—authenticate with user name as typed, rather than with user name normalized by LDAP (see `Radius Set`).
- `Timeout`—the time in seconds that the system waits for a Radius authentication request to complete before aborting the attempt. This value is returned as a literal string (see Literal Strings in Responses on page 49).

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
6 Radius Get Timeout
* 6 30
6 OK Completed
```

List

Responds with the current list of Radius servers that a Mirapoint system can query.

Syntax

tag `Radius List` *pattern start count*

where:

- *pattern* may be either `*` or " " to list everything.
- *start* is the number of the first Radius server you want to see. The empty string (`""`) implicitly means `0`.
- *count* is the number of Radius servers you want to see. The empty string (`""`) implicitly means all servers. If *count* is greater than the total number of Radius servers, `list` returns as many servers as possible.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
4 Radius List "" "" ""
* 4 radius.example.com
4 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

tag Radius Set *parameter value*

where:

- *parameter* is one of:
  - `Secret`—the secret key shared between the Radius server and the Mirapoint system. You must set the same secret key on the Mirapoint system that you set on your Radius server. You must also grant access for the Mirapoint system on your Radius server. May be set in "`(encodedpass=)`" form.
  - `Sendusernameastyped`—when `On`, authenticate with the login name that the user typed, not with the name returned by an LDAP query for login ID. This does not change the `user:LoginID` attribute, it only affects what gets sent to the Radius server. The default is `Off`.
  - `Timeout`—the length in seconds that the system waits for a Radius authentication request to complete before aborting the attempt. The default timeout value is 5 seconds.
- *value* is the value to which you want to set *parameter*.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Radius Set Timeout 30
5 OK Completed
```

49

The Relay Command

The `Relay` command lets you specify IP networks or DNS domains for which the SMTP service accepts messages for relay to remote hosts. A message is relayed if sent from (or to) a domain in the relay list. Controlling message relaying can help prevent your system from being used to spread somebody else's junk mail.

Before relaying is allowed, DNS lookup on the sending host must determine that PTR and "A" records both exist, and match.

It is not necessary to add relay addresses for mail domains or delegated domains. The `Relay` command has no effect on local messages sent to local mailboxes.

Networks

To specify a network from which to accept mail for relay, use a partial IP address. For example, if you specify `10.99`, the SMTP service will accept relays from (or to) `10.99.0.1` and `10.99.3.1`, but not from (or to) `10.20.0.1`. Numeric IP addresses are consulted only for outbound relaying, whereas domain names are consulted for both inbound and outbound relaying.

Subcommands

Add

Specifies an IP network or DNS domain name from (or to) which the system accepts mail for relay. Unless a relay address is explicitly added, the Mirapoint system will not relay messages from that network or domain.

You can add relays until the list gets about 9.6 MB in size. System performance is hardly affected by large relay lists.

Syntax

```
tag Relay Add relay
```

where *relay* is the IP network or DNS domain name from (or to) which you want to accept mail for relay.

For Signature Edition Rapid Antispam, *relay* must be a numeric IP address.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Relay Add example.com
2 OK Completed
```

Count

Responds with the number of IP networks and DNS domains from (or to) which the system currently accepts mail for relay.

Syntax

tag `Relay Count` *pattern*

where *pattern* is currently ignored.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
3 Relay Count ""
* 3 1
3 OK Completed
```

Delete

Disallows relaying for mail from (or to) the specified IP network or DNS domain . Only previously added relay addresses may be deleted.

Syntax

tag `Relay Delete` *relay*

where *relay* is the IP network or DNS domain from (or to) which you no longer want to accept mail for relay.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Relay Delete example.com
5 OK Completed
```

List

Responds with the list of IP networks and DNS domains from (or to) which the system currently accepts mail for relay.

Syntax

`tag Relay List pattern start count`

where:

- `pattern` is currently ignored.
- `start` is the first position in the list of IP networks and DNS domains that you want to see. The empty string (`""`) implicitly means `0`.
- `count` is number of IP networks and DNS domains that you want to see. The empty string (`""`) implicitly means all networks and domains. If `count` is greater than the total number of addresses, `list` returns as many as possible.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
4 Relay List "" "" ""
* 4 example.com
4 OK Completed
```

50

The Replicate Command

Remote Server Replication (RSR) maintains pairs of Mirapoint systems for offsite disaster recovery with near-time (4 to 24 hour) data replication. RSR works by synchronizing data from a Master server to a Replica server. If the Master server becomes unavailable, the Replica can replace it. For this to occur, an administrator must switch DNS records so the Replica serves as the new Master.

The RSR facility requires a license on both the Master server and the Replica server.

Synchronization copies data and various configuration settings from the Master to the Replica, but does not immediately replace the Replica system's data. Instead, data gets transferred to an inactive location, where copies of *Deltas* (changes made on the Master since last synchronization) go into a staging area. Deltas are applied to the inactive store only after the copy from Master to Replica is complete.

Both the Master and the Replica can be patched and upgraded, in tandem. After reboot of either system, replication continues normally. Loss of the Master server at any time (after initial synchronization) does not impair the Replica's ability to assume the role of Master soon after any failure. The replication facility employs the `Schedule` command (see below).

Mirapoint licenses are automatically transferred from the Master to Replica server when the Replica assumes the Master's role. All transferred licenses are provisional. If you want the Replica to take over permanently, or if you replace the Master with new hardware, you must apply new licenses. Please contact Mirapoint before you run `License Apply` or `License Fetch`, so as to avoid harm to pre-existing licenses.

Setting up Replication

Replication uses port 873 for synchronization, so if you are replicating through one or more firewalls, be sure to open up this port in both directions.

To set up replication between Master and Replica, use the following commands:

```
Replica> Replicate From masterHost

Master> Replicate To replicaHost
Master> Replicate Sync
Master> Replicate Status
Master> Schedule Add scheduleSync daily "Replicate Sync"

Replica> Schedule Add scheduleCheck daily "Replicate Checkswitch"

Replica> Replicate Switchover
Replica> Replicate Status
```

In the preceding commands, the replication facility:

- Sets up replication between *masterHost* and *replicaHost*.
- Performs the initial `Sync` operation, and displays status of the replication.
- Requests daily follow-on synchronizations with the `Schedule` command.
- Schedules daily verifications of the Replica's readiness using `Checkswitch`.
- If needed, switches Replica to assume duties of the Master, and displays status.

The storage required on the Replica system is sum of storage used on the Master, plus the largest Delta expected in any synchronization. As explained above, each Delta (changes since the last `Sync`) is completely transferred to the Replica, then atomically applied to update the system image.

Implications for Backup

Image backup and the `Replicate Sync` command cannot run simultaneously. If either a backup or a `Sync` is active when the other attempts to run, the second one fails. If you want to terminate any `Sync` operation at a scheduled time so that a scheduled backup can proceed instead, even if the `Sync` has not completed, use the `Schedule Add` command to schedule a `Replicate Cancel` operation on the Master before the image backup is due to start. This is a safe option, because when no `Sync` operation is active, `Replicate Cancel` has no impact.

You need not back up replication settings and management data on the Master server. In general terms, the Replica acts as a "warm backup" for the Master host. Moreover, restoration of a Master from a time before the last successful `Sync` would make it impossible to construct a coherent Replica without starting over.

You should back up replication settings and management data on the Replica host, because the Master can initiate a `Sync` at any point in time. If the Replica is restored to a time before the last few `Sync` operations, the Master simply synchronizes more data to the Replica host.

Questions and Answers

How resource hungry is the replication process?

That is difficult to answer. It depends on amount of data involved. Full replication is disk I/O intensive. Partial replication is CPU intensive. That said, network bandwidth is usually the limiting factor.

Is there a way we can run backups while replication is running or are the two mutually exclusive? How do we run backups?

Replication only affects backups when Sync is running. Both NDMP backup and replication use the same resource inside MOS, and only one process can use that resource at a time. If replication owns the resource, then the backup cannot start, and vice versa. The way we have typically addressed this is to work on the schedule for both so that they do not conflict. If we run the replication Sync first, then we use `Replicate Cancel` in a schedule on the master to clear the way for the backup. The correct order depends on priorities you assign to the two activities: is backup more or less important than replication?

Would Junk Mail be replicated or is it just “legitimate mail” replicated?

All data on the mail store gets replicated, including junk mail folders. Changing that is not possible because replication works at a lower level than mail folders or IMAP flags. If your site deploys a RazorGate Junk Mail Manager (JMM), you can choose not to replicate JMM.

How long does replication take? How often could we run replication? Is there any way to predict?

There are so many factors involved here that it’s really difficult to say with any degree of accuracy. The most obvious factors are size of the delta, the bandwidth between the master and the replica, and the Round Trip Time (RTT) of the link between the master and the replica. RTT doesn’t directly impact the replication time unless it gets too high, in which case replication throughput might start to suffer. The size of the delta is a little more difficult to measure. It can be estimated from the email volume that you take in each day, but replication copies more than email, so other services such as Calendar and AddressBook obviously impact size of the delta. Internal MOS structures must be copied also; this is not directly measurable.

We have one customer that is running a relatively lightly loaded system with 100 Mb/sec connectivity between the two systems who runs a `Sync` once an hour, and the `Sync` takes between 4 and 10 minutes. Another customer only has 1.5 Mb/sec (or slightly less) between systems and can run the `Sync` only once a day in the middle of the night, when it takes 3.5 hours. In this case, bandwidth is a massive bottleneck. They tested `Sync` on gigabit ethernet and it took less than 30 minutes. The best way to get ballpark figures is to actually do some replication tests under normal daily loading.

What would happen in the event of a disaster part way through a replication cycle? Does the system revert to the previous state, or do some of the changes apply?

The replication system has two holding areas that are not normally visible to MOS. The first is called the “inactive store” and contains a point-in-time snapshot of the mail store on the master at the start of the last completed `Sync`. The second contains the delta that is currently being transferred. The inactive store is not touched until all data from the delta has been copied to the replica. Then, and only then, does a process apply the delta to the inactive store. When the delta application starts, there is no more network traffic needed between the two systems and the master can fail but the delta still gets applied. Even if the replica reboots in the middle of delta application, the delta application resumes after reboot, so you end up with a consistent inactive store. This is critical as we cannot have incomplete or partial deltas being applied to the inactive store. Internal MOS data structures must accurately represent the state of mailboxes, or unexpected behavior could result.

Subcommands

Cancel

When run on the Master system, cancels the current `Sync` operation on the Master and aborts any actions on the Master related to that `Sync` operation. Cancelling a synchronization is typically done on the Master system. If no `Sync` is in process, `Cancel` returns with no action, making it useful for imposing limits on the amount of time allowed for a `Sync` operation.

You would cancel a synchronization on the Replica system only in rare cases such as performing maintenance when you cannot wait for the operation to complete. When run on the Replica system, `Cancel` stops the synchronization on the Replica if it is not in the midst of applying a Delta that is completely transmitted from the Master. This might or might not cause the Master to abort the transfer, depending on the transfer retry count. To restart synchronization, use the `Replicate Restart` command on the Replica.

Syntax

```
tag Replicate Cancel
```

Privilege Levels

Administrator

Domain Sensitivity

None.

Examples

```
2 Replicate Cancel
* 2 Sync Operation Terminated
2 OK Completed
```

```
3 Replicate Cancel
3 OK Completed
```

Checkswitch

Performs tests to determine if the Replica would be able to replace the Master in the event of a failure. Typically you run `Replicate Checkswitch` on a Replica system.

Syntax

```
tag Replicate Checkswitch
```

Privilege Levels

Administrator

Domain Sensitivity

None

Examples

```
4 Replicate Checkswitch
4 OK Completed
```

```
5 Replicate Checkswitch
5 NO Operation In Process
```

```
6 Replicate Checkswitch
6 NOTE:Replica NOT Viable, No Syncs Completed
```

```
7 Replicate Checkswitch
* 7 MOS Version Mismatch: Store '3.7.5GA' Replica '3.7.4GA'
7 NO Replica Not Ready
```

```
8 Replicate Checkswitch
* 8 Patchlist Mismatch
* 8 Replica Missing: "E3_Hash_2"
* 8 Replica Missing: "E3_FooBar_7"
* 8 Replica Missing: "E3_Widgit_9"
* 8 Replica Added: "debug-1"
* 8 Replica Added: "E3_Widgit_8"
8 NO Replica Not Ready
```

Clear

Cancels any replication processes underway, except for a Delta being applied on the Replica, and erases all replication information (data, logs, etc.) on the system where you run this command. `Clear` may be run on either Master or Replica, but if you intend to discontinue replication entirely, run it on both systems.

After you run `Clear` on the Replica, the Replica is no longer a viable replacement for the Master.

Running `Clear` on the Master does not prevent the Master from functioning as the Master again, nor does it prevent the Replica from replacing the Master with data current as of the timestamp at the last successful `Sync` command. However, no further `Sync` commands can be run.

To reestablish synchronization operations between the two appliances, the Replica must be cleared and the replication process restarted using the `Replicate From` command as originally. You can specify a different Replica, preserving the viability of the original Replica based on the last `Sync` it received.

Syntax

```
tag Replicate Clear
```

Privilege Levels

Administrator

Domain Sensitivity

None

Examples

```
8 Replicate Clear
8 OK Completed
```

Finalize

Shuts down services on the Master and cleanly synchronizes changes made since the last transfer of data to the Replica. You run this command on the Master server. After you run this command, the Master is unavailable to users until rebooted. `Finalize` is not intended for use during normal, ongoing replication operations.

`Finalize` is an excellent way to migrate all users from an old server to a new server, respectively designated Master and Replica. It is also a way to restore operations from the Replica (acting as primary) back to the original Master after a disaster affecting the Master data center has passed, once you've swapped system roles.

This command is similar to the `Replicate Sync` command, but it also shuts down services on the Master before synchronizing and transferring data. This ensures that the Replica becomes a complete copy of the Master. See `Replicate Sync` for more information about all the actions performed by this command.

Syntax

```
tag Replicate Finalize
```

Privilege Levels

Administrator

Domain Sensitivity

None

Examples

```
9 Replicate Finalize
* 9 WARNING: Services Stopped
* 9 Sync Started
9 OK Completed
```

From

You run the `Replicate From` command once on the Replica to prepare the system for receiving data from the Master. `Replicate From` ensures that no replication has been previously configured.

Syntax

```
tag Replicate From masterHost
```

where *masterHost* is the hostname or IP address of the Master system.

Privilege Levels

Administrator

Domain Sensitivity

None

Examples

```
4 Replicate From master.example.com
4 OK Completed
```

```
5 Replicate From 192.168.1.2
5 OK Completed
```

```
6 Replicate From master2.example.com
6 NO REPLICATE.COMM.FAILURE Master:master2.example.com Replica:rep.example.com
```

```
7 Replicate From master.example.com
7 No Replication Already Initiated
```

Get

Responds with the value of the specified parameter.

Syntax

tag `Replicate Get` *parameter*

where *parameter* is one of those specified under `Replicate Set`.

Privilege Levels

Administrator

Domain Sensitivity

None

Examples

```
2 Replicate Get RetryCount
* 2 20
2 OK Completed
```

```
3 Replicate Get RetryDelay
* 3 300
3 OK Completed
```

```
4 Replicate Get BandwidthLimit
* 4 9000
4 OK Completed
```

Restart

Restarts a failed replication server. This command is not necessary during normal operations. You run the `Restart` command on the Replica server. This command is performed automatically when the Replica reboots.

Syntax

tag `Replicate Restart`

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Replicate Restart
2 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

tag `Replicate Set` *parameter value*

where:

- *parameter* is one of:
  - `RetryCount`—The number of attempts made to copy a Delta to the Replica system before aborting the `Sync` operation. *Value* can range from 0–100. A value of zero (0) means retry forever. The default is zero.
  - `RetryDelay`—The number of seconds to wait after a copy attempt failure before retrying the copy. *Value* can range from 10–600. The default is 10.
  - `BandwidthLimit`—Limit on the number of kilobytes per second of bandwidth used for data transfer between the Master and Replica servers. *Value* can range from 0–1,000,000 KBps. A value of zero (0) means no bandwidth limit. The default is zero.
- *value* is the value you want to assign to *parameter*.

Privilege Levels

Administrator

Domain Sensitivity

None

Examples

```
2 Replicate Set RetryCount 20
2 OK Completed
```

```
3 Replicate Set RetryCount 500
3 NO Limit Exceeded

4 Replicate Set RetryDelay 300
4 OK Completed

5 Replicate Set BandwidthLimit 9000
5 OK Completed
```

Status

Displays a brief summary of the current replication status. You can run the `Status` command on either the Master or the Replica. They return similar information. Status information appears under these headings:

- **Host Role**—Whether the host is the Master or Replica. Values can be `Master`, `Replica`, or `Undecided`.
- **Sync**—Whether a synchronization is currently active. Values can be `Active` or `Inactive`.
- **Delta Application**—Whether copied Deltas are currently being applied on the Replica. Values can be `Active` or `Inactive`.
- **Replica Current As Of**—Timestamp of the last successfully-completed `Sync` command and the number of hours since that time. Or possibly a note that no synchronization has ever occurred on the Replica.
- **Last Sync**—The status of the last `Sync` command. Values can be `Running`, `Completed`, `Failed`, `Cancelled`, or `None`.
- **Last Sync Size**—The size and total elapsed time of the last `Sync` transfer. Size is specified in megabytes; elapsed time is specified in hours and minutes.
- **Syncs**—Provides statistics on the number of `Sync` commands that have been `Attempted`, `Completed`, or `Failed/Cancelled`.
- **Replication Server**—Reports whether the Replica system is listening for connections. Values displayed on the Master can be `Running` or `Unreachable`; values displayed on the Replica can be `Running` or `Down`.
- **Master Store Size**—Provides storage statistics for the Master system, including total capacity (megabytes), storage used (megabytes), total number of inodes (files and directories), and the number of inodes in use.
- **Replica Store Size**—Provides storage statistics for the Replica system, including total capacity (megabytes), storage used (megabytes), total number of inodes (files and directories), and the number of inodes in use.

Syntax

```
tag Replicate Status
```

Privilege Levels

Administrator

Domain Sensitivity

None

Examples

The first example shows all the possible status results. The second example shows results while a Sync is in progress.

```
6 Replicate Status
* 6 Host Role is: [Master|Replica|Undecided]
* 6 Sync is: [Active|Inactive]
* 6 Delta Application is: [Active|Inactive]
* 6 Replica Current As Of: <ISO DATE-TIMESTAMP>; <HH:MM> Ago
* 6 NOTE: Replica NOT Viable, No Syncs Completed
* 6 Last Sync: [Applying|Running|Completed|Failed|Cancelled|None]
* 6 Last Completed Sync Size: <int> MB Elapsed Time: <HH:MM>
* 6 Syncs: Attempted: <int>, Completed: <int>, Failed/Cancelled: <int>
* 6 Replication Server: [Running|Down|Unreachable]
* 6 Master Store Size: <int> MB, Used: <int> MB, Inodes: <int>, Used: <int>
* 6 Replica Store Size: <int> MB, Used: <int> MB, Inodes: <int>, Used: <int>
6 OK Completed
```

```
7 Replicate Status
* 7 Host Role is: Master
* 7 Sync is: Active
* 7 Delta Application is: Active
* 7 Replica Current As Of: Thu Jan 4 22:39:11 GMT 2007; 01:13 Ago
* 7 Last Sync: Applying
* 7 Last Completed Sync Size: 4838 MB Elapsed Time: 00:45
* 7 Syncs: Attempted: 2, Completed: 1, Failed/Cancelled: 0
* 7 Replication Server: Running
* 7 Master Store Size: 188527 MB, Used: 4871 MB, Inodes: 24208126, Used: 174027
* 7 Replica Store Size: 66375 MB, Used: 5457 MB, Inodes: 8542398, Used: 178834
7 OK Completed
```

Switchover

Switchover causes the Replica to assume the role of the Master by:

- Checking that there is no ongoing application of Deltas. If a Delta application is in progress, the Switchover command fails and must be rerun by the administrator once the Delta application completes.
- Confirming that the software configuration (MOS and patches) on the Replica matched those on the Master at the time of the last completed Sync operation.
- Shutting down the Replica server to avoid initiation of new Sync operations.
- Shutting down the services on the active store.
- Archiving the active store.
- Replacing the active store with contents of the inactive store.
- Applying all licenses from the Master (as temporary licenses on the Replica).
- Rebooting the Replica to apply the saved configuration as replicated from the Master.

Syntax

```
tag Replicate Switchover
```

Privilege Levels

Administrator

Domain Sensitivity

None

Examples

```
7 Replicate Switchover
* 7 Stopping Services
* 7 Installing Replica Store
* 7 Scheduling Reboot
7 OK Completed
```

```
8 Replicate Switchover
8 NO Operation In Process
```

```
9 Replicate Switchover
* 9 MOS Version Mismatch: Store '3.7.5GA' Replica '3.7.4GA'
* 9 Patchlist Mismatch
* 9 Replica Missing: "E3_Hash_2"
* 9 Replica Missing: "E3_FooBar_7"
* 9 Replica Missing: "E3_Widgit_9"
* 9 Replica Added: "debug-1"
* 9 Replica Added: "E3_Widgit_8"
9 NO Replica Not Ready
```

Sync

The `Sync` command:

- Confirms that the Master is still able to transfer data to the Replica.
- Transfers configuration data to the store with the image backup prep utility.
- Records a timestamp.
- Sets a snapshot on the store.
- Scans the snapshot for changes made since the last `Sync` operation, or for the initial `Sync` since the system inception.

You run the `Sync` command on the Master server. After the scan is complete and files are analyzed, the necessary data gets copied to the Replica server. Because the snapshot, scan, and data transfer to the Replica can take a comparatively long time to complete, the `Sync` command returns immediately and reports its termination status in a log message. To check progress of a `Sync`, use the `Status` command.

Syntax

```
tag Replicate Sync
```

Privilege Levels

Administrator

Domain Sensitivity

None

Examples

```
2 Replicate Sync
* 2 Sync Started
2 OK Completed

3 Replicate Sync
3 NO Replication Not Initiated

4 Replicate Sync
4 NO Host Not Master

5 Replicate Sync
5 NO Sync Already Running

6 Replicate Sync
6 NO Snapshot In Use

7 Replicate Sync
7 NO Comm Failure

8 Replicate Sync
8 NO Insufficient Master Disk Space

9 Replicate Sync
9 NO Insufficient Replica Disk Space

10 Replicate Sync
10 NO Replicate Versions Incompatible
```

To

You run the `Replicate To` command once on the Master to prepare the system for sending data to the Replica. The `Replicate To` command:

- ◆ Ensures that no replication has been previously configured.
- ◆ Transfers configuration data to the Replica.
- ◆ Ensures that the Master and Replica have compatible software and sufficient hardware.
- ◆ Tests that the Master can transfer files appropriately to the Replica.

Syntax

```
tag Replicate To replicaHost
```

where *replicaHost* is the hostname or IP address of the Replica system.

Privilege Levels

Administrator

Domain Sensitivity

None

Examples

```
2 Replicate To replica.example.com
2 OK Completed
```

```
3 Replicate To 192.168.3.4
3 OK Completed
```

```
4 Replicate To replica2.example.com
4 No Comm Failure
```

```
5 Replicate To replica.example.com
5 No Replication Already Initiated
```

Version

Displays the version of RSR software running on either Master or Replica system.

Syntax

```
tag Replicate Version
```

Privilege Levels

Administrator

Domain Sensitivity

None

Examples

```
2 Replicate Version
2 1.0
2 OK Completed
```

51

The Reputation Command

Mirapoint Reputation Hurdle is a global IP reputation service that determines the legitimacy of the connecting email sender, providing an additional defense layer to the RazorGate security appliance.

Serving as the first layer of protection against spam, phishing and malware including: worms, Trojan horses, rootkits, spyware, adware and zombies/botnets, Reputation Hurdle removes up to 80 percent of unwanted threats by blocking malicious email before it reaches the mail system.

Reputation Hurdle identifies the good and bad senders at the network perimeter, granting optimal access to legitimate messages while blocking malicious email. Through a technology called Recurrent Pattern Detection, Reputation Hurdle identifies and separates good and bad senders and determines trust and threat levels. Sender reputations are constantly analyzed and updated in real-time. If the sender has a reputation for sending UCE or other malicious content, it is throttled down, quarantined, or rejected. Once a message clears at the gate, it gets passed into the next line of defense where it will go through RazorGate's antivirus and antispam scanning engines.

Additional benefits of the Mirapoint Reputation Hurdle include:

- Sender throttling based on Reputation or IP Address/CIDR
- IP address throttling
- Detection of zombie attacks within minutes
- Reduction of mail volume saving valuable bandwidth and backend filtering resources
- Monitoring of billions of messages per day to compute sender reputation scores
- Real-time adjustments to changing reputations
- Increases security
- Saves disk space and server capacity

The Reputation Hurdle service requires a license. Without a license, the `Reputation` command is hidden and cannot be used.

Reputations

The reputation values are computed from worldwide real-time data about email senders. Depending on the past email sending history of the connecting sender, it is classified into one of the following categories (which also serve as parameters for some subcommands):

- `good-1`
- `fair-1`
- `neutral-1`
- `poor-1`
- `bad-1`
- `worst-1`

Domain Sensitivity

These commands impact the licensed appliance only and not individual domains.

Subcommands

Addfixed

Adds the specified CIDR to the internal database of administrator-set fixed reputations. This allows an administrator to override the Mirapoint Reputation Hurdle global reputation for that CIDR. This can be useful in special situations where the Mirapoint appliance must accept mail from a particular CIDR with a bad reputation, or, the reputation of a particular CIDR is perceived to be worse or better than the global view.

Syntax

```
tag reputation addfixed cidr rep
```

where:

- *cidr* is a specified IP address or IP range.
- *rep* is the reputation value. See Reputations for acceptable reputation values.

Privilege Levels

- Administrator

Domain Sensitivity

None

Example

```
tag reputation addfixed 10.10.245.0/24 good-1
tag OK Completed
```

Countfixed

Returns the count of CIDRs specified by the Administrator stored in the internal database of fixed reputations.

Syntax

```
tag reputation countfixed pattern
```

where *pattern* must be either the null string ("") or asterisk (*) to match everything.

Privilege Levels

- Administrator

Domain Sensitivity

None

Example

```
tag reputation countfixed ""
* tag 1
tag OK Completed
```

Deletefixed

Removes a CIDR from the internal database of administrator-set fixed reputations. This command reverses the action of the `AddFixed` command and is useful when it is no longer necessary to special case the CIDR specified.

Syntax

```
tag reputation deletefixed cidr
```

where `cidr` is a specified IP address or IP range.

Privilege Levels

- Administrator

Domain Sensitivity

None

Example

```
tag reputation deletefixed 10.10.245.0/24
tag OK Completed
```

Getaction

Displays the action(s) the Reputation Hurdle system will take upon receiving a connection from a sender with the specified reputation. The output reflects default settings or actions configured with the `Setaction` command.

Syntax

```
tag reputation getaction rep
```

where `rep` is the reputation value. See Reputations for acceptable reputation values.

Privilege Levels

- Administrator

Domain Sensitivity

None

Example

```
tag reputation getaction worst-1
* tag DROP
tag OK Completed

tag reputation getaction fair-1
* tag spf tag
tag OK Completed
```

Getdelay

Displays the delay associated with the "delay by reputation" action. The delay can range from 0 to 30 seconds. The default delay is 0 for all reputations.

Syntax

```
tag reputation getdelay rep
```

- where *rep* is the reputation value. See Reputations on page 496 for acceptable reputation values.

Privilege Levels

- Administrator

Domain Sensitivity

None

Example

```
tag reputation getdelay poor-1
* tag 15
tag OK Completed
tag reputation getdelay good-1
* tag 0
tag OK Completed
```

Listfixed

Produces two elements per line: the CIDR block and the reputation value.

Syntax

tag `reputation listfixed` *pattern start count*

where:

- *pattern* is a string to match CIDR blocks. You must enter either a null string ("") or an asterisk (*) to match everything.
- *start* is the number of the first block to list. The null string ("") indicates zero (0).
- *count* is the number of blocks to list. The null string ("") indicates all.

Privilege Levels

- Administrator

Domain Sensitivity

None

Example

```
tag reputation listfixed "" "" ""
* tag "10.10.245.0/24" "good-1"
tag OK Completed
```

Query

Queries the current reputation of the given IP address. SMTP service must be running to check a sender's reputation.

Syntax

tag `reputation query` *ip-address*

where *ip-address* is the sender's IP address.

Returns the following information:

- Queried IP address
- Reputation type
- Reputation Source

  - "fixed" —reference id will be none
  - "queried"

- Queried Reference ID
- Reputation action performed while connected to the queried IP address

Privilege Levels

- Administrator

Domain Sensitivity

None

Example

```
tag reputation query 10.10.245.111
* tag "Sender IP: 10.10.245.111"
* tag "Reputation type: good-1"
* tag "Reputation source: Fixed"
* tag "Query reference id: "
* tag "Reputation action: TAG"
tag OK Completed

tag reputation query 59.54.117.49
* tag "Sender IP: 59.54.117.49"
* tag "Reputation type: Neutral-1"
* tag "Reputation source: Queried"
* tag "Query reference id: 0001.0A090304.47F5A36C.0027"
* tag "Reputation action: MAILHURDLE TAG"
tag OK Completed
```

Setaction

Specifies a list of actions to be taken upon connection from a sender with the specified reputation (*rep*).

Syntax

tag reputation setaction *rep* *"action-list"*

where:

- *rep* is the reputation value. See Reputations on page 496 for acceptable reputation values.
- *"action-list"* is a list of actions (no order implied). Enclose the list in double-quotes and separate each action by a space.

  Available actions:

  - `Drop`—causes the server to drop the connection from any server with the specified reputation. If combined with `Delay`, the connection is dropped after the specified delay time.
  - `Permfail`—causes the server to allow the remote server with the specified reputation to attempt to send messages, but the response of the rcpt command is 55x, which rejects the messages with a permanent failure response codes. May only be combined with the `Delay` action.
  - `Tempfail`— causes the server to allow the remote server with the specified reputation to attempt to send messages, but the response of the rcpt command is 45x, which instructs the sending server to resend the messages after some delay. 45x the message until reputation changes. May be combined with the `Delay` action only.
  - `MailHurdle`—bypasses MailHurdle processing unless indicated for a reputation, or unless the message is rejected with a numerical reject code.
  - `SPF`—bypasses SPF (sender policy network) processing unless indicated for a reputation, or unless the message is rejected with a numerical reject code.
  - `Delay`—causes the servers with a set reputation to wait for the specified number of seconds before connecting with the Mirapoint server. Delay by reputation.
  - `Tag`—adds a header. Only applicable where we accept the connection and any messages from the sender, and is ignored when other actions specified include `Drop`, `Permfail`, or `Tempfail`.
  - `Default`—restores the "factory default"action for that reputation level.

Table 5 Default Actions For Each Reputation Type

Reputation Type	Default Action
Good-1	TAG
Fair-1	Delay TAG
Neutral	MH TAG
Poor-1	Tempfail
Bad-1	Permfail
Worst-1	Drop

Privilege Levels

- Administrator

Domain Sensitivity

None

Example

```
tag reputation setaction "fair-1" "spf tag"
tag OK Completed
```

Setdelay

Sets the delay associated with the "delay by reputation" action. This action inserts a small delay into each SMTP transaction at the "dot" if no other action is set; otherwise it inserts the delay before executing other actions. This slows down the traffic to some extent. The delay may be between 0 and 30 seconds.

The default delay is 0 for all reputations and must be configured before this action is useful.

Syntax

```
tag reputation setdelay rep seconds
```

where:

- `rep` is the reputation value. Acceptable values are:
  - `good-1`
  - `fair-1`
  - `neutral-1`
  - `poor-1`
  - `bad-1`
  - `worst-1`
- `seconds` is a time value between 0 and 30.

Privilege Levels

- Administrator

Domain Sensitivity

None

Example

```
tag reputation setdelay "poor-1" 15
tag OK Completed
```

Update

Updates the reputation engine.

Syntax

```
tag reputation update key
```

where `key` value is currently `repdefault`.

Privilege Levels

- Administrator

Domain Sensitivity

None

Example

```
tag reputation update repdefault
tag NO File transfer failed
tag reputation update repdefault
tag OK Completed
```

The Restore Command

The `Restore` command recovers data previously saved by the `Backup` command on a tape drive attached to the local Mirapoint system, or from an RMT archive on a remote Solaris system. See Chapter 7, The Backup Command for details.

Each `Restore` subcommand starts a restore operation and returns immediately; you may log out and log back in again any number of times without interfering with a restore. To monitor the progress of a restore, use `Restore Status`.

Subcommands

Abort

Aborts the restore operation identified by the specified job ID. For the format of the job ID, see `Restore Status`.

Syntax

```
tag Restore Abort jobid
```

where *jobid* is the job ID of the restore operation you want to abort.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
1 Restore Abort Restore-5928
1 OK Completed
```

All

Restores onto your Mirapoint system the entire contents of the backup archive on the specified storage device. The restore operation does *not* delete messages from any mailbox; it adds the backup contents of each mailbox to that mailbox.

Of course, new messages that arrived between the time of backup and the time of restore cannot be recovered. Messaged deleted between those times are recovered, and any folder changes revert to their state at backup time.

This command starts the restore operation and immediately responds with a unique job ID identifying the operation. For the format of the job ID, see `Restore Status`.

Syntax

`tag` `Restore All` *`device options`*

where:

- *`device`* has one of these formats:

  `host-name``:`*`device-name`*

  `user-name``@`*`host-name`*`:`*`device-name`*

  `Tape`

  where:
  - *`host-name`* is the host name of the remote system to which the backup device is attached.
  - *`device-name`* is the full remote system path of the special file referring to the backup device.
  - *`user-name`* is the name of the user identity to assume *on the remote system* when doing the restore. This **backup user** must have read access to *`device-name`*. If you don't specify a backup user, the command uses the default name `mira`.
  - `Tape` is a special keyword identifying the tape drive attached to your Mirapoint system.
- *`options`* is a number indicating block size, or may contain any of the following:
  - `blocksize=` is the block size in bytes to use in reading from the tape device. The empty string (`""`) implies the default block size, 10240 bytes (10 KB) for RMT and 61440 bytes (60 KB) for tape. Tape block size is unalterable, so there is no reason to use this option.
  - `nomessages=` is a directive to restore only the system, user, mailbox, and DL information, not any messages.
  - `continue=` modifies the behaviour of restore when used with a locally attached stacker device. The `true` flag causes the restore to continue when the next tape comes online. The `false` flag gives default behavior of waiting for permission to continue. If the tape does not come online within five (5) minutes, restore fails. There is no grace period: if the stacker has exhausted its supply of tapes, no retries are allowed after refilling the stacker.

  `"(blocksize=`*`bytes`*`)(nomessages=)(continue=true)"`

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Restore All allansun:/dev/rmt/0c 10240
* 2 Restore-5928
2 OK Completed
```

Media

This command can:

- ◆ Report whether the specified restore operation is waiting for you to change the media
- ◆ Inform the backup system that you have changed the media to allow the specified restore operation to resume

For the format of the job ID, see `Restore Status`.

Syntax

tag `Restore Media` *keyword jobid*

where:

- ◆ *keyword* is one of:
  - ❖ `Changed`—Informs the backup system that you have changed the media and instructs the specified restore operation to resume
  - ❖ `Wanted`—Reports whether the restore operation is suspended waiting for you to change the media
- ◆ *jobid* is the job ID of the restore operation for which the media is being used.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Restore Media Wanted Restore-5928
* 3 Waiting for volume #2 for allansun:/dev/rmt/0c.
3 OK Completed
4 Restore Media Changed Restore-5928
4 OK Completed
```

Selective

Restores onto your Mirapoint system only the specified mailboxes from the backup archive on the specified storage device. This command starts the restore operation and immediately responds with a unique job ID identifying the operation. For the format of the job ID, see `Restore Status`.

Syntax

`tag` `Restore Selective` *`device blocksize mailboxes`*

where:

- *`device`* identifies the backup device (see `Restore All`).
- *`blocksize`* is the block size in bytes to use in reading from the backup device. The empty string (`""`) implies the default block size, 10240 bytes (10 KB) for RMT and 61440 bytes (60 KB) for tape. Tape block size is unalterable, so there is no reason to use this option.
- *`mailboxes`* is a quote-quoted, space-separated list of mailbox names relative to the root of the mailbox hierarchy. All user mailboxes must be specified as `user.`*`username`*. If a mailbox name contains a space, you must quote it again with a pair of \" escape-quotes inside the double quotes. If this is specified as `(type=brand)` then only contents of the branding directory `/usr/store/www` are restored.

Privilege Levels

Administrator

Domain Sensitivity

If the current context is a delegated domain, restores data within that domain.

Example

```
5 Restore Selective allansun:/dev/rmt/0c "" "user.al user.sal"
* 5 Restore-5978
5 OK Completed
```

```
6 Restore Selective allansun:/dev/rmt/0c "" "user.al \"user.big dog\" user.sal"
* 6 Restore-5979
6 OK Completed
```

Status

Responds with the current status of all outstanding `Restore` operations. Completed jobs are reported as “Done” only once, so you might see old jobs the first time you run this command, then none at all.

Syntax

`tag` `Restore Status`

Response

Each response line has the form:

`* tag Restore-id completed of total`

or:

`* tag Restore-id status`

where:

- *id* is a unique numeric identifier for the restore operation.
- *completed* is the number of files that have already been restored.
- *total* is the total number of files to be restored.
- *status* is the operation status, which is one of:
    - `done`—the operation completed successfully
    - A message requesting that the restore media be changed (see `Restore Media`)
    - An error message

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
7 Restore Status
* 7 Restore-5928 70001 of 71414
7 OK Completed
```

The Role Command

The `Role` command lets you assign privilege levels to users.

Roles

A **role** is a privilege level for a Mirapoint system user account. Currently, the roles a user may be assigned are, in hierarchy order:

- Administrator—For the primary domain, this role grants a user the ability to use all commands, including halt the system.
- Helpdesk administrator—Grants a user limited administration privileges, like a domain administrator, but across all delegated domains.
- Delegated administrator—Like a system administrator, but with a limited set of privileges only for the current domain.
- Quarantine—Allows a user to access a special quarantine mailbox where messages can be approved for delivery. This uses the same `Filter` quarantine command as Junk Mail Manager (JMM) but the filter is domain-wide.
- Backup operator—Grants a user the ability to perform system backups, and to view related system settings. A user assigned only this role may *not* perform restores, or change system configuration. Backup is unsupported on RazorGate.

Ordinary users (users with no special privileges) are not assigned any role.

Subcommands

Add

Assigns the specified role to a user.

Syntax

```
tag Role Add role user
```

where:

- `role` is one of:
  - `Admin`—Specifies the Administrator role, or Delegated administrator role if a delegated domain is current.
  - `Backup`—Specifies the Backup operator role. Setting not permitted inside a delegated domain.
  - `Helpdesk`—Specifies the Helpdesk role. Setting not permitted inside a delegated domain.
  - `Quarantine`—Specifies the Quarantine role for a user, permitting login to a special account to handle message approval.

  See Roles on page 511 and the *Administrator's Guide* for more information about these roles.

- `user` is the login name of the user to whom you want to assign `role`.

Privilege Levels

- Administrator
- Domain administrator (`Role Add Admin` only)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

A Delegated Domain Administration license is required to add the `Admin` role for a user within a domain.

Example

```
2 Role Add Admin demo
2 OK Completed
```

Count

Responds with the number of users in the specified role.

Syntax

```
tag Role Count role pattern
```

where:

- `role` is one of one of these described under `Role Add`.
- `pattern` is currently ignored.

Privilege Levels

- Administrator
- Domain administrator (`Role Count Admin` only)
- Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
3 Role Count Admin ""
* 3 2
3 OK Completed
```

Delete

Removes a user from the specified role.

You cannot delete the user `Administrator` from any role in your system's primary domain.

If there is only one user in any role in a delegated domain, you cannot delete that user from the role. In other words, once you have added at least one user to a role, there must always be at least one user in that role.

Syntax

```
tag Role Delete role user
```

where:

- *`role`* is one of one of these described under `Role Add`.
- *`user`* is the user that you want to remove from *`role`*.

Privilege Levels

- Administrator
- Domain administrator (`Role Delete Admin` only)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
5 Role Delete Admin demo
5 OK Completed
```

List

Responds with a list of users in the specified role.

Syntax

```
tag Role List role pattern start count
```

where:

- *role* is one of of those described under the `Role Add` command.
- *pattern* is currently ignored.
- *start* is the number of the first user you want to see. The empty string ("") implicitly means 0.
- *count* is the number of users you want to see. The empty string ("") implicitly means all users in the role. If *count* is greater than the total number of users in the role, `list` returns as many users as possible.

Privilege Levels

- Administrator
- Domain administrator
- Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
4 Role List Admin "" "" ""
* 4 "Administrator"
* 4 "demo"
4 OK Completed
```

54

The Schedule Command

The `Schedule` command schedules another command to run at periodic intervals.

Subcommands

Add

Adds a given command to the run schedule. A maximum of 1000 schedules can exist on the system at any one time.

On failure, or when the command generates intermediate output (`*` lines with tag), the output and result of the command goes to the `schedule-output@localhost` DL, which initially contains the user `administrator`. On success (no intermediate output), no email gets sent anywhere.

Syntax

```
tag Schedule Add schedname frequency datespec tag command
```

where:

- *schedname* is the name of this schedule event, using alphanumeric characters only (spaces are not allowed).
- *frequency* must be `hourly`, `daily`, `weekly`, or `monthly`. This argument controls how often the system attempts to run this command.
- *datespec* controls when the command is executed. If frequency is hourly, set this to the minute within the hour when the item should be run. If frequency is daily, set this to the hour of the day when the item should be run. If frequency is weekly, set this to the day of the week when the item should be run (0-6 indicate Sunday to Saturday). If frequency is monthly, set this to the day of the month when the item should be run. The argument "" means 0 (zero).
- *command* is a single scheduled command. For complicated quoting, use the literal string `{n+}` syntax described in Literal Strings on page 48. There is no mechanism for executing a sequence of multiple commands.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

The first example is a CLI command. The remaining protocol examples show how to schedule virus update, message expiration, NTP, and how to quote.

```
> Schedule Add fixtime daily 4
Enter event command: Ntp Synchronize ntp.example.com
OK Completed

2 Schedule Add virusupdate daily 2 ". virus update"
2 OK Completed

3 Schedule Add aging daily 2 ". mailbox msgexpirenow (domains=*)(users=*) 0"
3 OK Completed

4 Schedule Add fixtime hourly 40 ". ntp synchronize ntp.example.com"
4 OK Completed

5 Schedule Add complicated daily 4 {29+}
001 nasty quoting '("*''''''
5 OK Completed
```

Count

Tallies the number of command items currently scheduled.

Syntax

tag Schedule Count *pattern*

where *pattern* must be either "" (null string) or * (asterisk).

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

None

Example

```
6 Schedule Count ""
* 6 3
6 OK Completed
```

Delete

Removes the named item from the system list of scheduled commands.

Syntax

```
tag Schedule Delete schedname
```

where *schedname* is name of an item previously added with `Schedule Add`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
7 Schedule Delete complicated
7 OK Completed
```

Getcommand

Retrieves the administration command associated with the named schedule.

Syntax

```
tag Schedule Getcommand schedname
```

where *schedname* is name of an item previously added with `Schedule Add`.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

None

Example

```
8 Schedule Getcommand virusupdate
* 8 "tag virus update"
8 OK Completed
```

Getdatestring

Retrieves the time specification associated with the named schedule.

Syntax

```
tag Schedule Getdatestring schedname
```

where *schedname* is name of an item previously added with `Schedule Add`.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

None

Example

```
10 Schedule Getdatestring fixtime
* 10 15
10 OK Completed
```

Getfrequency

Retrieves the periodic interval associated with the named schedule.

Syntax

```
tag Schedule Getfrequency schedname
```

where *schedname* is name of an item previously added with `Schedule Add`.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

None

Example

```
11 Schedule Getfrequency fixtime
* 11 "hourly"
11 OK Completed
```

List

Displays the commands currently scheduled on the system.

Syntax

```
tag Schedule List pattern start count
```

where *pattern* must be either "" (null string) or * (asterisk); *start* and *count* specify the beginning and duration of items.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

None

Example

```
8 Schedule List "" "" ""
* 8 virusupdate
* 8 fixtime
8 OK Completed
```

The Service Command

The `Service` command lets you enable, disable, start, and stop email and related services on your Mirapoint system.

Subcommands

Disable

Disables the specified service—the next time the system boots, the service will *not* start automatically.

Syntax

```
tag Service Disable service
```

where *`service`* is one of:

- `Calendar`—the WebCal service (requires a license).
- `Dir`—the Directory service (requires a license).
- `Imap`—the IMAP4 service (requires a license).
- `Ndmp`—the NDMP service.
- `Nis`—NIS login authentication.
- `Pop`—the POP3 service (requires a license).
- `Smtp`—the SMTP service.
- `Snmp`—the SNMP service.
- `Webmail`—service for WebMail (requires a license).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Service Disable Snmp
8 OK Completed
```

Enable

Enables the specified service—the next time the system boots, the service will be started automatically.

Syntax

`tag Service Enable service`

where *service* is one of:

- `Calendar`—the WebCal service (requires a license).
- `Dir`—the Directory service (requires a license).
- `Imap`—the IMAP4 service (requires a license).
- `Ndmp`—the NDMP service.
- `Nis`—NIS login authentication.
- `Pop`—the POP3 service (requires a license).
- `Smtp`—the SMTP service.
- `Snmp`—the SNMP service.
- `Webmail`—service for WebMail (requires a license).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Service Enable Snmp
2 OK Completed
```

Enabled

Responds with the enabled status of the specified service (`YES` or `NO`).

Syntax

`tag Service Enabled service`

where *service* is one of:

- `Calendar`—the WebCal service (requires a license).
- `Dir`—the Directory service (requires a license).
- `Imap`—the IMAP4 service (requires a license).
- `Ndmp`—the NDMP service.
- `Nis`—NIS login authentication.
- `Pop`—the POP3 service (requires a license).
- `Smtp`—the SMTP service.
- `Snmp`—the SNMP service.
- `Webmail`—service for WebMail (requires a license).

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
3 Service Enabled Snmp
* 3 YES
3 OK Completed
```

Listservices

Responds with a list of valid service names. Some services (such as Calendar, Dir, and WebMail) do not appear unless licensed. After licensing, such optional services should be enabled and started.

Syntax

```
tag Service Listservices
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Service Listservices
* 2 calendar
* 2 dir
* 2 imap
* 2 pop
* 2 smtp
* 2 snmp
* 2 nis
* 2 webmail
* 2 ndmp
2 OK Completed
```

Start

Starts the specified service.

Before starting a service, you must enable it using `Service Enable`.

Syntax

tag `Service Start` *service*

where *service* is one of:

- `Calendar`—the WebCal service (requires a license).
- `Dir`—the Directory service (requires a license).
- `Imap`—the IMAP4 service (requires a license).
- `Ndmp`—the NDMP service.
- `Nis`—NIS login authentication.
- `Pop`—the POP3 service (requires a license).
- `Smtp`—the SMTP service.
- `Snmp`—the SNMP service.
- `Webmail`—service for WebMail (requires a license).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Service Start Snmp
4 OK Completed
```

Started

Returns the running status of the specified service (YES or NO).

Before starting a service, you must enable it using `Service Enable`.

Syntax

tag `Service Started` *service*

where *service* is one of:

- `Calendar`—the WebCal service (requires a license).
- `Dir`—the Directory service (requires a license).
- `Imap`—the IMAP4 service (requires a license).
- `Ndmp`—the NDMP service.
- `Nis`—NIS login authentication.
- `Pop`—the POP3 service (requires a license).
- `Smtp`—the SMTP service.
- `Snmp`—the SNMP service.
- `Webmail`—service for WebMail (requires a license).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Service Started Snmp
* 5 YES
5 OK Completed
```

Stop

Stops the specified service.

Syntax

tag `Service Stop` *service*

where *service* is one of:

- `Calendar`—the WebCal service (requires a license).
- `Dir`—the Directory service (requires a license).
- `Imap`—the IMAP4 service (requires a license).
- `Ndmp`—the NDMP service.
- `Nis`—NIS login authentication.
- `Pop`—the POP3 service (requires a license).
- `Smtp`—the SMTP service.

- `Snmp`—the SNMP service.
- `Webmail`—service for WebMail (requires a license).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Service Stop Snmp
6 OK Completed
```

56

The Smtp Command

The `Smtp` command configures behavior of SMTP service on a Mirapoint system. You can manage filter hosts in a multisystem setup, add RBL hosts, test addresses and host status, and set or fetch dozens of service parameters.

Email addresses should use ASCII alphanumeric characters (A-Z a-z 0-9) plus any of the following characters: `+-._`

The following characters are not allowed in email addresses: `!"#$%(),:;<>@[\]‘|`

The following characters might be allowed but are generally not used: `&’*/=?^{}~`

Mirapoint rejects addresses containing ! (bang paths) or % (explicit paths).

To protect against envelope spoofing, you can enable `Smtp Set Smtpauth Required`, `Smtp Set Senderisauth On`, and `Smtp Set Ldapmasqsender On`.

When the SMTP service fails to deliver a message, it usually provides a reason why, often formulated by a non-Mirapoint system. RFC 2821 specifies the numeric reply codes, including 400s for transient errors and 500s for permanent errors. The text strings given with these numeric codes are non-standard and may vary.

On new systems after release 3.8.0, Fastpath is the default mail transfer agent. Fastpath attempts to minimize queueing by processing messages in-memory, but resorts to fast queueing if necessary. Upgraded systems continue using traditional SMTP until you enable Fastpath by running `Smtp Set Fastpath On`. Fastpath offers complete functionality and higher throughput than traditional SMTP, although you might notice some interface differences. If your site has been running traditional SMTP with no problems, and Fastpath has no clear benefits for your organization, Mirapoint suggests you try out Fastpath in a lab setting before moving to it.

DirectPath™ is a Fastpath mode that runs entirely in-memory. It does not support these features: Antispam, Antivirus, autoprovisioning, autoreply, DL expansion, domain filtering, domain signature, forwarding, LDAP mailgroups, LDAP service interruptions, masquerade, more than 20 recipients, RBL header, and quarantine. With DirectPath filtering at end-of-receipt, filtered addresses are taken before alias expansion, so they are DL names instead of DL recipient lists. Customers should enable DirectPath only when recommended by Mirapoint technical staff.

Fastpath and DirectPath each require a license. A Fastpath license is auto-granted on all new systems. A DirectPath license is auto-granted on RazorGate appliances.

Due to recent changes in DNS domain-name resolution on the Internet, nonexistent domain names within the **.com** and **.net** domains, which previously did not resolve, now resolve. Messages sent to nonexistent domains are now sent to a catch-all domain on the Internet, rather than being bounced. Furthermore, SMTP sender validation always succeeds, permitting relay of messages by bogus senders.

To address these problems, Mirapoint released patches and changed Release 3.4 to issue permanent (5xx) errors in such cases. This assures that messages addressed to bogus domains are immediately bounced, and that messages from bogus senders are permanently rejected if sender validation (`Smtp Set Sendercheck`) is enabled.

Mirapoint Added Headers

Mirapoint subsystems may add the following SMTP headers to a message:

- `X-DSN-`*`HeaderString`*, during reprocessing of bounced messages.
- `X-Junkmail`, during domain filtering, antispam scanning, and user filtering. May indicate `Blacklisted` due to `Uce Addexception Blacklist`.
- `X-Junkmail-Info`, after scanning by Antispam Principal Edition, if `Uce Setoption Headerinfo` is enabled.
- `X-Junkmail-IP-Whitelist`, during domain filtering and user filtering, due to `Uce Addexception WhitelistIp`.
- `X-Junkmail-Recipient-Whitelist`, during domain filtering and user filtering, due to `Uce Addexception WhitelistTo`.
- `X-Junkmail-Status`, during antispam scanning to indicate score.
- `X-Junkmail-SD-Raw`, during Signature Edition Rapid Antispam scanning.
- `X-Junkmail-Whitelist`, during domain filtering and user filtering, due to `Uce Addexception Whitelist`.
- `X-Mirapoint-Received-SPF`, during an SPF (sender policy framework) check. It contains the sender's IP address, name from HELO/EHLO used in the SMTP greeting, the complete MAIL-FROM identity, and the enforcement-check result.
- `X-Mirapoint-IP-Reputation`, during a reputation check. It contains the reputation type, reputation source, scoring center's reference ID, and action performed.
- `X-Mirapoint-Old-Envelope-From`, by wiretap to preserve envelope sender.
- `X-Mirapoint-Old-Envelope-To`, by wiretap to preserve envelope recipient.
- `X-old-subject`, after whitelisting to restore original subject line.
- `X-Mirapoint-Virus`, during antivirus scanning to show attachment status.
- `X-Mirapoint-Virus-Scanfailure`, when the virus scanner failed to scan an attachment. For instance, scanning an encrypted EICAR virus produces this. Some customers filter out all messages containing this header.
- `X-XMS` (Mirapoint state) in `MAIL FROM` header, to track scanning and filtering.

Specifying Periods of Time

Several `Smtp` subcommands require you to specify periods of time. You specify time units using these single-character suffixes:

- `w`weeks
- `d`days
- `h`hours
- `m`minutes
- `s`seconds

For example, these time strings would be valid:

- `3w`three weeks
- `6d`six days
- `2h30m`two hours, thirty minutes
- `2m30s`two minutes, thirty seconds

Specify units in decreasing order. For example, `2h30m` is valid, but `30m2h` is not.

Subcommands

Addfilterhost

This command was deprecated in the 3.6 release.

Addlistener

Adds multiple SMTP listeners, possibly associated with interfaces at different IP addresses, to the current host system. Listeners may not conflict with each other. For example, it is an error to have both 10.0.0.9:25 and *:25.

Syntax

```
tag Smtp Addlistener listenerSpec
```

where *`listenerSpec`* takes the form *`IPaddr:Port`* as follows:

- *`IPaddr`* is a valid IP address on which to listen (critical for multi-homed machines) or asterisk (*) to signify SMTP should listen on all interfaces.
- *`Port`* is a valid port number on which to listen, from 1 to 65535.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
33 Smtp Addlistener 10.0.0.9:625
33 OK Completed
```

Addrblhost

Adds the host name of a Realtime Blackhole List (RBL) server to the system's list. This should be called DNSBL for DNS blacklist. You enable RBL checking with the `Smtp Set Rbl` command, but checking does not occur until you add one RBL host.

SMTP service checks RBL hosts for domains that are under boycott for allowing the transmission of unsolicited commercial email, also called spam. You can add up to 8 RBL hosts, but that could slow mail delivery. RBL hosts are checked in the order added until one responds or the list is exhausted. By default the RBL list is empty, although on upgrade previous host names are preserved if RBL is enabled.

If there is a matching entry for the connecting (numeric) IP address in the relay list, then RBL checking is bypassed, allowing you to override entries on the RBL hosts.

Syntax

```
tag Smtp Addrblhost hostname
```

where *hostname* must be a valid Internet address; bare IP addresses are not allowed.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
28 Smtp Addrblhost rbl.mail-abuse.com
28 OK Completed
29 Smtp Addrblhost relays.ordb.org
29 OK Completed
```

Addrtest

This command expands a given address, performs routing and DL expansion, and displays all address to which the mail would be sent, noting the host if applicable, or `Local` for locally delivered mail. As of release 3.4, forwarding and autoreply are done in final delivery, so they are no longer expanded. `Addrtest` does not verify that destination addresses exist, it just expands them.

Syntax

tag Smtp Addrtest *address*

where *address* is the RFC 822 email address that you want to test.

Privilege Levels

- Administrator
- Helpdesk administrator

Domain Sensitivity

The current domain determines the default domain (emulates the sender's domain) for unqualified recipients and distribution lists.

Example

```
27 Smtp Addrtest alias@example.com
* 27 mail.example.com juser@example.com
* 27 Local user2
27 OK Completed
```

Countfilterhost

This command was deprecated in the 3.6 release.

Countlistener

Counts the number of SMTP listeners on the current system.

Syntax

tag Smtp Countlistener *pattern*

where *pattern* may be null string (" ") or asterisk (*) to count all listeners.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
34 Smtp Countlistener *
* 34 2
34 OK Completed
```

Countrblhost

Counts the number of Realtime Blackhole List (RBL) servers in the system list.

Syntax

tag `Smtp Countrblhost`

Privilege Levels

- Administrator
- Helpdesk administrator

Domain Sensitivity

None

Example

```
30 Smtp Countrblhost
* 30 2
30 OK Completed
```

Deletefilterhost

This command was deprecated in the 3.6 release.

Deletelistener

Deletes an SMTP listener, or range of listeners, previously created by `Addlistener`.

Syntax

`tag Smtp Deletelistener` *listenerSpec*

where *listenerSpec* takes the form *IPaddr:Port* as for `Smtp Addlistener`.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
38 Smtp Deletelistener 10.0.0.9:625
38 OK Completed
```

Deleterblhost

Deletes a Realtime Blackhole List (RBL) server from the system's list of host names. Deleting the last RBL server is functionally equivalent to disabling RBL checking.

Syntax

```
tag Smtp Deleterblhost hostname
```

where *hostname* is a host name previously specified with `Smtp Addrblhost`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
39 Smtp Deleterblhost rbl.mail-abuse.com
39 OK Completed
```

Disablelistener

Temporarily disable (without deleting) an SMTP listener, or range of listeners, previously created by `Addlistener`.

Syntax

```
tag Smtp Disablelistener listenerSpec
```

where *listenerSpec* takes the form *IPaddr:Port* as for `Smtp Addlistener`.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
36 Smtp Disablelistener 10.0.0.9:625
36 OK Completed
```

Enablelistener

Enable an SMTP listener, or range of listeners, previously created by `Addlistener` and disabled by `Disablelistener`. Multiple listeners are enabled at creation time.

Syntax

```
tag Smtp Enablelistener listenerSpec
```

where *listenerSpec* takes the form *IPaddr:Port* as for `Smtp Addlistener`.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
35 Smtp Enablelistener 10.0.0.9:625
35 OK Completed
```

Flushloginbeforesmtp

Flush any existing login-before-SMTP records from the system cache; see `Smtp Set LoginbeforeSmtp`. This command may be used to thwart theft-of-service attacks.

Syntax

```
tag Smtp FlushLoginbeforeSmtp
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
26 Smtp FlushLoginbeforeSmtp
26 OK Completed
```

Get

Responds with the value of the specified parameter.

Syntax

tag `Smtp Get` *parameter*

where *parameter* is one of:

- `AcceptPercent`—whether to accept or reject delivery to addresses containing percent sign (%). See `Smtp Set` on page 540.
- `AcceptUucp`—whether to accept or reject delivery to addresses containing exclamation mark (!). See `Smtp Set` on page 540.
- `Bannerdelay`—whether or not to delay the SMTP greeting banner.
- `Bannerdelaytime`—number of seconds to delay SMTP greeting banner.
- `Connectport`—the TCP port on which the SMTP service establishes outgoing connections. See `Smtp Set` on page 540.
- `Customerrorstring`—text string appended to standard error messages that display when temporary (4xx) and permanent (5xx) mail delivery failures are encountered during an SMTP session. See `SMTP Set`.
- `Dnsdisable`—Specifies which email notifications—requested at the SMTP layer for successful or unsuccessful delivery—are not sent. An empty string means that all notifications (Success, Failure and delay) are sent; `"All"` means that all notifications are not sent.
- `Dsnlimit`—the number of delivery status notification (DSN) messages that are sent to one sender in a domain. When this limit is reached, a final DSN message is sent saying that no more status messages will be sent today. When this occurs, a count is kept of each sender, domain, recipient, and reason. The collected DSN information is sent out early the next day.
- `Dsnretrycnt`—the number of retries for DSN messages; 0 means unlimited.
- `Fastpath`—whether SMTP service attempts fast email delivery path or not (`On` or `Off` or `Direct`). See `Smtp Set` on page 540.
- `Identhost`—hostname that appears in the SMTP greeting banner.
- `Ldapmasqsender`—whether message senders are rewritten using the results of LDAP queries (`Off`, `On`, `All`, or `Local`). See `Smtp Set` on page 542.
- `Ldaprouting`—whether message routing using an LDAP database is enabled, and whether only local addresses are LDAP routed (`Off`, `On`, `All`, or `Local`).
- `Ldapmxrouting`—whether LDAP mail routing uses the "MX" record instead of the "A" record for a host (`On` or `Off`). `Ldaprouting` must already be set `On`.
- `Listenport`—deprecated in Release 3.8.1.
- `Lmr`—the local message router (LMR) host that can route messages for all hosts in your organization. Messages that appear local but are not deliverable on the local host are sent to the LMR for routing to the correct host. This is especially useful in multi-tier architectures involving many Mirapoint systems, but is not compatible with domain-based routing.
- `LoginbeforeSmtp`—when this feature is enabled, and the Mirapoint server receives mail from a host that has logged into the POP or IMAP service recently, the host is allowed to send messages without further authentication. The value

of “recently” can be changed with `LoginbeforeSmtpTime` (see below). LDAP entries in multi-tier environments must contain fully-qualified host names.

- `LoginbeforeSmtpServer`—designates host that mediates `LoginbeforeSmtp`.
- `LoginbeforeSmtpTime`—the amount of time for which a `LoginbeforeSmtp` connection persists. The default is three hours; the minimum is one minute.
- `Mailreturn`—the time that the SMTP service tries to deliver a message before sending a bounce message to the sender (see Specifying Periods of Time on page 529).
- `Mailwarn`—the time that the SMTP service tries to deliver a message before sending a warning message to the sender indicating that the message has not yet been delivered (see Specifying Periods of Time on page 529).
- `Masq`—the **masquerade** domain that the SMTP service substitutes or appends to the return address of outgoing messages to make their origin appear uniform. For instance, if you set masquerade to `example.com`, both bare local addresses and hostname-labeled messages appear to originate from *`user`*`@example.com`, regardless of the sending system.
- `Maxmsg`—the maximum message size in bytes that the SMTP service accepts. Messages larger than this size are rejected.
- `Maxmsgcnt`—the maximum number of messages that can be transmitted in a single inbound SMTP session.
- `Maxrecip`—the maximum number of recipients to which the SMTP service will send a message. Refers to the number of recipients specified with SMTP, not the number of recipients after address expansion.
- `Mtaverify`—whether MailHurdle is enabled so incoming SMTP messages can be verified by familiarity or timed-out for retry. See `Smtp Set` on page 544.
- `Nomasquerade`—the given header(s) are not masqueraded, by either the `Masq` or `Ldapmasqsender` option.
- `Omr`—the fully qualified domain name of the outbound message router (OMR). This is the host that handles all outgoing mail. If no OMR is set, the Mirapoint system connects directly to destination hosts to send messages; in this case, `Smtp Get Omr` responds with the empty string.
- `Preservecnames`—retain original CNAME during routing, rather than rewriting to the primary DNS “A” name.
- `Rbl`—whether Realtime Blackhole List (RBL) lookups for message senders is enabled (`On` or `Off`). Messages from senders that are found in the RBL are treated according to the setting of the `Rblhandling` parameter, below.
- `Rblhandling`—when RBL lookups are enabled (see the `Rbl` parameter, above), specifies handling from messages from senders found in the RBL. The value is one of:
  - `Bounce`—rejects message, generating bounce messages to sender and log.
  - `Header`inserts an “X-Junkmail: RBL” header into the message body, allowing the recipient’s message filters to handle the message as desired.

- Receivedforhdrwhether or not to insert a “for” clause in the Received field of message headers. See Smtp Set on page 544.
- Recipientcheckenables immediate rejection of messages addressed to unknown users. See Smtp Set on page 545.
- Securitythe returned value is a quoted, space-separated list of the data-privacy schemes allowed for SMTP connections. The null string ("") indicates cleartext/clearout, which is the default. See Smtp Set on page 545.
- Sendercheckwhether SMTP service validates DNS domain of the address supplied in the envelope MAIL FROM. When On, if the sender’s domain does not exist in DNS, SMTP service rejects the message. See Smtp Set on page 540.
- Senderisauthwhether the sender addresses in message envelopes and From: headers are rewritten using the login name specified through SMTP authentication (On or Off). See Smtp Set on page 545.
- Senderisvalidrecipientdisallows sending of messages by unknown local users, or delivery by forged users. See Smtp Set on page 546.
- Smtpauthwhether and how SMTP authentication (the AUTH extension to ESMTP) is offered. Value can be either On, Off, NORELAYS, or REQUIRED. See Smtp Set on page 546.
- Spf—retrieves the setting of SPF (sender policy framework) enforcement, either Off, Block, or Tag. See Smtp Set.
- Trustfarmmemberswhether or not to deliver messages to a mailbox that is over quota. See Smtp Set on page 547.
- UceBlacklistwhether to consider Uce blacklist during RCPT-TO.
- UceSuspectlistwhether or not MailHurdle suspect list is automatically generated by Antispam scanning software.
- UceWhitelistwhether to consider Uce whitelist during RCPT-TO.
- UceWhitelistIpwhether to consider IP address whitelist during RCPT-TO.
- UceWhitelistTowhether to consider recipient whitelist during RCPT-TO.

Privilege Levels

- Administrator
- Helpdesk administrator (Uce* commands only)
- Backup operator

Domain Sensitivity

None

Example

```
14 Smtp Get Ldapmasqsender
* 14 OFF
14 OK Completed
```

```
15 Smtp Get Ldaprouting
* 15 OFF
15 OK Completed
16 Smtp Get LoginbeforeSmtp
* 16 NO
16 OK Completed
17 Smtp Get Mailreturn
* 17 3d
17 OK Completed
18 Smtp Get Masq
* 18 example.com
18 OK Completed
19 Smtp Get Maxmsg
* 19 64000
19 OK Completed
20 Smtp Get Maxrecip
* 20 100
20 OK Completed
21 Smtp Get Omr
* 21 mail.example.com
21 OK Completed
22 Smtp Get Rbl
* 22 ON
22 OK Completed
23 Smtp Get Rblhandling
* 23 HEADER
23 OK Completed
24 Smtp Get Smtpauth
* 24 OFF
24 OK Completed
25 Smtp Get Sendercheck
* 25 ON
25 OK Completed
```

Hoststat

Prints information about the persistent host database used for SMTP delivery. Lines of output appear in database order, apparently unsorted.

Syntax

tag Smtp Hoststat *host*

where *host* is a pattern representing a host name. The *host* may be empty ("") or an asterisk (*) to display all items. If *host* is supplied as a host name, this command displays information about the specified host only.

The Hoststat subcommand returns a tagged line containing host name, timestamp (last time a connection to the host was attempted or finished), and status message.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
1 Smtp Hoststat ""
* 1 mail.mirapoint.com 01:09:17 "250 ABC33911 Message accepted for delivery"
* 1 mirapoint 01:09:17 "250 2.1.5 Ok"
```

Listfilterhost

This command was deprecated in the 3.6 release.

Listlistener

Returns the IP address and port number of SMTP listeners on the current system.

Syntax

```
tag Smtp Listlistener pattern start count
```

where *pattern* may be null string (“ “) or asterisk (*) to match all listeners, constrained by *start* and *count*.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
35 Smtp Listlistener * ““ ““
* 35 10.0.0.9:625 Enabled
* 35 *:25 Enabled
35 OK Completed
```

Listrblhost

Displays the names of Realtime Blackhole List (RBL) servers in the system list.

Syntax

```
tag Smtp Listrblhost pattern start count
```

where *pattern* must currently be asterisk (*) or the null string (""), and numbers *start* and *count* can specify the start and duration of host names to list.

Privilege Levels

- Administrator
- Helpdesk administrator

Domain Sensitivity

None

Example

```
31 Smtp Listrblhost "" "" ""
* 31 rbl.mail-abuse.com
* 31 relays.ordb.org
31 OK Completed
```

Set

Sets the value of the specified SMTP parameter.

Syntax

tag Smtp Set *parameter value*

where:

- *parameter* is one of:
  - AcceptPercent—whether to accept or reject delivery to addresses containing percent sign (%). Default is No, so subnet-routed messages are normally rejected. If Yes, such messages are accepted (not recommended).
  - AcceptUucp—whether to accept or reject delivery to addresses containing exclamation mark (!). Default is No, so UUCP-style addresses are normally rejected. If set to Yes, such messages are accepted (not recommended).
  - Bannerdelaywhen On, impose a delay before displaying the initial greeting banner for inbound SMTP connections. If a remote mail server attempts to send data before the banner appears, its connection is dropped and the event is logged. Default is Off, which means do not impose a delay.
  - Bannerdelaytimetime duration to delay the SMTP greeting banner. Specify a *value* in seconds. The default is 5 seconds. Setting the delay time to zero (0) resets the delay to 5 seconds.
  - Connectportthe TCP port on which the SMTP service establishes outgoing connections. By default this is port 25, sometimes changed to 26 or 24, but any valid port between 1 and 65535 may be given as *value*. Listenport may use the same port for incoming messages. Specifying an invalid port causes SMTP connections to fail.
  - Customerrorstring—appends a string (maximum 256 characters) to standard error messages that display when temporary (4xx) and permanent (5xx) mail delivery failures are encountered during an SMTP session. The *value* will be the string, which you enclose in double quotes. You can use

the following variables names (which are dynamically replaced with the corresponding data from the attempted mail message):

- `clientip` = IP address of the sending system
- `clientname` = hostname of the sending system
- `error` = SMTP error code number
- `hostname` = hostname of the system receiving the message (and generating the error response)
- `mailfrom` = sender address
- `rcptto` = recipient address
- `reason` = reason for error

Enter `""` (empty string) for *`value`* to display the standard SMTP error responses.

Sample entry:

```
1 Smtp Set Customerrorstring "http://
example.com?s=$(mailfrom)&cip=$(clientip)&smtphost=$(hostname)&clientna
me=$(clientname)&rcpt=$(rcptto)&error=$(error)&reason=$(reason)"

1 OK Completed
```

❖ `Dsndisable`—determines whether delivery status notification (DSN) messages are sent for successful and/or unsuccessful email delivery. You must specify one or more of the following in a space-separated list enclosed in double quotes:
- `All`—Disables all DSN messages. No other options can be set with `All`.
- `Success`—Disables only the DSN message for successful delivery; a DSN is not sent if mail delivery is successful.
- `Failure`—Disables only the DSN message for unsuccessful delivery; a DSN is not sent if an error is encountered during a mail delivery attempt.
- `Delay`—Disables only the DSN mesage for a delayed delivery; a DSN is not sent if mail is delayed.

An empty string (`""`) enables all the DSNs; that is, all of them will be sent.

❖ `Dsnlimit`—the number of delivery status notification (DSN) messages to be sent to one sender in a domain. When this limit is reached, a final DSN message is sent saying that no more status messages will be sent today. After this occurs, a count is kept of each sender, domain, recipient, and reason. The collected DSN information is sent out early the next day. Default `Dsnlimit` is 100, and zero (0) indicates no limit and no summary message.

❖ `Dsnretrycnt`—the maximum number of retries for DSN messages. The default is 0 (unlimited), which means that DSNs are retried until they reach the delivery timeout; see `Smtp Set Mailreturn`. Setting a positive number means that DSN retries will cease after that number is reached.

❖ `Fastpath`—when `On`, causes SMTP to use the fast email path if possible. Turning it `Off` causes SMTP to use the traditional email path exclusively. Fastpath requires a license, automatically granted on new systems since release 3.6. Changing these settings causes restart of SMTP service:

  - Off—Use the traditional email service with queuing. Before release 3.8.0 this was the default, and remains so on upgraded systems.
  - On—Use the fast email path. Fastpath does not support TLS-only in 3.8.x releases, but falls back to traditional SMTP for TLS-only sessions.
  - Direct—Enable this option only as recommended by Mirapoint technical staff. Requires a license, provided on RazorGate appliances. DirectPath handles messages in-memory to avoid message queuing overhead. Before enabling DirectPath, you must set LMR and OMR to the same hostname.
- Identhost—hostname you want to appear when receiving and sending email. This new hostname appears in the SMTP greeting banner (with 220, 221, and 250 responses), and in the HELO or EHLO identification, instead of the actual system hostname. The value of Identhost must be a syntactically correct DNS host.domain name or numeric IP address (such as 10.1.1.1) but does not actually have to exist. By design, the setting of Identhost does not affect Diag Smtpconnect.
- Ldapmasqsender—whether message senders are rewritten using the results of LDAP queries. When this feature is enabled, the SMTP service uses the User:Publishedname query (see Chapter 30, The Ldap Command) to look up the sender's address. If query results are different from the sender's address, the address in the message headers and envelope are rewritten using the results of the query, and the sender's full name is rewritten using the results of the User:Fullname query. Uses the character set associated with Locale Set Unannounced if present. May be:
  - Off—Do not rewrite message headers for any users. The default.
  - On—Rewrite message headers for all users (same as All).
  - All—Rewrite message headers for all users (same as On).
  - Local—Rewrite message headers only for local users.

- Ldaprouting—specifies whether email routing using an external LDAP database is currently enabled, and which addresses are LDAP routed. See Chapter 30, The Ldap Command. You can enable this only if the Mail Routing license is installed on a system; contact your Mirapoint service agent for details. Practical length limits are 127 bytes for the left-hand side and 255 for the right-hand side. Do not use domain-based routing in conjunction with the LMR. Parameter *value* may be one of:
  - Off—Do not attempt LDAP mail routing of messages. The default.
  - On—Same as Local. Supported for backward compatibility.
  - All—Attempt LDAP mail routing for all messages, even those addressed non-locally. This reduces the number of domains you would need to create on the message router with the Local setting. It also permits Antispam scanning of messages addressed non-locally.
  - Local—Route locally addressed messages (intended for mail domains and delegated domains) according to the LDAP values for Mailhost and Routingaddr (or equivalents). Same as On.
  - Quarantine—The system issues a User:QuarantineProfile query to determine where to route mail according to the All value above. If this attribute is not found, uses User:MailProfile routing instead.

- Ldapmxrouting—whether LDAP mail routing uses the "MX" record instead of the "A" record for a host. Default Off. Before enabling, Ldaprouting must already be set to On. When MX routing turned On, SMTP routes messages to the MX record(s) for a given host, using normal MX to "A" record resolution. This setting is useful for redirecting a domain's mail to a different host, or for failover. By default (Off) SMTP routes messages to the address associated with "A" record for the mailhost defined in LDAP for a given address. If the returned host is an IP address, that address is contacted and this setting is ignored.

- Listenport—the TCP port on which SMTP service listens for incoming connections. This could be any port from 1 to 1023 (default 25) excluding in-use ports. Deprecated in release 3.8.1 and replaced by Addlistener.
- Lmr—the local message router (LMR) that can route messages for all hosts in your organization. Default none. Messages that appear local but are not deliverable on the local host are sent to the LMR for routing to the correct host. This may be useful in multi-tier installations involving many Mirapoint systems. Do not use LMR with LDAP domain-based routing: they are incompatible because together they cause mail loops (for example, if the LMR sends a message to a given host due to domain-based route, and the user is nonlocal, the host sends it back to the LMR, and so on).
- LoginbeforeSmtp—when value is set to YES, and SMTP service receives a message from a host that recently logged into POP or IMAP service, senders on that host are allowed to transmit mail without further authentication. The default value is NO. Toggling value restarts POP, IMAP, and WebMail services. The duration of "recently" can be set by LoginbeforeSmtpTime. For three-tier LoginbeforeSmtp (with user proxying), you must first run Trustedhost Add for each cooperating message server. LDAP entries in multi-tier environments must contain fully-qualified host names. Restart SMTP service to reread any LDAP configuration changes.
- LoginbeforeSmtpServer—configures a LoginbeforeSmtp query server. Value can be any valid hostname or IP address. The default is localhost, if not specified. For multi-tier implementations, all systems must use the same query server. This should be set on each cooperating message server.
- LoginbeforeSmtpTime—the duration of a LoginbeforeSmtp session. The value may be specified as d (days), h (hours), or m (minutes) in capitals or lowercase. The default value is three hours; the minimum is one minute.
- Mailreturn—the time that the SMTP service tries to deliver a message before sending a bounce message to the sender (see Specifying Periods of Time on page 529).
- Mailwarn—the time that the SMTP service tries to deliver a message before sending a warning message to the sender indicating that the message hasn't yet been delivered (see Specifying Periods of Time on page 529).
- Masq—the **masquerade** is a DNS domain name to use for rewriting message headers, usually for the sake of uniformity. The masquerade can be appended to unqualified addresses (those not containing @ at signs) or it can replace the right hand side of addresses (after the at sign). For example, @mas02.example.com could be transformed into @example.com by Masq. With a relay host, masquerade may affect message envelopes.
- Maxmsg—the maximum message size in bytes that the SMTP service accepts. Messages larger than this size are rejected. The default maximum message size is 30 MB, and 128 MB is the upper limit you may set. Use M to specify megabytes, for example 64m. Setting zero (0) returns to default.

- `Maxmsgcnt`—the maximum number of messages that can be transmitted in a single inbound SMTP session. This can be set to any positive value up to and including 1000. After exceeding this limit, the connection is dropped and a 451 error results for the next recipient. Default is 0 (unlimited).
- `Maxrecip`—the maximum number of recipients to whom SMTP service will send a message. This setting refers to the number of recipients specified with SMTP, not the number of recipients after address expansion. Default value is 50,000 and may not currently be set higher than that. If you change this setting, restart SMTP service (with `Service Start`) for it to take effect.
- `Mtaverify`—when set to `On`, enables MailHurdle so that incoming SMTP messages can be verified by familiarity or timed-out for retry. Default is `Off`. See "The Mtaverify Command" on page 409.
- `Nomasquerade`—the specified header(s) are not masqueraded, by either the `Masq` or `Ldapmasqsender` option. You can specify one or more of the following in a space-separated list:
  - `From` prevents masquerade of From and Resent-From fields.
  - `Reply-To` stops masquerade of Reply-To and Resent-Reply-To fields.
  - `Sender` prevents masquerade of Sender and Resent-Sender fields.
  - `Disposition-Notification-To` prevents masquerade of that field.
  - `Return-Receipt-To` prevents masquerade of Return-Receipt-To field.
- `Omr`—the fully qualified domain name of the outbound message router (OMR). Default is the null string. (See `Smtp Get`.)
- `Preservecnames`—when `On`, retain the original CNAME during routing, rather than rewriting to the primary DNS "A" name. Default is off.
- `Rbl`—whether Realtime Blackhole List (RBL) lookup for senders is enabled (`On`, `Off`, or `Ignorerelays`). Default is `Off`. The `Ignorerelays` setting is equivalent to `On`, except the RBL check is performed even if the host is on the relay list. Messages from senders that are found in the RBL are treated according to the setting of the `Rblhandling` parameter. You must also add one or more RBL host names; see `Smtp Addrblhost`.
- `Rblhandling`—when RBL lookups are enabled (see the `Rbl` parameter, above), specifies the handling of messages from senders found in the RBL. The value must be one of:
  - `Bounce`—rejects the message, generating a bounce message to both the sender and the SMTP log. This is the default setting. Both messages say "Mail from *IP.address* refused, see RBL server *RBL.hostname*".
  - `Header`—inserts an "`X-Junkmail`: RBL" header into the message, allowing the recipient's filters to dispose of the message as desired. The effect of setting `Rblhandling` to `Header` depends on Antispam settings. Only when Principal Edition Antispam scanning is enabled, RBL is used to calculate the UCE score, and an appropriate `X-Junkmail` header is added based on that score and any whitelist or blacklist settings. With Rapid Antispam, use of RBL header is undefined. If Antispam scanning is unlicensed or not enabled, messages are categorized as junk mail based on RBL alone, and the "`X-Junkmail`: RBL" header is retained.
- `Receivedforhdr`—whether or not to insert "for" clause in the `Received` field of message headers. Turning this option `On` inserts a line showing the address of the intended recipient of each message. This has an impact on SMTP throughput and storage size, since a separate message splits off for each recipient. A unique "for" header is generated for each unique envelope

recipient, even if they are destined for the same user. The “for” header contains the original envelope recipient as received by SMTP, before any mail routing or mailgroup/DL expansion. Setting this parameter `Off` (the default) disables insertion of the “for” clause. The 4th line below shows an example of the “for” clause:

```
Received: from omr.outside.com (omr.outside.com [10.0.0.1])
          by mail.example.com (Example Corp.)
          with ESMTP id AAA00001
          for juser@mail.example.com;
          Tue, 12 Dec 2000 11:43:49 -0800 (PST)
```

❖ `Recipientcheck`—when `On`, enables immediate rejection of messages addressed to unknown users. Default is `Off`. An address after “RCPT TO:” that is destined for a locally known domain, or where the domain portion of the address matches the current host, is considered a local address. With `Ldaprouting` off, messages for local addresses are accepted if a matching account or DL exists on the local system; messages for non-local addresses are relayed normally, possibly to the LMR. With `Ldaprouting` enabled, messages for local addresses are accepted if an LDAP record exists for the address or a matching account exists on the system. With the `Local` option, messages for non-local addresses are relayed normally. With the `All` option, messages for non-local addresses are rejected if no matching LDAP record exists, otherwise they are relayed normally. When `Recipientcheck` rejects a message, the sender gets the error “User unknown” (or “No such mailbox” with Fastpath turned off).

❖ `Security`—sets the data-privacy schemes allowed for SMTP connections. All SSL-related options require an encryption license. Mirapoint does not yet support secure SMTP over SSL on port 465. The parameter value must be a quoted, space-separated list including any of the following settings:
- `Secureauthonly`—The SMTP AUTH command will not accept cleartext passwords over an unencrypted connection.
- `Ssl`—Secure Sockets Layer (SSL v2 and v3) and Transport Layer Security (TLS v1) data encryption is allowed for SMTP connections after a `STARTTLS` handshake, usually on port 25. To get this in Mozilla or Thunderbird, go to Account Settings > Outgoing Server SMTP > Edit, then under Security select TLS (not SSL, which uses port 465).
- `Sslout`—SSL (above) is allowed for outbound SMTP connections.
- `Starttls`—Same as `Ssl`; for consistency with IMAP and POP.
- `Starttlsout`—Same as `Sslout`; for consistency with IMAP and POP.
- `""`—The default. In addition to security schemes specified above, the `Smtp` service always allows clear text connections both inbound and outbound (the default `cleartext/clearout`), as required for standards conformance. To disable other security schemes, specify the null string.

❖ `Sendercheck`—whether SMTP service validates the domain part of the address supplied in the envelope MAIL FROM field. Default is `On`. When enabled, if the sender’s domain does not exist in DNS, SMTP service rejects the message. Requires DNS lookup. Can be turned `Off` to accept mail from domains with misconfigured DNS. Note that UCE Blocked Senders and Allowed Senders (see Chapter 66, The Uce Command) operate independently of `Sendercheck`.

❖ `Senderisauth`—specifies whether sender addresses in message envelopes and `From:` headers are rewritten using the login name specified through

SMTP authentication (On or Off). Default off. (See Smtpauth below.) If the connecting system does not authenticate, this setting has no effect.

- Senderisvalidrecipient—prevents sending of messages by unknown local users, or delivery by forged users. Parameter value may be:
  - On—disallows the sending of messages by any unknown user. If the "MAIL FROM:" field in the header contains an unknown local user, the system returns a "550 Not permitted" error. The definition of valid user is the same as for Smtp Set Recipientcheck. Useful only on the OMR.
  - Off—allows sending of email by unknown local users, the default. This is useful for supporting temporary users.
  - Reject—contrary to On, immediately rejects messages from valid local users. It might help to think of this as sender is Not valid recipient. If the "MAIL FROM:" field matches an valid local user, the system returns a "550 Not permitted" error. On an IMR, this filters out forged emails apparently from someone inside the organization, which is a common spam tactic. Never use this setting on a message server or OMR.

- Smtpauth—specifies whether SMTP authentication is offered (Mirapoint systems always accept it). SMTP clients that use the AUTH command to authenticate themselves, giving their IMAP or POP login and password, are allowed to relay messages to other hosts. Parameter value may be one of:
  - Norelays—means that authentication is offered unless the IP address or host name of the connecting host is present in the relay list (see The Relay Command on page 477). This is the default. Some mail clients require users always to authenticate themselves if the SMTP server supports authentication. By not offering authentication to relay hosts, NORELAYS allows users of those mail clients to connect from known hosts on the relay list without having to authenticate.
  - Off—means that SMTP authentication is never offered.
  - On—means that SMTP authentication is always offered.
  - Required—means that authentication is always required for all hosts, except those on the relay list, before acceptance of any (E)SMTP commands except AUTH, EHLO, HELO, NOOP, RSET, STARTTLS, and QUIT. Moreover, authentication is always offered to all hosts, including those on the relay list.

- Spf—specifies handling of SPF (sender policy framework) enforcement.
  - Off—disables SPF enforcement checking.
  - Block—enables SPF enforcement in "block" mode, in which the sender is refused (blocked) if an enforcement check yields one of the following results: FAIL, TEMPERROR, or PERMERROR. When in this mode a log event MTA.SPF.ACTION generates for each message that includes: the triple used for the enforcement check, the RCPT-TO identity (for informational purposes), the result of the enforcement check and the action taken (BLOCK).
  - Tag—enables SPF enforcement checking in "tag and accept" mode, in which the sender is accepted regardless of enforcement-check results, but the received message is tagged with an X-Header (X-Mirapoint-Received-SPF) containing: the sender's IP address, the name from HELO/EHLO used in the SMTP greeting, the complete MAIL-FROM identity, and the enforcement-check result. When in this mode a log

event MTA.SPF.ACTION generates for each message that includes: the triple used for the enforcement check, the RCPT-TO identity (for informational purposes), the result of the enforcement check and the action taken (TAG).

  - `Trustfarmmembers`—trust security and authentication information passed from members of a farm. The LDAP attribute `miException` may then be set to the value `Overridequota` to specify whether or not to deliver messages into a mailbox that is over quota. When `Trustfarmember` is `On`, the system accepts `AUTH=USER` parameters on "Mail From" ESMTP commands if the connection comes from a system that is a farm member. The default setting is `Off`. The farm directory determines membership, based on DNS lookups. It is the job of SMTP to perform translations of the parameter passed via `AUTH=` before sending, including translations between LDAP authentication names, similar to proxying.
  - `UceBlacklist`—whether or not the SMTP service factors in the standard antispam blacklist (see `Uce Addexception` command) in its response to a `RCPT-TO` request. If `On`, when domain blacklisting indicates that the triplet is junk, the recipient is permanently failed with a 550 error. Also applies to local user blacklist. Default is Off. "Prioritize Blocked Senders" in the GUI.
  - `UceSuspectlist`—turns On (enables) or disables automatic generation of the MailHurdle suspect list by Antispam scanning software. Default is Off. When the suspect list reaches 5 KB in size, the oldest entries auto-expire. MailHurdle consideration of the suspect list must also be enabled with the `Mtaverify Set Checksuspectlist` command.
  - `UceWhitelist`—whether or not the SMTP service factors in the standard antispam whitelist (see `Uce Addexception` command) in its response to a `RCPT-TO` request. If `On`, when domain whitelisting indicates that the triplet is definitely not junk, the recipient is not subject to `Mtaverify` checking. Also if a user is network-local, it checks the user whitelist. Takes first precedence. Default is Off. "Prioritize Allowed Senders" in the GUI.
  - `UceWhitelistIp`—same as `UceWhitelist`, but specified by IP address instead of domain name. Second precedence. Default is Off.
  - `UceWhitelistTo`—same as `UceWhitelist` but for the recipient whitelist. However local checking is not done. Third precedence: whitelist overrides blacklist. Default is Off. "Prioritize Allowed Mailing Lists" in the GUI.

- *`value`* is the value you want to assign to *`parameter`*.

Privilege Levels

- Administrator
- Helpdesk administrator (`Uce*` commands and Fastpath only)
- Backup Operator (Fastpath only)

Domain Sensitivity

None. However, `Mtaverify` commands cannot be run inside a domain.

Example

```
2 Smtp Set Dsndisable ""
2 OK Completed

3 Smtp Set Dsndisable "Success Failure Delay"
3 OK Completed

4 Smtp Set Dsndisable "All"
4 OK Completed

5 Smtp Set Ldapmasqsender Off
5 OK Completed

6 Smtp Set Ldaprouting On
6 OK Completed

7 Smtp Set LoginbeforeSmtp Yes
7 OK Completed

8 Smtp Set Mailreturn 3d
8 OK Completed

9 Smtp Set Masq example.com
9 OK Completed

10 Smtp Set Maxmsg 64000
10 OK Completed

11 Smtp Set Maxrecip 100
11 OK Completed

12 Smtp Set Omr mail.example.com
12 OK Completed

13 Smtp Set Rbl On
13 OK Completed

14 Smtp Set Rblhandling Header
14 OK Completed

15 Smtp Set Smtpauth Off
15 OK Completed

16 Smtp Set Sendercheck On
16 OK Completed

17 Smtp Set Trustfarmmembers On
17 OK Completed
```

Testrblhost

Checks an RBL host to see if the specified IP address is listed on its blackhole list.

Syntax

tag Smtp Testrblhost *hostname IPaddr*

where:

- *hostname* should be the host name of a valid RBL server.
- *IPaddr* is the IP address to check. If blank, IP address “127.0.0.2” is used.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
32 Smtp Testrblhost relays.ordb.org 127.0.0.1
* 32 RBL lookup returns negative
32 OK Completed
```

57 The Snmp Command

The `Snmp` command lets you configure the SNMP service on your Mirapoint system. SNMP stands for simple network management (monitoring) protocol.

Before Mirapoint system software release 3.0, it is necessary to restart SNMP service after making changes with `Snmp Add`, `Delete`, `Addtrap`, or `Deletetrap`.

Access Profiles

An **access profile** is a named collection of information that governs an SNMP client's access to specific SNMP objects on your Mirapoint system.

If you don't explicitly define any access profiles using `Add`, the SNMP service allows the `public` SNMP community read access to the entire MIB-II tree.

For MIB (management information base) examples, see http://*host*/help/snmp-mibs. System objects for applications, interfaces, and events (including traps) are fully documented in the largest of these, MASTER-MIB.

Traps

An SNMP **trap** is an asynchronous notification of an event that is sent to specified hosts. The Mirapoint system sends all SNMP traps to all hosts in the trap list managed by the `Snmp Addtrap` and `Snmp Deletetrap` commands. The same events that generate email alerts (refer to monitoring tasks in the *Administrator's Guide*) generate SNMP traps. For explanations of alerts, see these online helps:

```
http://miServer/help/apps/locale/en_US.ISO_8859-1/Alerts/SYSTEM.html
http://miServer/help/apps/locale/en_US.ISO_8859-1/Alerts/RAID.html
http://miServer/help/apps/locale/en_US.ISO_8859-1/Alerts/STANDBY.html
http://miServer/help/apps/locale/en_US.ISO_8859-1/Alerts/UPS.html
```

Although not shown in a MIB, starting or stopping SNMP service generates a trap.

Subcommands

Add

Creates an SNMP access profile.

Syntax

tag Snmp Add *label source community oid read write notify*

where:

- *label* identifies the access profile so you can refer to it later. Space characters are not allowed in this string.
- *source* is the host name or IP address to which you want to grant access. The special value "default" refers to all hosts.
- *community* is the SNMP community name that SNMP clients must specify to be allowed to query your Mirapoint system. Space characters are not allowed in this string.
- *oid* is the ID of the SNMP object to which you want to grant access. The value ".1" specifies the entire MIB-II tree.
- *read* is one of:
  - read—allows read access.
  - noread—disallows read access.
- *write* must currently be nowrite, which disallows write access.
- *notify* must currently be nonotify, which disallows fine-grained control of traps. (Currently, all traps are sent to the hosts in the trap list managed by Snmp Addtrap and Snmp Deletetrap.)

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Snmp Add mylabel default public .1 read nowrite nonotify
2 OK Completed
```

Addtrap

Specifies a host to which SNMP traps should be sent.

Syntax

tag Snmp Addtrap *host community*

where:

- *host* is the fully-qualified name of a host to which SNMP traps are to be sent.
- *community* is the community string sent along with traps. Space characters are not allowed in this string.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
3 Snmp Addtrap test1.example.com public
3 OK Completed
```

Count

Responds with the current number of SNMP access profiles.

Syntax

tag Snmp Count *pattern*

where *pattern* is currently ignored.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
4 Snmp Count ""
* 4 1
4 OK Completed
```

Counttrap

Responds with the number of hosts to which SNMP traps are currently being sent.

Syntax

tag Snmp Counttrap *pattern*

where *pattern* is currently ignored.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
5 Snmp Counttrap ""
* 5 1
5 OK Completed
```

Delete

Deletes the specified access profile.

Syntax

tag Snmp Delete *label*

where *label* identifies the access profile you want to delete.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Snmp Delete mylabel
6 OK Completed
```

Deletetrap

Deletes the specified host from the SNMP trap list.

Syntax

tag Snmp Deletetrap *host*

where *host* is the host that you no longer want to receive SNMP traps.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
7 Snmp Deletetrap test1.example.com
7 OK Completed
```

Get

Responds with the value of the specified parameter.

Syntax

`tag Snmp Get parameter`

where *parameter* is one of:

- `Syscontact`—the system contact information stored in the `system` table in the MIB-II tree.
- `Syslocation`—the system location information stored in the `system` table in the MIB-II tree.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
8 Snmp Get Syscontact
* 8 "Glenn Green"
8 OK Completed
```

List

Responds with the SNMP access profile specified by a valid label, or responds with a list of all valid SNMP access profiles.

Syntax

`tag Snmp List pattern start count`

- *pattern* must currently be a valid SNMP access profile label. If *label* is the empty string, `Snmp List` responds with a list of all valid SNMP access profile labels.
- *start* is the first position in the list of profiles that you want to see. The empty string (`""`) implicitly means `0`.

- *count* is number of lines from the list of profiles that you want to see. The empty string ("") implicitly means all profiles. If *count* is greater than the total number of profiles, `list` returns as many profiles as possible.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
9 Snmp List mylabel "" ""
* 9 mylabel default public .1 read nowrite nonotify
9 OK Completed
```

Listtrap

Responds with a list of hosts, and community of each host, to which SNMP traps are currently being sent.

Syntax

tag Snmp Listtrap *pattern start count*

where:

- *pattern* is currently ignored.
- *start* is the first position in the list of hosts that you want to see. The empty string ("") implicitly means 0.
- *count* is number of hosts that you want to see. The empty string ("") implicitly means all hosts. If *count* is greater than the total number of hosts, `list` returns as many hosts as possible.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
10 Snmp Listtrap "" "" ""
* 10 test1.example.com public
10 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

```
tag Snmp Set parameter value
```

where:

- `parameter` is one of:
  - `Syscontact`—the system contact information stored in the `system` table in the MIB-II tree.
  - `Syslocation`—the system location information stored in the `system` table in the MIB-II tree.
- `value` is the value you want to assign to parameter.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Snmp Set Syscontact "Glenn Green"
5 OK Completed
```

58

The Spf Command

The `SPF` command controls operation of the SPF (sender policy framework) for email on Mirapoint systems.

SPF helps establish the reputation of valid senders, and can be used to exempt them from MailHurdle. Conversely, ReputationHurdle can be used to exempt some IP addresses from SPF checking (see `Reputation Setaction`.) SPF checking must be enabled in the SMTP service to take effect (see `Smtp Set Spf`).

SPF Records in DNS

SPF is an initiative to reduce Internet spam by increasing our ability to verify the sending SMTP server. As part of the effort, your organization should create at least one SPF record in DNS.

You could think of an SPF record as the opposite of an MX record. MX records inform the net what servers **receive** mail for a domain. SPF records say what servers are supposed to **send** mail for a domain.

Suppose a spammer forges an example.com address to send junk mail, connecting from somewhere other than example.com. The email most likely contains a forged MAIL FROM: <person@example.com> header, but the originating IP address is not one of those advertised in example.com's SPF record. Therefore the email is bogus.

SPF breaks email forwarding, but Mirapoint SMTP was changed some time ago to do remailing (changing envelope sender) instead of forwarding.

Mirapoint recommends that you publish a one-line SPF record for each mail domain you wish to protect from being spoofed by spammers. Here is an example:

```
example.com.   1H IN TXT   "v=spf1 MX PTR A:example.com ~all"
```

- TXT indicates a text record. The `v=spf1` designates SPF version 1.
- The MX keyword says that all published MX hostnames are valid emailers.
- The PTR keyword says that hosts with PTR record matching example.com are valid emailers.
- The A: syntax says that all hosts in the domain example.com are valid emailers.
- The `~all` mechanism always matches, with soft fail, and goes at the end of an SPF record. (Soft fail gets all sites to accept email from early adopters like us.)

Some sites prefer to be explicit about which IP addresses send email:

```
smtp.example.com.   1H IN TXT   "v=spf1 ip4:192.168.0.1/16 ~all"
```

Subcommands

Add

Adds the specified host to one of the SPF special-handling lists.

Three items are used to perform an SPF enforcement check: the sender's IP address, the domain name given after an HELO/EHLO request, and the domain name appearing in the MAIL-FROM header. During mail processing, a sender is considered "on the list" if either the MAIL-FROM domain name, or the name given after HELO/EHLO, appears on the enforce or white list.

To avoid duplicate domain or host entries in an SPF list, the domain or host name is resolved using `gethostbyname`. If this call fails, the domain name is rejected with a `NO` error. If it succeeds, the following items are added to the SPF list:

- the given domain name
- all of the given domain name's aliases (if any)
- all IP addresses resolved for the given domain name

Syntax

tag `Spf Add` *parameter domain/host*

where:

- *parameter* may be:
  - `Enforcelist`—SPF enforcement checking is always done on the specified domain or host. The SPF enforcelist is limited to 8192 entries.
  - `Whitelist`—The specified domain or host is always exempt from SPF enforcement checks. The SPF whitelist is limited to 8192 entries.
- *domain/host* is a fully qualified domain or host name, or dotted IP address, of an SMTP server, and must be resolvable by `gethostbyname`.

Privilege Levels

Administrator

Domain Sensitivity

The command is rejected if a delegated domain is current.

Example

```
2 Spf Add Whitelist mail.example.com
2 OK Completed

2a Spf Add Whitelist 205.217.153.166
2a NO Duplicate entry for the whitelist
```

Count

Returns the number of domains or hosts in an SPF special-handling list.

Syntax

```
tag Spf Count parameter pattern
```

where *parameter* may be either `Enforcelist` or `Whitelist`, and *pattern* may be either * or null string (" ").

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

The command is rejected if a delegated domain is current.

Example

```
3 Spf Count Whitelist ""
* 3 1
3 OK Completed
```

Delete

Deletes the specified domain or host as from the SPF special-handling list.

Syntax

```
tag Spf Delete parameter host
```

where *parameter* may be either `Enforcelist` or `Whitelist`, and *domain/host* is a fully qualified domain or host name; see `Spf Add`.

Privilege Levels

Administrator

Domain Sensitivity

The command is rejected if a delegated domain is current.

Example

```
7 Spf Delete Whitelist mail.example.com
7 OK Completed
```

Get

Retrieves value of the specified parameter.

Syntax

tag `Spf Get` *parameter*

where *parameter* is one of those specified under `Spf Set`.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

The command is rejected if a delegated domain is current.

Example

```
6 Spf Get Enforcerelay
* 6 ON
6 OK Completed
```

List

Responds with a list of domains or hosts in an SPF special-handling list.

Syntax

tag `Spf List` *parameter pattern start count*

where *parameter* may be either `Enforcelist` or `Whitelist`, and *pattern* may be either * or " " to list everything; additionally *start* is the first item to list, and *count* is the number of items to list.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

The command is rejected if a delegated domain is current.

Example

```
4 Spf List Whitelist "" "" ""
* 4 mail.example.com
4 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

```
tag Spf Set parameter setting
```

where *parameter* may be one of:

- `Enforcerelay`—(Default *setting* is `Off`) When set `On`, enables SSPF enforcement checks on relay list senders.
- `Enforcetrusted`—(Default *setting* is `Off`) When set `On`, enables SPF enforcement checks on senders in the trusted-host list.

Privilege Levels

Administrator

Domain Sensitivity

The command is rejected if a delegated domain is current.

Example

```
5 Spf Set Enforcerelay On
5 OK Completed
```

Test

Checks the status of SPF enforcement for a reemote sender, given three items.

Syntax

```
tag Spf Test IPaddr FROM HELO
```

where you specify the following:

- `IPaddr`—A string that defines the IP address of the purported sender.
- `FROM`—Specifies the envelope MAIL-FROM address of the purported sender. It must be an email address matching one of the forms:
  - someone@somewhere.node.com
  - someone@1.2.3.4
  - someone [on local host]
- `HELO`—A string that defines DNS hostname or IP address of a remote SMTP server. This string is validated so it must match a DNS hostname or IP address:
  - somewhere.node.com
  - somewhere [the local domain]
  - 1.2.3.4

Privilege Levels

Administrator

Domain Sensitivity

The command is rejected if a delegated domain is current.

Example

```
8 Spf Test 10.0.2.153 switten@doesnotexist.biz example.com
* 8 IP Address: 10.0.2.153
* 8 Sender: switten@doesnotexist.biz
* 8 HELO: example.com
* 8 SPF Response: None (5)
* 8 SPF Errors: None (0)
* 8 SPF Warnings: (2)  No DNS data for 'doesnotexist.biz'.
8 OK Completed
```

59

The Ssh Command

The `Ssh` command retrieves and changes the SSH port number.

Subcommands

Get

Retrieves an SSH value, currently only port number.

Syntax

```
tag Ssh Get Port
```

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
2 Ssh Get Port
* 3 22
3 OK Completed
```

Set

Changes an SSH value, currently only port number.

Syntax

```
tag Ssh Set Port portnum
```

where *portnum* may be 22 or a number from 1025 to 65535, designating a port not already in use. Null string ("") sets the SSH port back to its default value 22. Change of SSH port is recorded in the system log.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Ssh Set Port 622
2 OK Completed
```

60

The Sshkey Command

The `Sshkey` command manages secure keys for SSH logins to Mirapoint systems.

Secure Shell (SSH) Keys

An **SSH key** allows you to use an SSH client to log into your Mirapoint system's command-line interface (CLI) without entering a login name and password. You must generate an SSH key for each user account on each SSH client machine that you want to use to log into the CLI.

To allow SSH logins, your system must have an SSH license installed (see `License Apply`); you must also enable SSH logins for the administration service using `Admin Set Security`.

To generate SSH keys, you must first obtain an SSH client and install it on your desktop system. The Unix **ssh**(1) command is one such client. F-Secure SSH 1.1 and SecureCRT are two clients available for other types of system.

Each SSH client distribution provides a mechanism for generating SSH keys. See your SSH client documentation for details.

Subcommands

Add

Adds an SSH key for the specified user. The key allows the user to log into the command-line interface using SSH without entering a username and password(see Secure Shell (SSH) Keys on page 567).

To allow SSH logins, your system must have an SSH license installed (see `License Apply`); you must also enable SSH logins for the administration service using `Admin Set Security`.

Syntax

```
tag Sshkey Add user key
```

where:

- *user* is the user name for which you want to add an SSH key.
- *key* is the SSH key you want to add. See your SSH client documentation to learn how to generate this key.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own user account)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
1 Sshkey Add demo "512 33
    47512468470018412158392753836473373864077250642460874741357149566647627129
    74430883491346168753715209337559927693547448380390064020473505988462370567
    286519 no comment at this time"
1 OK Completed
```

Count

Responds with the number of SSH keys that the specified user can use for SSH login authentication.

Syntax

```
tag Sshkey Count user pattern
```

where:

- *user* is the user for whom you want the number of SSH keys.
- *pattern* is currently ignored.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- User (can access only his or her own user account)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
2 Sshkey Count demo ""
* 2 1
2 OK Completed
```

Delete

Deletes an SSH key for the specified user.

Syntax

`tag Sshkey Delete user key`

where:

- *`user`* is the user name for which you want to delete an SSH key.
- *`key`* is SSH key you want to delete.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- User (can access only his or her own user account)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
3 Sshkey Delete demo "512 33
    475124684700184121583927538364733738640772506424608747413571495666476271 29
    744308834913461687537152093375599276935474483803900640204735059884623705 67
    286519 no comment at this time"
3 OK Completed
```

List

Responds with a list of SSH keys that the specified user can use for SSH login authentication.

Syntax

`tag Sshkey List user pattern start count`

where:

- *`user`* is the user for whom you want to list the SSH keys.
- *`pattern`* is currently ignored.
- *`start`* is the number of the first SSH key you want to see. The empty string (`""`) implicitly means `0`.
- *`count`* is the total number of SSH keys you want to see. The empty string (`""`) implicitly means all the user's SSH keys. If *`count`* is greater than the user's total number of SSH keys, `list` returns as many SSH keys as possible.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator
- User (can access only his or her own user account)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
4 Sshkey List demo "" "" ""
* 4 "512 33
   47512468470018412158392753836473373864077250642460874741357149566647627129
   74430883491346168753715209337559927693547448380390064020473505988462370567
   286519 no comment at this time"
4 OK Completed
```

61

The Ssl Command

The `Ssl` command manages SSL server certificates on a Mirapoint system.

SSL requires a license. Either strong or weak encryption may be licensed (not both). SSL is a prerequisite for many other features, including secure HTTP connections, and secure shell (SSH), which requires another license.

For practical information about managing SSL certificates, see the Knowledge Base article *Obtaining and Applying an SSL Certificate Using the CLI*.

If you have a license for strong encryption, you have the flexibility to select the ciphers used in SSL encryption. Using the `Ssl Setminkeylen` and `Ssl Setmaxkeylen` commands, you can choose ciphers that use a certain key-length range. Ciphers that fall within the minimum and maximum key lengths inclusive will be used. Use of higher key-length ciphers provides greater security at the cost of speed.

Subcommands

Getauthrequired

Retrieves the value that determines whether authentication is required to enable cipher suites. `On` means authentication is required; `Off` means authentication is not required.

Syntax

```
tag Ssl Getauthrequired
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
7 Ssl Getauthrequired
* 7 ON
```

```
7 OK Completed
```

Getcert

Responds with the current SSL (X.509) certificate corresponding to the specified IP address or hostname. This must be associated with a network interface on the Mirapoint system, or else a warning results. The response is encoded as a literal string in PEM (Privacy-Enhanced Mail) format, a base-64 text encoding. This command should be used only on secure connections.

Syntax

tag Ssl Getcert *interface*

where *interface* is an IP address or fully qualified DNS hostname associated with one of the network interfaces on your Mirapoint system.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
1 Ssl Getcert mail.example.com
* 1 {2126}
-----BEGIN CERTIFICATE-----
MIIDSzCCAjMCDwE...
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQD...
-----END RSA PRIVATE KEY-----
1 OK Completed
```

Getcsr

Responds with a certificate signing request (CSR) corresponding to the certificate associated with the specified interface. Submit this CSR to a certifying authority (CA) when requesting a server certificate. The name or address must be associated with a network interface on the Mirapoint system. The response is encoded as a literal string in PEM (Privacy-Enhanced Mail) format, a base-64 text encoding.

Syntax

tag Ssl Getcsr *interface*

where *interface* is one of:

- An IP address or fully qualified DNS hostname associated with one of the network interfaces on your Mirapoint system.
- "" (empty string)—refers to the system's primary interface.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Ssl Getcsr mail.example.com
* 2 {754}
-----BEGIN CERTIFICATE REQUEST-----
MIIB6jCCAVMCAQAwgakxCz...
-----END CERTIFICATE REQUEST-----
2 OK Completed
```

Getintca

Retrieves the intermediate certifying authority certificate for this system.

Syntax

tag Ssl Getintca *interface*

where *interface* could be "" (to denote the primary interface), a hostname, or an IP address naming an interface on the Mirapoint system. This command returns a literal in standard PEM format.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Ssl Getintca mail.example.com
* 6 {754}
-----BEGIN CERTIFICATE REQUEST-----
MIIB6jCCAVMCAQAwgakxCz...
-----END CERTIFICATE REQUEST-----
6 OK Completed
```

Getmaxkeylen

Responds with the maximum encryption key length (in bits) of a cipher.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
9 Ssl Getmaxkeylen
* 9 128
9 OK Completed
```

```
13 Ssl Getmaxkeylen
13 NO Not licensed for Strong Encryption
```

Getminkeylen

Displays the minimum encryption key length (in bits) of a cipher.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
10 Ssl Getminkeylen
* 10 "64"
10 OK Completed
```

Getversion

Displays currently permitted versions of the SSL/TSL protocol.

Syntax

tag Ssl Getversion

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Ssl Getversion
* 2 "SSLV2 SSLV3 TLSV1"
2 OK Completed
```

```
4 Ssl Getversion
* 4 "SSLV3 TLSV1"
```

```
4 OK Completed
```

Newcert

Generates a new self-certified SSL (X.509) certificate corresponding to the specified interface. The hostname or IP address must be associated with a network interface on the Mirapoint system. Wildcard hostname certificates are supported.

This command deletes existing certificates. Once this happens it is not possible to retrieve the previous certificate. If you must change the hostname or IP address for your system unexpectedly, and you do not have a new certificate, you can use this command to generate a temporary self-certified certificate until you are able to obtain a new certificate from your certifying authority.

If you delete a certificate accidentally using this command, you may be able to work with Mirapoint support to retrieve the pre-existing certificate.

Syntax

```
tag Ssl Newcert interface
```

where *interface* is one of:

- An IP address or fully qualified DNS hostname associated with one of the network interfaces on the Mirapoint system
- `*` —Regenerates certificates for all interfaces associated with the system.
- `""` (empty string)—Regenerates certificates only for the primary interface.
- `(if=`*interface*`)(subject=`*DN*`)(issuer=`*DN*`)`——Where *interface* specifies a network interface as in the first bullet item, and *DN* is a distinguished name in RFC 2253 format. Here is an example *hostname* with RFC 2253 format:

  ```
  "(if=10.0.2.1)(subject=CN=m1.example.com)(issuer=CN=m1.example.com)"
  ```

  If only one of `(subject=)` or `(issuer=)` is present, the other takes the same DN. If the `(if=)` argument is not present, it is interpreted as `(if="")`, which means the primary interface on the system. The given DNs are parsed out and written into the newly-generated certificate. Subsequent `Newcert` commands will not use the same DNs; they must be resupplied on every `Newcert`.

  If the common name (`CN` attribute) is present in the `subject` DN, it must match the hostname associated with the interface. If not, this error results:

  ```
  tag NO Subject's common name must be hostname
  ```

  If the DN fails to parse correctly due to syntax errors, incorrect attributes, or missing `CN` or `Email`, this error results:

  ```
  tag NO Improperly formed DN
  ```

Privilege Levels

Administrator

Domain Sensitivity

None (allowed in a domain)

Example

```
3 Ssl Newcert mail.example.com
3 OK Completed
```

Setauthrequired

Sets cipher usage with or without authentication.

Syntax

tag `Ssl Setauthrequired` *value*

where *value* is either:

- `On`—authentication required.
- `Off`—authentication not required.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Ssl Setauthrequired On
8 OK Completed
```

Setcert

Sets the SSL (X.509) certificate for the specified interface. Use this command to install a certificate generated by a certifying authority (CA). The name or address must be associated with a network interface on the Mirapoint system. If the system has multiple network interfaces, run `Setcert` for each IP address. The supplied certificate must be encoded as a literal string in PEM (Privacy-Enhanced Mail) format, a base-64 text encoding. The command prompts you to enter the certificate, or the output of Ssl Getcert, followed by a dot (.) on a line by itself.

Syntax

tag `Ssl Setcert` *interface certificate*

where:

- `interface` is one of:
  - An IP address or fully qualified DNS hostname associated with one of the network interfaces on your Mirapoint system.
  - `""` (empty string)——refers to your system's primary interface.
- `certificate` is a literal string containing PEM-encoded SSL (X.509) certificate obtained from a certifying authority, or instead, the output of `Ssl Getcert`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Ssl Setcert mail.example.com {754+}
-----BEGIN CERTIFICATE REQUEST-----
MIIB6jCCAVMCAQAwgakxCz...
-----END CERTIFICATE REQUEST-----
4 OK Completed
```

Setintca

Allows customers to set the intermediate certifying authority certificate at the same time they put the global ID on the box.

The system checks the intermediate certifying authority certificate to make sure that it is legitimate and that it matches the signature on the server certificate for the named interface. This implies that you must run `Ssl Setcert` before `Setintca`.

The supplied certificate must be encoded as a literal string in PEM (Privacy-Enhanced Mail) format, a base-64 text encoding. The command prompts you to enter the certificate followed by a dot character (.) on a line by itself.

Syntax

tag `Ssl Setintca` *interface certificate*

where:

- *interface* could be "" (denoting the main interface) or a hostname or IP address naming an interface on the system.
- *certificate* is a literal string consisting of a PEM-encoded SSL (X.509) certificate obtained from a certifying authority (CA).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Ssl Setintca mail.example.com {754+}
-----BEGIN CERTIFICATE REQUEST-----
MIIB6jCCAVMCAQAwgakxCz...
-----END CERTIFICATE REQUEST-----
5 OK Completed
```

Setmaxkeylen

Sets the maximum key length (in bits) of a cipher.

Syntax

tag Ssl Setmaxkeylen *value*

where *value* is a non-negative integer between 40 and 168.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Ssl Setmaxkeylen 128
8 OK Completed

11 Ssl Setmaxkeylen 168
11 NO Not licensed for Strong Encryption
```

Setminkeylen

Sets the minimum key length (in bits) of a cipher.

Syntax

tag Ssl Setminkeylen *value*

where *value* is a non-negative integer between 0 and 128.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
9 Ssl Setminkeylen 64
9 OK Completed

12 Ssl Setminkeylen 128
12 NO Not licensed for Strong Encryption
```

Setversion

Sets permitted versions of the SSL/TSL protocol. This command allows you to either permit all versions (SSLv2, SSLv3, and TLSv1) or to exclude SSLv2 and permit only SSLv3 and TLSv1. By default, all are permitted. Mirapoint recommends the exclusion of SSLv2 because of known vulnerabilities regarding man-in-the-middle attacks. This setting affects inbound and outbound connections.

Syntax

`tag Ssl Setversion "version"`

where *version* can be:

- All versions SSL/TSL versions: `SSLv2`, `SSLv3`, and `TLSv1`. Alternatively, you can simply enter the empty string ("").
- `SSLv3` and `TLSv1s`—Excludes SSLv2.

For both cases, separate each item by a space and enclose in quotes. Order does not matter.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
1 Ssl Setversion ""
1 OK Completed

10 Ssl Setversion "SSLv2 SSLv3 TLSv1"
10 OK Completed

3 Ssl Setversion "SSLv3 TLSv1"
3 OK Completed

5 Ssl Setversion
5 NO Missing required argument to Ssl

6 Ssl Setversion "SSLv3"
6 NO Invalid value
```

```
7 Ssl Setversion "SSLv2"
7 NO Invalid value

8 Ssl Setversion "SSLv2 SSLv3"
8 NO Invalid value

9 Ssl Setversion "SSLv2 TLSv1"
9 NO Invalid value
```

The Stat Command

The `Stat` command reports current statistics about your Mirapoint system activity. Currently, there is only one subcommand, `Get`, which responds with statistics for a specified system parameter.

Many of these same statistics are available, sometimes with slightly different names, in the daily and weekly reports that administrators receive by email. Refer to the *Mirapoint System Administrator's Guide* and the Administration Suite online help for information about daily and weekly reports.

Subcommands

Get

Responds with statistics about the specified service or system parameter, or all parameters if you specify * instead. Counters may wrap to zero after *maxint*, 2^32.

This section lists all official statistics from `Stat Get *`, and some unofficial statistics appearing in various public interfaces. Many other statistics could be available, but are not documented because they are subject to change or removal.

Syntax

tag `Stat Get` *parameter*

where *parameter* is one of:

- *—A special value that refers to all of the following parameters.
- `DIR.AINBOUND`—Total number of inbound directory server connections since start of service.
- `DIR.BINDSSIMPLE`—Number of simple Bind requests since start of service.
- `DIR.BINDSSTRONG`—Number of strong Bind requests since start of service.
- `DIR.OPS`—Total number of directory operations since start of service. This is a 32-bit counter so it could wrap on a busy directory server.
- `DIR.OPSADD`—Total number of Add operations since start of service.
- `DIR.OPSMODIFY`—Total number of Modify operations since start of service.
- `DIR.OPSMODIFYRDN`—Total number of ModifyRDNs since start of service.

- DIR.OPSREMOVE—Total number of Remove operations since start of service.
- DIR.OPSSEARCH—Total number of Search operations since start of service.
- GETMAIL.CONFIGFAILED—Number of times a request could not be processed (before message transfer) due to configuration error. Examples: cannot look up in DNS, wrong user/password combination. This is an accumulating counter.
- GETMAIL.MESSAGES—Number of messages retrieved by Getmail command. This is an accumulating counter.
- GETMAIL.REQUESTS—Number of times users have requested a Getmail Check through WebMail or administration service. This includes successful requests and all configuration or transfer failures, and is an accumulating counter.
- GETMAIL.TOORECENT—Number of users who did a Getmail Check, but were refused because they checked too recently. This is an accumulating counter.
- GETMAIL.TRANSFERFAILED—Number of times a request could not be processed due to an error in transmission. This is an accumulating counter.
- LDAP.AREPLYCACHECNT—Number of autoreply cache lookups.
- LDAP.AREPLYCACHEMS—Milliseconds spent in autoreply cache lookups.
- LDAP.AREPLYCACHEOK—Number of autoreply cache hits.
- LDAP.AREPLYCNT—Number of autoreply lookups.
- LDAP.AREPLYLDAPCNT—Number of autoreply LDAP lookups.
- LDAP.AREPLYLDAPMS—Milliseconds spent in autoreply LDAP lookups.
- LDAP.AREPLYLDAPOK—Number of successful autoreply LDAP lookups.
- LDAP.AREPLYMS—Milliseconds spent in autoreply lookups.
- LDAP.AREPLYOK—Number of successful autoreply lookups.
- LDAP.AUTHCACHECNT—Number of authentication cache lookups.
- LDAP.AUTHCACHEMS—Milliseconds spent in authentication cache lookups.
- LDAP.AUTHCACHEOK—Number of successful authentication cache lookups.
- LDAP.AUTHCNT—Number of authentication operations.
- LDAP.AUTHLDAPCNT—Number of authentication LDAP lookups.
- LDAP.AUTHLDAPMS—Milliseconds spent in authentication LDAP lookups.
- LDAP.AUTHLDAPOK—Number of successful authentication LDAP lookups.
- LDAP.AUTHMS—Milliseconds spent in LDAP authentication lookups.
- LDAP.AUTHOK—Number of successful authentication operations.
- LDAP.BINDANONCNT—Number of anonymous LDAP bind operations.
- LDAP.BINDANONMS—Milliseconds spent in anonymous LDAP bind operations.
- LDAP.BINDANONOK—Number of successful anonymous bind operations.
- LDAP.BINDUSERCNT—Number of user LDAP bind operations.
- LDAP.BINDUSERMS—Milliseconds spent in user LDAP bind operations.

- `LDAP.BINDUSEROK`—Number of successful user LDAP bind operations.
- `LDAP.CACHETIMECOSDN`—Time of last flush of COS table.
- `LDAP.CACHETIMEGROUP`—Time of last flush of mailgroup table.
- `LDAP.CACHETIMEUSER`—Time of last flush of user table.
- `LDAP.COSCNT`—Number of COS lookups.
- `LDAP.COSDNCACHECNT`—Number of COS cache lookups.
- `LDAP.COSDNCACHEMS`—Milliseconds spent in COS cache lookups.
- `LDAP.COSDNCACHEOK`—Number of successful COS cache lookups.
- `LDAP.COSDNLDAPCNT`—Number of COS LDAP lookups.
- `LDAP.COSDNLDAPMS`—Milliseconds spent in COS LDAP lookups.
- `LDAP.COSDNLDAPOK`—Number of successful COS LDAP lookups.
- `LDAP.COSMS`—Milliseconds spent in COS lookups.
- `LDAP.COSOK`—Number of successful COS lookups.
- `LDAP.MGRPCACHECNT`—Number of mailgroup cache lookups.
- `LDAP.MGRPCACHEMS`—Milliseconds spent in mailgroup cache lookups.
- `LDAP.MGRPCACHENEG`—Number of successful lookups after negative cache hit.
- `LDAP.MGRPCACHEOK`—Number of successful mailgroup cache lookups.
- `LDAP.MGRPCNT`—Number of mailgroup lookups.
- `LDAP.MGRPLDAPCNT`—Number of mailgroup LDAP lookups.
- `LDAP.MGRPLDAPINDIRCT`—Number of mailgroup indirect LDAP lookups.
- `LDAP.MGRPLDAPMS`—Milliseconds spent in mailgroup LDAP lookups.
- `LDAP.MGRPLDAPOK`—Number of successful mailgroup LDAP lookups.
- `LDAP.MGRPMS`—Milliseconds spent in mailgroup lookups.
- `LDAP.MGRPOK`—Number of successful mailgroup lookups.
- `LDAP.SYSCONN`—Number of active connections to LDAP subsystem.
- `LDAP.SYSLDAPREQ`—Number of outstanding LDAP requests.
- `LDAP.USERCACHECNT`—Number of user cache lookups.
- `LDAP.USERCACHEMS`—Milliseconds spent in user cache lookups.
- `LDAP.USERCACHEOK`—Number of successful user cache lookups.
- `LDAP.USERCNT`—Number of user lookups.
- `LDAP.USERLDAPCNT`—Number of user LDAP lookups.
- `LDAP.USERLDAPMS`—Milliseconds spent in user LDAP lookups.
- `LDAP.USERLDAPOK`—Number of successful user LDAP lookups.
- `LDAP.USERLDIFOK`—Number of successful user LDIF lookups.
- `LDAP.USERMS`—Milliseconds spent in user lookups.

- `LDAP.USEROK`—Number of successful user lookups.
- `MBOXLISTD.CHANGE`—Number of `CHANGE` operations since boot.
- `MBOXLISTD.CHANGEFAIL`—Number of failed `CHANGE` operations since boot.
- `MBOXLISTD.CHANGEMS`—Milliseconds to perform `CHANGE` operations since boot.
- `MBOXLISTD.GETACL`—Number of `GETACL` operations since boot.
- `MBOXLISTD.GETACLFAIL`—Number of failed `GETACL` operations since boot.
- `MBOXLISTD.GETACLMS`—Milliseconds to perform `GETACL` operations since boot.
- `MBOXLISTD.LIST`—Number of `LIST` operations since boot.
- `MBOXLISTD.LISTFAIL`—Number of failed `LIST` operations since boot.
- `MBOXLISTD.LISTMS`—Milliseconds to perform `LIST` operations since boot.
- `MBOXLISTD.LOCATE`—Number of `LOCATE` operations since boot
- `MBOXLISTD.LOCATEFAIL`—Number of failed `LOCATE` operations since boot
- `MBOXLISTD.LOCATEMS`—Milliseconds to perform `LOCATE` operations since boot
- `MBOXLISTD.MAILHOST`—Number of `MAILHOST` operations since boot,
- `MBOXLISTD.MAILHOSTFAIL`—Number of failed `MAILHOST` operations since boot.
- `MBOXLISTD.MAILHOSTMS`—Milliseconds to perform `MAILHOST` operations since boot.
- `MBOXLISTD.MBOXLCLALL`—Number of local mailboxes cached.
- `MBOXLISTD.MBOXLCLSHARED`—Number of shared local mailboxes cached.
- `MBOXLISTD.MBOXLDAPSHARED`—Number of shared global mailboxes cached.
- `MBOXLISTD.PUBLISH`—Number of `PUBLISH` operations since boot.
- `MBOXLISTD.PUBLISHFAIL`—Number of failed `PUBLISH` operations since boot.
- `MBOXLISTD.PUBLISHLASTMS`—Milliseconds to commit (LDAP write) the last `PUBLISH` operation.
- `MBOXLISTD.PUBLISHMS`—Milliseconds to commit (LDAP write) all `PUBLISH` operations since boot.
- `MBOXLISTD.PUBLISHTIME`—Clock time of the last requested `PUBLISH`.
- `MBOXLISTD.SYNC`—Number of `SYNC` operations since boot.
- `MBOXLISTD.SYNCFAIL`—Number of failed `SYNC` operations since boot.
- `MBOXLISTD.SYNCFETCHLASTMS`—Milliseconds to fetch from LDAP in last `SYNC` operation.
- `MBOXLISTD.SYNCFETCHMS`—Milliseconds to fetch from LDAP in `SYNC` operations since boot.
- `MBOXLISTD.SYNCLASTMS`—Milliseconds to perform last `SYNC` operation.
- `MBOXLISTD.SYNCMS`—Milliseconds to perform `SYNC` operations since boot.
- `MBOXLISTD.SYNCTIME`—Time of the last committed `SYNC` operation.

- MTAVERIFY.ANOTRETRIED—On the Mtaverify server, running total of triplets that never retried delivery, indicating the messages were probably spam.
- MTAVERIFY.APASSED—Running total of messages passed by MailHurdle.
- MTAVERIFY.ATOTAL—Running total of triplets examined by the server.
- MTAVERIFY.INITIALDENY—Number of triplets under initial timeout for retry.
- MTAVERIFY.INITIALACTIVE—Number of triplets under initial entry lifetime.
- MTAVERIFY.ACTIVE—Number of triplets passed under allowed entry lifetime. The MTAVERIFY statistics are visible only on an Mtaverify server.
- NFS.PHYSREAD—Number of physical NFS read operations.
- NFS.PHYSWRITE—Number of physical NFS write operations.
- NFS.RPCTIMEOUTS—Number of RPC timeouts while establishing connections.
- NFS.RPCINVALID—Number of invalid RPC requests during connections.
- NFS.RPCUNEXPECTED—Number of unexpected RPC connection responses.
- NFS.RPCRETRIES—Number of times RPC requests were retried.
- NFS.RPCREQUESTS—Total number of RPC requests handled.
- NFS.OPGETATTR—Number of Getattr (get file attributes) procedure calls.
- NFS.OPGETATTRLAT—Average recent latency in milliseconds of Getattr.
- NFS.OPGETATTRRATE—Average number of Getattr calls per second.
- NFS.OPSETATTR—Number of Setattr (set file attributes) procedure calls.
- NFS.OPSETATTRLAT—Average recent latency in milliseconds of Setattr.
- NFS.OPSETATTRRATE—Average number of Setattr calls per second.
- NFS.OPLOOKUP—Number of Lookup (pathname lookup) procedure calls.
- NFS.OPLOOKUPLAT—Average recent latency in milliseconds of Lookup.
- NFS.OPLOOKUPRATE—Average number of Lookup calls per second.
- NFS.OPREADLINK—Number of Readlink (follow symbolic link) calls.
- NFS.OPREADLINKLAT—Average recent latency in milliseconds of Readlink.
- NFS.OPREADLINKRATE—Average number of Readlink calls per second.
- NFS.OPREAD—Number of Read (file read) procedure calls.
- NFS.OPREADLAT—Average recent latency in milliseconds of Read.
- NFS.OPREADRATE—Average number of Read calls per second.
- NFS.OPWRITE—Number of Write (file write) procedure calls.
- NFS.OPWRITELAT—Average recent latency in milliseconds of Write.
- NFS.OPWRITERATE—Average number of Write calls per second.
- NFS.OPCREATE—Number of Create (file create) procedure calls.
- NFS.OPCREATELAT—Average recent latency in milliseconds of Create.

- NFS.OPCREATERATE—Average number of Create calls per second.
- NFS.OPREMOVE—Number of Remove (delete file or directory) procedure calls.
- NFS.OPREMOVELAT—Average recent latency in milliseconds of Remove.
- NFS.OPREMOVERATE—Average number of Remove calls per second.
- NFS.OPRENAME—Number of Rename (file rename) procedure calls.
- NFS.OPRENAMELAT—Average recent latency in milliseconds of Rename.
- NFS.OPRENAMERATE—Average number of Rename calls per second.
- NFS.OPLINK—Number of Link (create hard link) procedure calls.
- NFS.OPLINKLAT—Average recent latency in milliseconds of Link.
- NFS.OPLINKRATE—Average number of Link calls per second.
- NFS.OPSYMLINK—Number of Symlink (create symbolic link) procedure calls.
- NFS.OPSYMLINKLAT—Average recent latency in milliseconds of Symlink.
- NFS.OPSYMLINKRATE—Average number of Symlink calls per second.
- NFS.OPMKDIR—Number of Mkdir (make directory) procedure calls.
- NFS.OPMKDIRLAT—Average recent latency in milliseconds of Mkdir.
- NFS.OPMKDIRRATE—Average number of Mkdir calls per second.
- NFS.OPRMDIR—Number of Rmdir (remove directory) procedure calls.
- NFS.OPRMDIRLAT—Average recent latency in milliseconds of Rmdir.
- NFS.OPRMDIRRATE—Average number of Rmdir calls per second.
- NFS.OPREADDIR—Number of Readdir (read directory content) calls.
- NFS.OPREADDIRLAT—Average recent latency in milliseconds of Readdir.
- NFS.OPREADDIRRATE—Average number of Readdir calls per second.
- NFS.OPREADDIRPLUS—Number of Readdirplus (extended Readdir) calls.
- NFS.OPREADDIRPLUSLAT—Average recent latency in millisecs of Readdirplus.
- NFS.OPREADDIRPLUSRATE—Average number of Readdirplus calls per second.
- NFS.OPACCESS—Number of Access (check permissions) procedure calls.
- NFS.OPACCESSLAT—Average recent latency in milliseconds of Access.
- NFS.OPACCESSRATE—Average number of Access calls per second.
- NFS.OPMKNOD—Number of Mknod (create device) procedure calls.
- NFS.OPMKNODLAT—Average recent latency in milliseconds of Mknod.
- NFS.OPMKNODRATE—Average number of Mknod calls per second.
- NFS.OPFSSTAT—Number of Fsstat (dynamic filesystem info) procedure calls.
- NFS.OPFSSTATLAT—Average recent latency in milliseconds of Fsstat.
- NFS.OPFSSTATRATE—Average number of Fsstat calls per second.
- NFS.OPFSINFO—Number of Fsinfo (static filesystem info) procedure calls.

- NFS.OPFSINFOLAT—Average recent latency in milliseconds of Fsinfo.
- NFS.OPFSINFORATE—Average number of Fsinfo calls per second.
- NFS.OPPATHCONF—Number of Pathconf (return POSIX info) procedure calls.
- NFS.OPPATHCONFLAT—Average recent latency in milliseconds of Pathconf.
- NFS.OPPATHCONFRATE—Average number of Pathconf calls per second.
- NFS.OPCOMMIT—Number of Commit (write-out cached data) procedure calls.
- NFS.OPCOMMITLAT—Average recent latency in milliseconds of Commit.
- NFS.OPCOMMITRATE—Average number of Commit calls per second.
- RAID.BATTERYSTATUS—Same as RAID.BATTERY1STATUS (see below).
- RAID.BATTERY1STATUS—On certain models, indicates battery condition of the first RAID controller. Here are possible values for battery status:
  - Startup—The system just started, and is still determining its status.
  - Charged—The battery is fully charged.
  - Charging—The battery is charging because it became discharged while the system was off, or during a battery maintenance cycle.
  - Bad—The battery is bad and should be replaced, or possibly it is missing. This condition generates a system alert.
  - None—Battery is missing or the cable connecting it to the system is loose. Generates system alert.
  - Unknown—The battery is in an unknown state. Generates system alert. Please contact Mirapoint Technical Support, because this is not normal.
  - Discharging—The battery is discharging. If due to battery maintenance, usually preceded by Initialize or Recondition (see below).
  - Initialize/Recondition Charging—The initial charging in an initialization or reconditioning cycle.
  - Initialize/Recondition Discharging—The initial discharging before an initialization or reconditioning cycle.
  - Initialize/Recondition Recharging—A recharge during an initialization cycle; this rarely occurs.
  - Recondition Needed—The battery must be recalibrated. Please contact Mirapoint Technical Support, because this is not normal.

  An initialization cycle puts the battery in a variety of states depending on the state it started from. The steps are: Initialize Discharging (skipped if the battery starts at 0), Discharging (also skipped), Initialize Charging, and Charging.
- RAID.BATTERY2STATUS—On certain models, indicates battery condition of the second RAID controller. See above for possible battery status values.
- RAID.BATTERYTIME—Same as RAID.BATTERY1TIME (see below).
- RAID.BATTERY1TIME—On certain models, indicates charge time left in the first RAID controller battery.
- RAID.BATTERY2TIME—On certain models, indicates charge time left in the second RAID controller battery.
- RAID.CAB0FAN—Current status of first disk shelf's fans; value is one of:

  - `0`—OK.
  - 1—A fan has failed.
- `RAID.CAB1FAN`—Current status of second disk shelf's fans; value is one of:
  - `0`—OK.
  - 1—The fan has failed.
- `RAID.CAB2FAN`—Current status of third disk shelf's fans; value is one of:
  - `0`—OK.
  - 1—The fan has failed.
- `RAID.CAB3FAN`—Current status of fourth disk shelf's fans; value is one of:
  - `0`—OK.
  - 1—The fan has failed.
- `RAID.CAB0MONFAIL`—Indicates a communication failure with the first disk shelf's health monitoring module; the value is one of:
  - `0`—OK.
  - 1—The voltage, fan, and temperature values might be unreliable.
- `RAID.CAB1MONFAIL`—Indicates a communication failure with the second disk shelf's health monitoring module; the value is one of:
  - `0`—OK.
  - 1—The voltage, fan, and temperature values might be unreliable.
- `RAID.CAB2MONFAIL`—Indicates a communication failure with the third disk shelf's health monitoring module; the value is one of:
  - `0`—OK.
  - 1—The voltage, fan, and temperature values might be unreliable.
- `RAID.CAB3MONFAIL`—Indicates a communication failure with the fourth disk shelf's health monitoring module; the value is one of:
  - `0`—OK.
  - 1—The voltage, fan, and temperature values might be unreliable.
- `RAID.CAB0TEMP`—The first disk shelf's temperature; the value is one of:
  - `0`—OK.
  - 1—The temperature has exceeded the recommended maximum.
- `RAID.CAB1TEMP`—The second disk shelf's temperature; the value is one of:
  - `0`—OK.
  - 1—The temperature has exceeded the recommended maximum.
- `RAID.CAB2TEMP`—The third disk shelf's temperature; the value is one of:
  - `0`—OK.
  - 1—The temperature has exceeded the recommended maximum.
- `RAID.CAB3TEMP`—The fourth disk shelf's temperature; the value is one of:
  - `0`—OK.
  - 1—The temperature has exceeded the recommended maximum.

- `RAID.CAB0VOLTAGE`—The power supply voltage of the first disk shelf; the value is one of:
  - `0`—OK.
  - 1—Out-of-range.
- `RAID.CAB1VOLTAGE`—The power supply voltage of the second disk shelf; the value is one of:
  - `0`—OK.
  - 1—Out-of-range.
- `RAID.CAB2VOLTAGE`—The power supply voltage of the third disk shelf; the value is one of:
  - `0`—OK.
  - 1—Out-of-range.
- `RAID.CAB3VOLTAGE`—The power supply voltage of the fourth disk shelf; the value is one of:
  - `0`—OK.
  - 1—Out-of-range.
- `RAID.CACHEON`—Same as `RAID.CACHE1ON` (see below).
- `RAID.CACHE1ON`—State of the first battery-backed RAID controller cache:
  - `0`—Write-back.
  - 1—Write-through.
- `RAID.CACHE2ON`—State of the second battery-backed RAID controller cache. See above for possible values.
- `RAID.CMDREAD`—The number of read commands issued to the RAID within the last second.
- `RAID.CMDTOT`—The number of commands of all types issued to the RAID within the last second.
- `RAID.CMDWRITE`—The number of write commands issued to the RAID within the last second.
- `RAID.FAILED`—Returns the number of drives that have failed.
- `RAID.HBA1TEMP`—Returns temperature of the HBA card (if so equipped).
- `RAID.HBA1VOLTAGE`—Returns voltage of the HBA card (if so equipped).
- `RAID.KBREAD`—The number of kilobytes read from RAID within the last second.
- `RAID.KBWRITE`—The number of kilobytes written to RAID within the last second.
- `RAID.LATREAD`—The average read latency in milliseconds.
- `RAID.LATWRITE`—The average write latency in milliseconds.
- `RAID.MAILSTORE`—The percentage of total space currently being used in the mail store disk partition.

- RAID.MAILSTOREBYTES—The number of bytes currently used in the mail store disk partition.
- RAID.MAILSTOREF—The percentage of the maximum allowed number of files currently being used in the mail store disk partition.
- RAID.SYSSTORE—The percentage of total space currently being used in the system disk partition.
- RAID.SYSSTOREF—The percentage of the maximum allowed number of files currently being used in the system disk partition.
- RAID.SYSSTOREFULL—There is no space left on the system disk partition.
- RAID.WARNING—Returns the number of drives that give warnings.
- STANDBY.CHASTEMP—Same as SYSTEM.CHASTEMP but for the failover unit.
- STANDBY.CPUSTATUS1—Same as SYSTEM.CPUSTATUS1 but for failover unit.
- STANDBY.CPUSTATUS2—Same as SYSTEM.CPUSTATUS2 but for failover unit.
- STANDBY.CPUTEMP—Same as SYSTEM.CPUTEMP but for the failover unit.
- STANDBY.CPUTEMP1—Same as SYSTEM.CPUTEMP1 but for the failover unit.
- STANDBY.CPUTEMP2—Same as SYSTEM.CPUTEMP2 but for the failover unit.
- STANDBY.FANC—Same as SYSTEM.FANC but for the failover unit.
- STANDBY.FANC1—Same as SYSTEM.FANC1 but for the failover unit.
- STANDBY.FANC2—Same as SYSTEM.FANC2 but for the failover unit.
- STANDBY.FANMB1—Same as SYSTEM.FANMB1 but for the failover unit.
- STANDBY.FANMB2—Same as SYSTEM.FANMB2 but for the failover unit.
- STANDBY.FANMB3—Same as SYSTEM.FANMB3 but for the failover unit.
- STANDBY.FANMB4—Same as SYSTEM.FANMB4 but for the failover unit.
- STANDBY.MEMORY—Same as SYSTEM.MEMORY but for the failover unit.
- STANDBY.NODESTATUS—Current state of the standby node; the value is one of:
  - 0—Standby does not exist or is not booted. If this is the case, all other STANDBY.* values are undefined.
  - 1—Standby node has booted and is monitoring the active node.
- STANDBY.POWER—Status of power supplies on the standby; see SYSTEM.POWER.
- STANDBY.TOUCH—Same as SYSTEM.TOUCH but for the failover unit.
- STANDBY.UPSCAPACITY—Percent at which the UPS battery of the standby mode is charged. In normal operation this should be close to or at 100%. During a power outage the battery drains until the system shuts down. After the power outage it charges back up to 100%.
- STANDBY.UPSCURRENT—The current being provided by the UPS battery. In normal operation the unit does not run on UPS, so this value is 0. When power fails and the unit runs off UPS, this value is greater than 0. Usually shutdown is immediate, so customers should never see this value greater than 0.
- STANDBY.UPSRUNNING—Same as UPS.RUNNING but for the failover unit.

- STANDBY.UPSSHUTDOWN—Same as UPS.SHUTDOWN but for the failover unit.
- STANDBY.UPSTIME—Same as UPS.TIME but for the failover unit.
- STANDBY.VOLTAGE—Same as SYSTEM.VOLTAGE but for the failover unit.
- SYSTEM.ADMINC—The current number of administration server connections.
- SYSTEM.AGREETCHATTER—Number of SMTP drops caused by Bannerdelay.
- SYSTEM.AMKDELIVERED—The number of kilobytes of email message data delivered to local mailboxes since system inception or counter overflow.
- SYSTEM.AMKIN—The number of kilobytes of email message data received since system inception or counter overflow.
- SYSTEM.AMKOUT—The number of kilobytes of email message data sent out since system inception or counter overflow.
- SYSTEM.AMMSGATTACH—The number of messages received with attachments since system inception or counter overflow. This presents a high-order view of mail attachments. Any MIME-style message part boundary counts
- SYSTEM.AMMSGDELIVERED—Number of emails delivered to local mailboxes since system inception or counter overflow.
- SYSTEM.AMMSGIN—Number of email messages received since system inception or counter overflow.
- SYSTEM.AMMSGOUT—Number of email messages sent since system inception or counter overflow.
- SYSTEM.AMMSGRECP—Number of message recipients since system inception or counter overflow.
- SYSTEM.AMMSGSPAM—Count of messages that Antispam subsystem(s) have classified as junk mail. This increments when a message is marked as spam, under any threshold, and the counter persists across reboot.
- SYSTEM.AMMSGVIRUS—Number of email viruses found since system inception or counter overflow.
- SYSTEM.APKTCOLL—The number of network packet collisions since system inception or counter overflow.
- SYSTEM.APKTIN—The number of inbound network packets since system inception or counter overflow.
- SYSTEM.APKTOUT—The number of outbound network packets since system inception or counter overflow.
- SYSTEM.ASMTPCIN—The number of inbound SMTP connections.
- SYSTEM.ASMTPCOUT—The number of outbound SMTP connections.
- SYSTEM.AVENGINE—Whether or not the antivirus engine's pattern files are up to date.
- SYSTEM.BINWEBMAIL—Block reads for WebMail. Normally a hidden statistic. Other applications have similar SYSTEM.BIN* statistics.
- SYSTEM.BOUTWEBMAIL—Block writes for WebMail. Usually a hidden statistic. Other applications have similar SYSTEM.BOUT* statistics.

- SYSTEM.CHASTEMP—The system chassis temperature; value is one of:
  - 0—OK.
  - 1—The temperature has exceeded the recommended maximum.
- SYSTEM.CONTEXT—Recent context switches in system memory.
- SYSTEM.CPUSTATUS1—The status of processor 1; value is one of:
  - 0—CPU is present and working.
  - 1—CPU is either missing or not functional.
- SYSTEM.CPUSTATUS2—The status of processor 2; value is one of:
  - 0—CPU is present and working.
  - 1—CPU is either missing (on single CPU systems) or not functional.
- SYSTEM.CPUTEMP—The processor temperature; value is one of:
  - 0—OK.
  - 1—The temperature has exceeded the recommended maximum.
- SYSTEM.CPUTEMP1—The temperature of multiple processor 1; value is one of:
  - 0—OK.
  - 1—The temperature has exceeded the recommended maximum.
- SYSTEM.CPUTEMP2—The temperature of multiple processor 2; value is one of:
  - 0—OK.
  - 1—The temperature has exceeded the recommended maximum.
- SYSTEM.DNS*n*RESP—Response time of DNS server(s) as a 2-minute average, where *n* is a number from 1 to 9.
- SYSTEM.FANC—The current status of the CPU fan; value is one of:
  - 0—OK.
  - 1—The fan has failed.
- SYSTEM.FANC1—The current status of multiple CPU fan 1; value as for FANC.
- SYSTEM.FANC2—The current status of multiple CPU fan 2; value as for FANC.
- SYSTEM.FANMB*n*—Current status of motherboard fan *n*, where *n* is a number from 1 to 9; value is one of:
  - 0—OK.
  - 1—The fan has failed.
- SYSTEM.HWMONFAIL—Usually 0. Value is 1 in case of health monitor failure, which generates an alert indicating that the following reported SYSTEM values may be inaccurate: FANCx, FANMBx, CHASTEMP, CPUTEMPx, VOLTAGE, POWER, and CPUSTATUSx.
- SYSTEM.IDLECPU—The percentage of compute cycles that the processor is idle. This number can fluctuate between zero and 100.
- SYSTEM.IDLECPU1—The percentage of cycles that multiple processor 1 is idle.
- SYSTEM.IDLECPU2—The percentage of cycles that multiple processor 2 is idle.
- SYSTEM.IMAPC—The current number of IMAP connections.

- `SYSTEM.IMAPL`—The number of processes listening for IMAP connections.
- `SYSTEM.IMAPLOGINS`—Whether the system has exceeded its IMAP user-limit.
- `SYSTEM.KERB4`*`n`*`RESP`—Response of Kerberos4 server(s) as 2-minute average, where *`n`* is a number from 1 to 9.
- `SYSTEM.KERB5`*`n`*`RESP`—Response of Kerberos5 server(s) as 2-minute average, where *`n`* is a number from 1 to 9.
- `SYSTEM.LDAP`*`n`*`RESP`—Response time of LDAP server(s) as a 2-minute average, where *`n`* is a number from 1 to 9.
- `SYSTEM.LMRRESP`—Response time of SMTP LMR as a 2-minute average.
- `SYSTEM.LMRSTATUS`—Whether the local mail router is accepting messages.
- `SYSTEM.LOAD`—The run-queue of the system averaged over the past minute. This represents the number of processes waiting for resources. Busy systems range from 20 to 50. Anything over 65 is considered overloaded. If you receive an alert based on `SYSTEM.LOAD`, check **Performance Graphs** in the GUI.
- `SYSTEM.MAILQUEUE`—The number of messages in the SMTP delivery queue.
- `SYSTEM.MEMORY`—The status of system main memory; the value is one of:
  - `0`—OK.
  - 1—Memory error.
- `SYSTEM.MKIN`—The average kilobytes per second of email message data received during the last monitoring interval.
- `SYSTEM.MKOUT`—The average kilobytes per second of email message data sent during the last monitoring interval.
- `SYSTEM.MMSGIN`—The average number of email messages per second received during the last monitoring interval.
- `SYSTEM.MMSGOUT`—The average number of email messages per second sent during the last monitoring interval.
- `SYSTEM.NETMEDIA`—The current Ethernet media speed (100 is 100-Base-T).
- `SYSTEM.NIS`*`n`*`RESP`—Response time of NIS server(s) as a 2-minute average, where *`n`* is a number from 1 to 9.
- `SYSTEM.NTP`*`n`*`RESP`—Response time of NTP server(s) as a 2-minute average, where *`n`* is a number from 1 to 9.
- `SYSTEM.OMRRESP`—Response time of SMTP OMR as a 2-minute average.
- `SYSTEM.PKTCOLL`—The average number of network packet collisions per second during the last monitoring interval.
- `SYSTEM.PKTIN`—The average number of inbound network packets per second during the last monitoring interval.
- `SYSTEM.PKTOUT`—The average number of outbound network packets per second during the last monitoring interval.
- `SYSTEM.POPC`—The current number of POP connections.
- `SYSTEM.POPL`—The number of processes listening for POP connections.

- SYSTEM.POPLOGINS—Whether the system has exceeded its POP user-limit.
- SYSTEM.POWER—Current status of power supplies. Before 5-series hardware, there was only one statistic even if there were two power supplies. With 5-series and later hardware, POWER1 and POWER2 are reported separately. Value is one of:
    - 0—OK.
    - 1—The power supply has failed.
- SYSTEM.POWER1—Current status of power supply one; values as for POWER.
- SYSTEM.POWER2—Current status of power supply two; values as for POWER.
- SYSTEM.RADIUS*n*RESP—Response of Radius server(s) as 2-minute average, where *n* is a number from 1 to 9.
- SYSTEM.RBL*n*RESP—Response time of RBL server(s) as a 2-minute average, where *n* is a number from 1 to 9.
- SYSTEM.ROUTERRESP—Response time of default gateway as 2-minute average.
- SYSTEM.SMTPC—The current number of SMTP connections.
- SYSTEM.SMTPL—The number of processes listening for SMTP connections.
- SYSTEM.SSLC—The current number of SSL connections.
- SYSTEM.TIMAPC—The total number of IMAP connections since system installation. In weekly-reports, IMAPCONN represents the number of connections over the past hour.
- SYSTEM.TIMAPPBLOG—The total number of bad IMAP Proxy logins (invalid credentials) to a primary message server.
- SYSTEM.TIMAPPFAIL—The total number of failed IMAP commands (any IMAP command) to a primary message server.
- SYSTEM.TIMAPPLOG—The total number of successful IMAP Proxy logins to a primary message server.
- SYSTEM.TIMAPPOPS—The total number of IMAP commands issued to a primary message server.
- SYSTEM.TIMAPPOPSFAIL—The total number of failed IMAP commands issued to a primary message server.
- SYSTEM.TIMAPRBLOG—The total number of bad IMAP Proxy logins (invalid credentials) to a remote message server.
- SYSTEM.TIMAPRFAIL—The total number of failed IMAP commands (any IMAP command) to a remote message server.
- SYSTEM.TIMAPRLOG—The total number of successful IMAP Proxy logins to a remote message server.
- SYSTEM.TIMAPROPS—The total number of IMAP commands issued to a remote message server.
- SYSTEM.TIMAPROPSFAIL—The total number of failed IMAP commands issued to a remote message server.
- SYSTEM.TOTALINBOXES—The total number of system inboxes.

- `SYSTEM.TOTALUSERS`—The total number of system users in all domains, same as `User Count "*@*"` command.
- `SYSTEM.TOUCH`—Current status of the touch screen; value is one of:
  - `0`—OK.
  - 1—The touch screen has failed.
- `SYSTEM.TPOPC`—Total number of POP connections since system installation. In weekly-reports, `POPCONN` is the number of connections over the past hour.
- `SYSTEM.UCE1`—Count of messages scored 1-10 as junk mail since boot.
- `SYSTEM.UCE2`—Count of messages scored 11-20 as junk mail since boot.
- `SYSTEM.UCE3`—Count of messages scored 21-30 as junk mail since boot.
- `SYSTEM.UCE4`—Count of messages scored 31-40 as junk mail since boot.
- `SYSTEM.UCE5`—Count of messages scored 41-50 as junk mail since boot.
- `SYSTEM.UCE6`—Count of messages scored 51-60 as junk mail since boot.
- `SYSTEM.UCE7`—Count of messages scored 61-70 as junk mail since boot.
- `SYSTEM.UCE8`—Count of messages scored 71-90 as junk mail since boot.
- `SYSTEM.UCE9`—Count of messages scored 91-150 as junk mail since boot.
- `SYSTEM.UCE10`—Count of messages scored > 150 as junk mail since boot.
- `SYSTEM.UPTIME`—Cumulative time the system has been running since boot. The value is a string of the format:

  "*`sec converted`*"

  where:
  - *`sec`* is the time in seconds
  - *`converted`* is the time displayed in a more readable form, using the following unit suffixes:
    - `d`—days
    - `h`—hours
    - `m`—minutes
    - `s`—seconds

  For example, `6d20h5m5s` indicates 6 days, 20 hours, 5 minutes, and 5 seconds.
- `SYSTEM.VOLTAGE`—The system power supply voltage; the value is one of:
  - `0`—OK.
  - 1—Out-of-range.
- `SYSTEM.WEBMLLOGINS`—The system has exceeded its WebMail user-limit.
- `UPS.RUNNING`—Whether the system is running on the UPS battery; the value is one of:
  - `0`—The system is not running on battery.
  - Value greater than 0—The system is running on battery.
  - `-1`—Cannot determine whether the system is running on battery.

- UPS.SHUTDOWN—Battery charge as a percentage of capacity; this value is used to determine whether to shut down or start the system.
- UPS.TIME—The approximate time in minutes that the UPS could sustain the system if the power were to be disrupted. If this is only 10 or 15 minutes, the batteries are reaching their end of life, so you should consider changing them.
- WEBMAIL.ACTIVE05—Number of WebMail sessions active in last 5 minutes. Note that WebMail sessions are persistent across reboot, so WebMail statistics are not reset at boot time.
- WEBMAIL.ACTIVE60—Number of WebMail sessions active in the past hour but inactive during the last 5 minutes (mutually exclusive with ACTIVE05).
- WEBMAIL.CHECKMAIL—Number of CheckMail link-clicks.
- WEBMAIL.CHECKMAILMS—Milliseconds to process CheckMails.
- WEBMAIL.DORMANT—The number of WebMail sessions inactive for more than an hour (mutually exclusive with ACTIVE05 and ACTIVE60). After 6 hours, an inactive WebMail session drops out of the statistics.
- WEBMAIL.FOLDERPAGE—Number of page up/down clicks.
- WEBMAIL.FOLDERPAGEMS—Milliseconds to process WebMail page up/down.
- WEBMAIL.JSEXTRAS—Number of jsextras.js requests.
- WEBMAIL.JSEXTRASMS—Milliseconds to process jsextras.js.
- WEBMAIL.LOGIN—Number of user logins to WebMail.
- WEBMAIL.LOGINMS—Milliseconds to process WebMail logins.
- WEBMAIL.MSGDEL—Number of WebMail messages deleted.
- WEBMAIL.MSGDELMS—Milliseconds to process WebMail deletes.
- WEBMAIL.MSGREAD—Number of WebMail messages read.
- WEBMAIL.MSGREADMS—Milliseconds to process WebMail reads.
- WEBMAIL.MSGSENT—Number of WebMail messages replied-to.
- WEBMAIL.MSGSENTMS—Milliseconds to process WebMail replies.
- WEBMAIL.SORT—Number of WebMail sort operations.
- WEBMAIL.XMLBODYSTRUCT—Number of bodystructure.xml requests.
- WEBMAIL.XMLBODYSTRUCTMS—Milliseconds to process bodystructure.xml.
- WEBMAIL.XMLEXPUNGE—Number of expunge.xml requests.
- WEBMAIL.XMLEXPUNGEMS—Milliseconds to process expunge.xml.
- WEBMAIL.XMLGETSID—Number of getsid.xml requests.
- WEBMAIL.XMLGETSIDMS—Milliseconds to process getsid.xml.
- WEBMAIL.XMLINDEX—Number of index.xml requests.
- WEBMAIL.XMLINDEXMS—Milliseconds to process index.xml.
- WEBMAIL.XMLRFC822—Number of rfc822.xml requests.

- WEBMAIL.XMLRFC822MS—Milliseconds to process rfc822.xml.
- WEBMAIL.XMLSEARCH—Number of search.xml requests.
- WEBMAIL.XMLSEARCHMS—Milliseconds to process search.xml.
- WEBMAIL.XMLSETFLAGS—Number of setflags.xml requests.
- WEBMAIL.XMLSETFLAGSMS—Milliseconds to process setflags.xml.
- WEBMAIL.XMLSORT—Number of sort.xml requests.
- WEBMAIL.XMLSORTMS—Milliseconds to process sort.xml.
- WEBMAIL.XMLSTATUS—Number of status.xml requests.
- WEBMAIL.XMLSTATUSMS—Milliseconds to process status.xml.
- WEBMAIL.XMLVERSID—Number of verifysid.xml requests.
- WEBMAIL.XMLVERSIDMS—Milliseconds to process verifysid.xml.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
Stat Get *
DIR.AINBOUND 2
DIR.BINDSSIMPLE 0
DIR.BINDSSTRONG 0
DIR.OPS 0
DIR.OPSADD 0
DIR.OPSMODIFY 0
DIR.OPSMODIFYRDN 0
DIR.OPSREMOVE 0
DIR.OPSSEARCH 0
RAID.CAB0FAN 0
RAID.CAB0MONFAIL 0
RAID.CAB0TEMP 0
RAID.CAB0VOLTAGE 0
RAID.CAB1FAN undef
RAID.CAB1MONFAIL undef
RAID.CAB1TEMP undef
RAID.CAB1VOLTAGE undef
RAID.CAB2FAN undef
RAID.CAB2MONFAIL undef
RAID.CAB2TEMP undef
RAID.CAB2VOLTAGE undef
RAID.CAB3FAN undef
RAID.CAB3MONFAIL undef
RAID.CAB3TEMP undef
RAID.CAB3VOLTAGE undef
RAID.FAILED 0
RAID.MAILSTORE 7
RAID.MAILSTOREBYTES 334863360
RAID.MAILSTOREF 0
RAID.SYSSTORE 83
RAID.SYSSTOREF 12
RAID.SYSSTOREFULL 83
RAID.WARNING 0
STANDBY.NODESTATUS 0
SYSTEM.ADMINC 1
SYSTEM.AMKDELIVERED 180
SYSTEM.AMKIN 0
SYSTEM.AMKOUT 26
SYSTEM.AMMSGATTACH 11
SYSTEM.AMMSGDELIVERED 22
SYSTEM.AMMSGIN 0
SYSTEM.AMMSGOUT 6
SYSTEM.AMMSGRECP 28
SYSTEM.AMMSGSPAM undef
SYSTEM.AMMSGVIRUS 0
SYSTEM.APKTCOLL 0.00
SYSTEM.APKTIN 525576.00
SYSTEM.APKTOUT 132753.00
SYSTEM.ASMTPCIN 1
SYSTEM.ASMTPCOUT 0
SYSTEM.CHASTEMP 0
SYSTEM.CPUTEMP 0
SYSTEM.CPUTEMP1 0
SYSTEM.DNS1RESP eserv.example.com 0.073417
SYSTEM.FANC 0
SYSTEM.FANC1 0
SYSTEM.FANMB1 0
SYSTEM.FANMB2 0
SYSTEM.FANMB3 undef
SYSTEM.HWMONFAIL 0
SYSTEM.IDLECPU 99.07
SYSTEM.IDLECPU1 98.91
SYSTEM.IDLECPU2 0.00
SYSTEM.IMAPC 0
SYSTEM.IMAPL 1
SYSTEM.LOAD 0.11
SYSTEM.MAILQUEUE 0
SYSTEM.MEMORY 0
SYSTEM.MKIN 0
SYSTEM.MKOUT 0
SYSTEM.MMSGIN 0
SYSTEM.MMSGOUT 0
SYSTEM.NTP1RESP gw.example.com 0.393333
SYSTEM.PKTCOLL 0.00
SYSTEM.PKTIN 3.80
SYSTEM.PKTOUT 1.20
SYSTEM.POPC 0
SYSTEM.POPL 1
SYSTEM.POWER 0
SYSTEM.ROUTERRESP gw2.example.com 0.236667
SYSTEM.SMTPC 0
SYSTEM.SMTPL 1
SYSTEM.SSLC 0
SYSTEM.TIMAPC 1
SYSTEM.TOUCH 0
SYSTEM.TPOPC 1
SYSTEM.UCE1 undef
SYSTEM.UCE10 undef
SYSTEM.UCE2 undef
SYSTEM.UCE3 15
SYSTEM.UCE4 undef
SYSTEM.UCE5 undef
SYSTEM.UCE6 undef
SYSTEM.UCE7 undef
SYSTEM.UCE8 undef
SYSTEM.UCE9 undef
SYSTEM.UPTIME  417493 4d19h58m13s
SYSTEM.VOLTAGE 0
UPS.RUNNING undef
UPS.SHUTDOWN undef
UPS.TIME undef
WEBMAIL.ACTIVE05 undef
WEBMAIL.ACTIVE60 undef
WEBMAIL.DORMANT undef
WEBMAIL.SORT 0
OK Completed
```

63

The Storage Command

The `Storage` command manages different types of storage on Mirapoint systems: direct-attached RAID (redundant array of independent disks), fabric-attached SAN (storage area network) and ethernet-connected NAS (network attached storage).

A license is required to run SAN Multipath, which is supported as of release 4.0.2. On 3.x releases, a license is required to run Dual HBA (host bus adapter). Other licenses are required to support SAN and NAS, but these are usually provided at the factory as part of an M5000S or M5000N system.

SAN multipath is a failover method in which any failure in the fabric path causes an immediate switch-over to an alternate fabric path, without requiring system reboot. Dual HBA is an older method that employs the alternate path after system reboot.

On 3.x systems, the size of a LUN was limited to 1 TB; larger total storage required multiple LUNs. On 4.x, LUN size and total storage are both limited to 4 TB.

Identifiers for Disks and Arrays

Disks are identified using quadruples, four numbers separated by dots:

controller.channel.target.logicalUnitNumber

where:

- *controller* is the number of the controller—Currently, there is a maximum of two controllers on Mirapoint systems, so this number is always `0` or `1`.
- *channel* is the channel number—Currently, some Mirapoint systems support two backend fibre loops, so this number is either `0` or 1.
- *target* is the disk number (SCSI ID)—Because there is a maximum number of disks per disk enclosure, this number starts at `0` goes on up.
- *logicalUnitNumber* (LUN ID)—This number is always `0` (zero) for non-SAN systems. SAN systems refer to this as the “LUN” and set it to a non-zero value.

For example, the ID of the fourth disk from the right in the disk enclosure on the second controller might be:

```
1.0.3.0
```

Array identifiers have a similar format, but their numbering scheme is based on system internals.

Subcommands

Addarray

Begins initializing the first array of unused RAID disks detected by `Storage Scan`. These disks must be installed as described in the hardware manual for your system. If the disks are installed incorrectly, `Storage Add` returns a `NO` response.

`Storage Addarray` starts initialization and returns a prompt immediately, while initialization proceeds. To monitor initialization progress, use `Storage Arrays`.

Upgrading a 9-drive 980 GB disk shelf to a 15-drive 1.7 TB disk shelf takes about six to seven hours on an unloaded system. Messaging services remain available, but initialization might take twice as long on a loaded system.

After initialization is complete, you must use `Storage Configure` to begin using the new array.

Syntax

tag `Storage Addarray`

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Storage Addarray
* 5 1.0.0.0
5 OK Started
```

Addspare

Adds an unused disk as a hot spare. The new disk must be installed as described in the hardware manual for your system. If the disk is installed incorrectly, `Storage Addspare` returns a `NO` response.

Syntax

tag `Storage Addspare`

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Storage Addspare
* 4 1.0.3.0
4 OK Completed
```

Alarm

Turns off the audible alarm triggered by a failure in the RAID subsystem, such as a disk failure.

For some Mirapoint systems, you cannot use this command to silence all alarms. Instead, you might have to press an alarm silence button somewhere. Refer to your *Hardware Installation and Maintenance* book for specific information.

Syntax

```
tag Storage Alarm
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Storage Alarm
2 OK Completed
```

Arrays

Responds with the status of all arrays. This subcommand is used mostly for monitoring the progress of `Storage Addarray` and `Storage Configure`.

Syntax

```
tag Storage Arrays
```

Response

There is one response line for each array; each line has this format:

```
* tag logicalID type disk-list state status percent
```

where:

- *logicalID* is the logical ID of an array or hot spare.
- *type* is one of:
  - RAID-5—the logical ID identifies a RAID-5 array.
  - RAID-1—the logical ID identifies a RAID-1 mirrored array.
  - RAID-10—the logical ID identifies a RAID-10 mirrored and striped array.
  - Spare—the logical ID identifies a hot spare.
- *disk-list* is a list of the physical IDs of the disks that make up an array; for a hot spare, this is just the physical ID of the disk. This list is preceded by an open parenthesis ('(') and followed by a close parenthesis (')').
- *state* is one of:
  - Optimal—the disk or array is OK (operating normally).
  - Failed—the disk or array has failed and is unusable.
  - Rebuild—the controller is rebuilding the disk or array.
  - Initialize—the array is being initialized by Storage Addarray.
  - Degraded—the array is running in degraded mode.
  - Missing—the disk has been physically removed from its bay.
- *status* is one of:
  - Inuse—the disk being used for data storage.
  - Unused—the disk not currently being used for data storage.
  - Config—the disk is being configured.
- *percent* only appears for disks and arrays that are currently being initialized, configured, or rebuilt; this field indicates the percentage complete for Storage Addarray and Storage Configure.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

For RAID arrays composed of 3x 400GB and 5x 400GB disks.

```
3 Storage Disks
* 0.0.0.0 381469 Inuse Optimal ( ECC no ) 0 0
* 0.0.1.0 381469 Inuse Optimal ( ECC no ) 0 0
* 0.0.2.0 381469 Unused Optimal ( ECC no ) 0 0
* 0.0.3.0 381469 Unused Optimal ( ECC no ) 0 0
* 0.0.4.0 381469 Unused Optimal ( ECC no ) 0 0
3 OK Completed
4 Storage Arrays
* 0.0.0.0 RAID-1 ( 0.0.0.0 0.0.1.0 ) Optimal Inuse
5 OK Completed
5 Storage Space
* 295651 632
5 OK Completed
```

```
5 Storage Disks
* 0.0.0.0 381469 Inuse Optimal ( ECC no ) 0 0
* 0.0.1.0 381469 Inuse Optimal ( ECC no ) 0 0
* 0.0.2.0 381469 Inuse Optimal ( ECC no ) 0 0
* 0.0.3.0 381469 Inuse Optimal ( ECC no ) 0 0
* 0.0.4.0 381469 Unused Optimal ( ECC no ) 0 0
5 OK Completed
6 Storage Arrays
* 0.0.0.0 RAID-10 ( 0.0.0.0 0.0.1.0 0.0.2.0 0.0.3.0 ) Optimal Inuse
6 OK Completed
7 Storage Space
* 614120 636
7 OK Completed
```

Battery

Stops and starts battery recalibration. If the ACM (advanced cooling module) is removed to replace a failed fan, the RAID controller loses battery calibration. Use this command to stop calibration at peak hours and restart it at a better time.

Syntax

`*tag* Storage Battery *action battnum*`

where *action* is either `Startcalibration` or `Stopcalibration` and *battnum* is a battery number, which can be determined with `Storage Get Bbstatus`. Some older Mirapoint systems do not support these battery commands.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
25 Storage Battery Stopcalibration
25 OK Started
```

Configure

Brings newly initialized disks online. Before running this command, you must first run `Storage Addarray` to initialize the new disks.

By design, this command stops and restarts email services on the system, and may reboot the system if storage cannot be unmounted because of any open files, which is likely except on very lightly loaded systems. Reconfiguration can take from five to 15 minutes or more to complete, depending on the size and number of new disks. Messaging services will be unavailable during this time. You can monitor progress using `Storage Arrays`.

After the storage space has been expanded with this command, it cannot be contracted, and the new array cannot be deleted.

Syntax

```
tag Storage Configure logicalID
```

where *logicalID* is the quadruple of the new array returned by `Storage Addarray`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
6 Storage Configure 1.0.0.0
6 OK Started
```

Configsan

Configures the given LUN for Mirapoint SAN storage, given a logical ID quadruple (see Identifiers for Disks and Arrays on page 601). LUN ID must be > 0 and <= 255.

The `Init` flag configures the initial LUN, designated for system and message store. Assuming you have not run `Grow`, the `Expandlun` flag finds and configures any space recently made available on the initial system LUN. Finally, the `Grow` flag configures other LUNs for additional message store.

LUN size must be >= 35 GB for `Init`. `Expandlun` and `Grow` require LUN size to be increased by >= 20 GB each time. On 3.x systems, the number of further `Expandlun` expansions is unlimited, but the maximum LUN size is 1 TB. On 4.x systems, only four further `Expandlun` operations are allowed, but the maximum LUN size is 4 TB. Afterwards with `Grow`, four further increases are allowed.

This command reboots the system in three circumstances: after initial installation (`Init` flag), after a `Reattach` operation (3.x only), and after a `Standby` operation (3.x only). With 3.x releases, but not with 4.x, a recovery CD (same release!) must be in the drive before reboot, to install software on the designated LUN.

LUN security and LUN zoning are critical in SAN environments because multiple systems may have write access to the same physical disk. LUN security is a method to set ACLs on the LUN for each server attached by FibreChannel to the SAN. LUN zoning allows a switch to be partitioned into smaller switch units.

Syntax

```
tag Storage Configsan logicalID flags
```

where:

- *logicalID*—The quadruple uniquely identifying the SAN LUN. The final quadruple (LUN ID) must be greater than zero and less than 256.
- *flags*—Configuration modifiers, one of:

- `Init`—Initializes the first LUN configuration. The LUN must be clear or an error results. This command is synchronous.
- `Initforce`—Similar to `Init` but also clears the LUN. This command could be dangerous because passing the wrong *`logicalID`* may wipe out production data.
- `Detach`—Unsupported in 4.x releases. Detaches an active head from its LUN, unconfiguring a 1+1 standby so it can be reconfigured as N+1.
- `Expandlun`—Allows a system already configured for SAN to expand the storage partition to use any additional space that had been added to the system LUN. Adding space to the system LUN is handled by the SAN storage, not the Mirapoint system. This command is asynchronous.
- `Grow`—Allows a SAN-configured system to grow its storage partition. The LUN must be clear or an error results. This command is asynchronous.
- `Growforce`—Similar to `Grow` but also clears the LUN. This command could be dangerous because passing the wrong *`logicalID`* may wipe out production data. This command will not overwrite any LUN that was previously initialized as Mirapoint storage (*`logicalID`* marked as `system` or `inuse`).
- `Standby`—Unsupported in 4.x releases. Initializes standby system in a Mirapoint failover cluster.
- `Reattach`—If a scratch install is required during a system recovery scenario, use this flag to reattach an existing LUN marked as `masked` or `system`. System recovery reattach is not allowed unless the current server has never been bound to a SAN LUN.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
16 Storage Configsan 4.0.0.4 init
16 OK Completed
```

Checkadd

Checks to see what would happen if you ran the `Storage Addarray` or `Addspare` command. Does not actually modify system storage.

Syntax

```
tag Storage Checkadd
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
23 Storage Checkadd
* 23 array yes "0.0.4.0 from disks 0.0.4.0, 0.0.5.0 and 0.0.6.0"
* 23 spare yes "0.0.2.0 from disk 0.0.2.0"
23 OK Complete
```

Countpaths

Returns the number of paths on the fabric between a Mirapoint appliance and its SAN storage. This is the same as the number of lines that `Listpaths` returns.

Multipath commands are not supported in 3.9.x releases.

Syntax

tag `Storage Countpaths` *pattern*

where *pattern* must be * or "" to indicate all paths.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
25 Storage Countpaths ""
* 25 2
25 OK Completed
```

Countpathstates

Given a path ID, returns the number of related multipaths on the fabric between a Mirapoint appliance and its SAN storage.

Multipath commands are not supported in 3.9.x releases.

Syntax

tag `Storage Countpathstates` *pathID*

where *pathID* is a multipath ID as returned by `Storage Listpaths` or a logical ID as returned by `Storage Listsan`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
27 Storage CountPathStates 2.0.0.1
* 27 4
27 OK Completed
```

Countsan

Returns the number of SAN (storage area network) devices found.

Syntax

```
tag Storage Countsan pattern
```

where *pattern* must be * (asterisk) or "" (null string).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
17 Storage Countsan *
* 17 4
17 OK Completed
```

Countsanport

Unsupported in 4.x releases. Returns the number of SAN devices on a specific port.

Syntax

```
tag Storage Countsanport portID pattern
```

where *portID* can be `Port0` or `Port1`, and *pattern* is a string matching SAN device traits, or * or "" to indicate all.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
23 Storage Countsanport port0 "" "" ""
* 23 NO Not supported
23 OK Completed
```

Delete

Deletes an array or hot spare, making it unavailable for use. You should use this command *only* for:

- Replacing a hot spare with higher-capacity disk.
- Removing an array that failed to initialize properly.

An array that has been put into use with `Storage Configure` cannot be deleted.

To replace a hot spare:

1. Use `Storage Delete` to delete the spare.
2. Physically replace the spare with a higher-capacity disk.
3. Use `Storage Addspare` to begin using the new spare.

If a disk fails while the array to which it belongs is being initialized:

1. Use `Storage Delete` to delete the array.
2. Physically replace the failed disk with a good disk of the same capacity.
3. Initialize the array again (see “Adding RAID Disks” in the administration client online help).

Privilege Levels

Administrator

Domain Sensitivity

None

Syntax

tag `Storage Delete` *logicalID*

where *logicalID* is the quadruple of the array you want to delete, as shown in the response from `Storage Arrays`.

Example

```
7 Storage Delete 1.0.3.0
7 OK Completed
```

Disks

Responds with a list of all disks in the RAID system, giving the logical ID, size in kilobytes (number of 1 kilobyte blocks), status, and state for each disk.

Syntax

tag `Storage Disks`

Response

`Storage Disks` generates a response line for each disk, which has this format:

`*` *`tag disk-id size status state`* `( ECC` *`ecc-status`* `)` *`diag-codes`*

where:

- *`disk-id`* is the disk identifier.
- *`size`* is the capacity of the disk in megabytes.
- *`status`* is one of:
  - `Inuse`—the disk is part of an array.
  - `Unused`—the disk not currently being used in an array or as a hot spare.
  - `Spare`—the disk is a hot spare.
- *`state`* is one of:
  - `Optimal`—the disk is OK (operating normally).
  - `Failed`—the disk has failed and is unusable.
  - `Rebuild`—the controller is rebuilding data from a lost disk on the disk.
  - `Initialize`—the disk is being initialized by `Storage Addarray`.
  - `Missing`—the disk has been physically removed from its bay.
- *`ecc-status`* is one of:
  - `yes`—the disk is error-correcting code (ECC) formatted.
  - `no`—the disk is not ECC formatted.
- *`diag-codes`* can provide Mirapoint technical support with information to help diagnose storage problems.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
8 Storage Disks
* 8 0.0.0.0 8460 Inuse Optimal ( ECC yes ) 0.c2.0.0 0.a2.0.0
* 8 0.0.1.0 8460 Inuse Optimal ( ECC yes ) 0.c2.0.0 0.a2.0.0
* 8 0.0.2.0 8460 Inuse Optimal ( ECC yes ) 0.c2.0.0 0.a2.0.0
8 OK Completed
```

Get

Responds with the value of the specified parameter.

Syntax

tag `Storage Get` *parameter*

where *parameter* is one of:

- `Bbstatus`—Shows the battery status; same as `RAID.BATTERYSTATUS` from the `Stat Get` command but without a delay. If the ACM (advanced cooling module) is removed to replace a failed fan, the RAID controller loses battery calibration. You can use the `Storage Battery` command to stop and restart recalibration. Status goes Recondition Needed, Recondition Discharging, Charging, Charged.
- `Bbwritethru`—Whether the RAID controller will switch from cacheing (the usual behavior) to **write-through mode** (`On` or `Off`). See `Storage Set`.
- `Bbwritethruthresh`—A battery charge in minutes of running time remaining. If `Storage Set Bbwritethru` is `On`, the RAID controller switches to write-through mode when the battery charge drops below this threshold.
- `Controller`—(Available in MOS 4 only) Displays RAID controller model information in this format: <Controller id> <Controller name> `F`<Firmware version>`<(`Firmware Build Number`)>` `B`<Bios Version>`<(`Bios Build Number`)>`
- `Diskvendor`—Shows the disk ID, vendor, model name, and firmware level of disks in the storage subsystem. This is useful for diagnosing system problems.
- `Idecache`—Whether the IDE disk write cache is enabled `On` or disabled `Off`. Enabling the IDE cache improves performance but reduces reliability, because if the system crashes during a disk write, contents of the cache will be lost.
- `Portwwn`—Returns the world wide port name (WWPN) for the HBA (host bus adapter) card. This value may be used as a filter for SAN vendors who offer functionality called LUN Security and Masking, which filters out FibreChannel packets at each SAN port based on the WWPN of a particular server. WWPN is used for low-level protocols, whereas world wide node name (WWNN) is used for high-level protocols.
- `Rebuildrate`—Speed at which a RAID disk gets rebuilt after coming online, if hardware supports this setting. See `Storage Set`.
- `Storetype`—Returns the type of message storage, either `local` for connected disk drives, `nas` for network attached storage (NFS remote mount), or `san` for storage area network (SAN). Cannot be altered with `Storage Set`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
9 Storage Get Bbwritethruthresh
* 9 4320
9 OK Completed
```

```
29 Storage Get Controller
* 29 0 3405 F5.2-0(12814) B5.2-0(12814)
29 OK Completed
```

```
19 Storage Get Portwwn
* 19 0x210000e08b056bf2
19 OK Completed
```

Getnas

Displays the configuration of network attached storage (NAS). Use this command to get the server name and path information associated with the current mail store. A NAS license is required.

Syntax

```
tag Storage Getnas
```

If network attached storage is in use as a mail store, the server name and exported path name are displayed.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
14 Storage Getnas
* 14 10.0.0.2:/vol/vol0/mail.storage
14 OK Completed
```

Getport

Responds with the value of the specified port parameter.

Syntax

```
tag Storage Getport portID parameter
```

where *portID* may be `Port0` or `Port1` and *parameter* is as follows:

- Portwwn—Returns the world wide port name (WWPN) for the port. This command run on Port0 should return the same value as Storage Get Portwwn.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
20 Storage Getport Port1 Portwwn
* 20 0x210000e08b018484
20 OK Completed
```

Listpaths

Displays the multiple paths on the SAN fabric between a Mirapoint appliance and its storage. Five returned fields show information about the LUNs and path status.

Multipath commands are not supported in 3.9.x releases.

Syntax

```
tag Storage Listpaths pattern start count
* tag logicalID multipathMode physicalDevices activePath multipathStatus
```

where *pattern* must be * or "" to indicate all paths, and *start* and *count* specify beginning item and duration.

The fields returned are as follows:

- *logicalID*—Same as what Storage Listsan returns for Logical ID.
- *multipathMode*—Currently always Failover.
- *physicalDevices*—LUN numbers in the multipath storage device.
- *activePath*—LUN number of storage device on the currently active path.
- *multipathStatus*—One of the following:
    - Optimal— All paths are active (on systems with 2 HBA ports).
    - Partial—All path are active (on systems with 1 HBA port).
    - Degraded—One path has failed (either 1 or 2 HBA ports).
    - Failed—All paths have failed (either 1 or 2 HBA ports).

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
21 Storage ListPaths "" "" ""
* 21 2.0.0.1 Failover "2.0.0.1 2.0.1.1 3.0.0.1 3.0.1.1" "3.0.0.1" "Optimal"
* 21 2.0.0.2 Failover "2.0.0.2 2.0.1.2 3.0.0.2 3.0.1.2" "3.0.0.2" "Optimal"
21 OK Completed
```

```
22 Storage ListPaths "" "" ""
* 22 2.0.0.1 Failover "2.0.0.1 2.0.1.1 3.0.0.1 3.0.1.1" "3.0.1.1" "Degraded"
22 OK Completed
```

```
23 Storage ListPaths "" "" ""
* 23 2.0.0.1 Failover "2.0.0.1 2.0.1.1 3.0.0.1 3.0.1.1" "" "Failed"
23 OK Completed
```

```
24 Storage ListPaths "" "" ""
* 24 2.0.0.1 Failover "2.0.0.1 2.0.1.1" "2.0.1.1" "Partial"
24 OK Completed
```

Listpathstates

Given a path ID, shows the state of related multipaths on the SAN fabric between a Mirapoint appliance and its storage. This helps diagnose what paths have failed.

Multipath commands are not supported in 3.9.x releases.

Syntax

```
tag Storage Listpathstates pathID start count
* tag multipathID status
```

where *pathID* is a logical ID as returned by `Storage Listpaths` (for multipath) or `Storage Listsan`, and *start* and *count* specify beginning item and duration.

The fields returned are as follows:

- *multipathID*—Each LUN that `Listpaths` returns for *physicalDevices*.
- *status*—One of the following:
  - `Optimal`—Path is fully operational.
  - `Failed`—Path has failed.
  - `Undefined`—Unknown state, usually due to unsupported configurations or SAN devices that probably require specific patches to work correctly.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
26 Storage ListPathStates 2.0.0.1 "" ""
* 26 2.0.0.1 "Optimal"
* 26 2.0.1.1 "Failed"
* 26 3.0.0.1 "Optimal"
* 26 3.0.1.1 "Optimal"
26 OK Completed
```

Listsan

Shows all SAN devices that have been found, with columns showing information about these devices. LUN IDs > 511 are ignored, and only one of the first 32 LUNs may be used in a `Configsan` command.

Syntax

```
tag Storage Listsan pattern start count
* tag logicalID vendorName size lunState info hostname
```

where *pattern* is a string matching SAN device traits (or * or "" to indicate all) and *start* and *count* specify beginning item and duration.

In the response line:

- *logicalID*—Quadruple (A.B.C.D) uniquely identifying the SAN LUN.
- *vendorName*—A string identifying the SAN vendor.
- *size*—The formatted storage size in megabytes.
- *lunState*—Describes LUN condition as one of the following:
  - `bound`—LUN is currently configured and bound to this server.
  - `masked`—LUN is unbound; may or may not have Mirapoint disk label.
- If *lunState* is `bound`, *info* designates one of the following:
  - `system`—First LUN bound to the local server.
  - `inuse`—Additional LUN bound to local server.
  - `unused`—LUN was being configured when something interrupted the process. To restart configuration, run `Storage Configsan` again.
- If *lunState* is `masked`, *info* designates one of the following:
  - `system`—LUN was the first device bound to a foreign server.
  - `inuse`—Additional LUN bound to a foreign server. Can also indicate LUN state for an initial install that failed midway.
  - `clear`—LUN has the first 16KB zeroed and is eligible for configuration.
  - `unknown`—LUN contains unknown data and cannot be configured into the system until the first 16KB segment is zeroed out.
  - `unsupported`—LUN's vendor, product, or revision level is not supported.
  - `error`—LUN reports read errors.
- *hostname*—Host name of the Mirapoint server that last owned this LUN. Column is truncated to 20 characters.

In system recovery after the current server is reinstalled, all previously used LUNs will appear as masked because the current server has not yet been bound to any.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

With SAN multipath in 4.x releases, only one LUN path appears (second example).

```
18 Storage Listsan * "" ""
* 18 2.0.0.1 HP 30000MB bound system san.mirapoint.com
* 18 2.0.0.2 HP 20000MB bound inuse san.mirapoint.com
18 OK Completed
```

```
28 Storage Listsan * "" ""
* 28 2.0.0.1 HP 35000 MB bound system san.mirapoint.com
28 OK Completed
```

Listsanport

Unsupported in 4.x releases. Returns a list of SAN devices on a specific port.

Syntax

```
tag Storage Listsanport portID pattern start count
```

where *portID* can be `Port0` or `Port1`, and *pattern* is a string matching SAN device traits, or * or "" to indicate all, while *start* and *count* specify beginning item and duration.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
22 Storage Listsanport port0 "" "" ""
* 22 NO Not supported
22 OK Completed
```

Scan

Scans the RAID system for changes in the hardware configuration, such as the insertion of new disks.

This command is deprecated. You must reboot the system to add a new SCSI device. Please refer to the hardware manual for your system for detailed instructions.

Frequent use of this command can degrade performance. It's best to use this subcommand only after installing new disks.

Syntax

```
tag Storage Scan
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
11 Storage Scan
11 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

```
tag Storage Set parameter value
```

where:

- *parameter* is one of:
  - `Bbwritethru`—Whether the RAID controller will switch from cacheing (the usual behavior) to **write-through mode** (`On` or `Off`) when the RAID controller battery charge drops below a certain threshold. In write-through mode, the controller does not store any data in its cache, but instead writes data to disk synchronously. Write-through mode ensures file system data integrity with some cost in performance.
  - `Bbwritethruthresh`—A battery charge specified as the minutes of running time remaining. If `Storage Set Bbwritethru` is `On`, the RAID controller switches to write-through mode when the battery charge drops below this `Bbwritethru` threshold. The default value on current hardware is 2880 minutes, but this could vary on new models. Setting this threshold to less than the default value is not recommended because it increases the risk of data corruption in the event of power failure.
  - `Idecache`—If present, turns the IDE disk write cache `On` or `Off`. If IDE exists on the system, the default is `On`. After enabling or disabling `Idecache` in 3.x releases, you must reboot the system before the change takes effect. Reboot is not necessary in 4.x releases: the change takes effect immediately! The IDE cache improves performance but reduces reliability, because if the

system crashes during a disk write, contents of the cache are lost. As part of Setup Wizard, you are asked to choose between "fast" and "safe" after you have selected a routing method. DirectPath always uses safe mode because it is fast enough, but for other routing methods you must choose. Changing the `Idecache` setting generates an `ALERT` message in the system log.
    - `Rebuildrate`—Speed at which RAID disk is rebuilt after coming online. This command succeeds only if RAID storage supports the functionality. May be set `High`, `Medium`, or `Low`. Low rate incurs less system overhead, but then storage remains degraded for a longer period. High rate is default and recommended for the X-45 disk shelf. The X-55 disk shelf does not support this setting, but rebuilds comparatively quickly.
- *`value`* is the value you want to assign to *`parameter`*.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
10 Storage Set BBwritethru On
10 OK Completed

20 Storage Set Idecache On
* 20 WARNING Setting will not take effect until after reboot
20 OK Completed
```

Space

Responds with the total size in megabytes of the RAID storage on your Mirapoint system followed by the amount currently being used in megabytes.

Syntax

```
tag Storage Space
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

In this example, the `space` subcommand shows that the system has a total of 13045 megabytes of storage, 8 megabytes of which is used.

```
**12 Storage Space**
* 12 13045 8
12 OK Completed
```

The Trustedadmin Command

The `Trustedadmin` command specifies Internet Protocol (IP) addresses from which the administration service accepts logins. Until you use the `Trustedadmin Add` command to specify trusted IP addresses, administration service accepts logins from any IP address.

For example, if you add a single address, such as `10.1.0.1`, as a trusted IP address, anyone trying to log into the Administration Suite, command-line interface, or the administration protocol from any other address is denied access. Connections to other services allowed, but only ordinary-user privileges are granted. Moreover for security reasons, administrators connecting from a non-trusted IP cannot change an administrator password. However ordinary users can change their passwords.

Domain administrators and Helpdesk roles are not affected by `Trustedadmin`.

Network Specifiers

In addition to designating individual IP addresses as trusted, you can specify an entire network as trusted using a **network specifier**, a string of the form:

`dotted-quad/mask-bits`

where *`dotted-quad`* is an IP address in dotted-quad notation, such as `10.0.0.0`, and *`mask-bits`* is the number of bits in the network mask to be used in comparing addresses. For example, if *`mask-bits`* is `8`, the network mask is `255.0.0.0`.

The IP address of a user's machine is masked with a network specifier's mask (using a bitwise AND); the result is compared with the network specifier's dotted quad. Only users whose IP addresses match one of the trusted IP addresses or network specifiers are allowed to log in with administration privileges. A user whose address doesn't match can only administer his or her own user account.

Subcommands

Add

Adds an IP address, or range of addresses, to the list of trusted IP addresses. If you try to login to the administration service from an IP address not on the trusted list, access is denied. If you connect using another method, for example WebMail, you are not allowed to change the administrator password.

Specifying a netmask says to trust a range of IP addresses. Not specifying a netmask is the same as netmask `/32`.

It is possible to "lock yourself out" of the administration interfaces by specifying an incorrect trusted address. If you do this, you must type the '**`c`**' command on the serial console to access the administration command-line interface and repeatedly use the `Trustedadmin Delete` command to clear your trusted address list.

Syntax

```
tag Trustedadmin Add address
```

where *`address`* is the IP address or network specifier you want to add to the list of trusted IP addresses.

Privilege Levels

Administrator (does not affect Domain administrator or Helpdesk)

Domain Sensitivity

None

Example

```
2 Trustedadmin Add 192.168.0.0/24
2 OK Completed
```

Count

Responds with the number of IP addresses and networks in the list of trusted IP addresses and networks.

Syntax

```
tag Trustedadmin Count pattern
```

where *`pattern`* is currently ignored.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
3 Trustedadmin Count ""
* 3 2
3 OK Completed
```

Delete

Deletes an IP address or network specifier from the list of trusted IP addresses.

Syntax

```
tag Trustedadmin Delete address
```

where *address* is the IP address or network specifier you want to delete.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Trustedadmin Delete 192.168.0.0/24
4 OK Completed
```

List

Responds with a list of all trusted IP addresses and networks.

Syntax

```
tag Trustedadmin List pattern start count
```

where:

- *pattern* is currently ignored.
- *start* is the first position in the list of trusted IP addresses that you want to see. The empty string ("") implicitly means `0`.
- *count* is number of trusted IP addresses that you want to see. If *count* is greater than the total number of trusted addresses, `list` returns as many trusted addresses as possible. The empty string ("") implicitly means all trusted addresses.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
**5 Trustedadmin List** "" "" ""
* 5 192.168.0.0/24
* 5 10.0.0.0/8
5 OK Completed
```

65

The Trustedhost Command

The `Trustedhost` command is a method of establishing trust between systems. It restricts who is allowed to make queries to the MailHurdle server for verification by the `Mtaverify` command. `Trustedhost` also provides keying for the SMTP service.

Subcommands

Add

Establishes trust relationships between machines or networks.

Syntax

```
tag Trustedhost Add type host-spec arg
```

where:

- *`type`* is one of the following to indicate type of trust relationship:
  - `Authgroup` means to trust the specified host for NTLM authentication. After *`host-spec`*, *`arg`* should be a public key in PEM format. No check is made to ensure that the key belongs to the host for which it is being added.
  - `Loginbeforesmtp` means that this query server participates in an agreement with the message server named by *`host-spec`*; *`arg`* should be the null string (""). For details see `Smtp Set Loginbeforesmtp` on page 535.
  - `Mtagroup` means to trust the specified host to properly convey the aggregate message state in `X-XMS` headers after presentation of the appropriate key. After *`host-spec`*, *`arg`* should be either a public key in PEM format, or a URL from where the key can be fetched. A URL of just "`http:`" is turned into the URL on the host being added. To disable `X-XMS` checking, run `Trustedhost Delete` for the `Mtagroup` relationship. See help about the `Key` command for more information.
  - `Mtaverify` means to trust this host as a MailHurdle client (not a server) for the `Mtaverify` command. After *`host-spec`*, *`arg`* should be null string ("").
  - `Popgroup` means to trust the specified host for APOP (authenticated POP). After *`host-spec`*, *`arg`* should be a public key in PEM format. No check is made to ensure that the key belongs to the host for which it is being added.
- *`host-spec`* is the host name or IP address added to the trusted list.
- *`arg`* modifies the trust relationship and depends on *`type`* (see above).

Privilege Levels

Administrator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
2 Mtaverify Add localhost
2 OK Completed
```

```
3 Trustedhost Add Mtaverify imr2.example.com ""
3 OK Completed
```

```
4 Trustedhost Add Mtagroup av1.example.com {19+}
[PEM stream here]
4 OK Completed
```

```
5 Key New Mta "" "" ""
5 OK Completed
6 Trustedhost Add Mtagroup hostname http:
6 OK Completed
```

Count

Counts the number of trust relationships between machines or networks.

Syntax

```
tag Trustedhost Count type pattern
```

where:

- *type* is one of those specified under `Trustedhost Add`.
- *pattern* must be "" or *, both of which match everything.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
5 Trustedhost Count Mtaverify ""
* 5 1
5 OK Completed
```

Counttypes

Counts the types available for use with the `Trustedhost Add` command.

Syntax

tag `Trustedhost Counttypes` *pattern*

where *pattern* must be "" or *, both of which match everything.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
7 Trustedhost Counttypes ""
* 7 2
7 OK Completed
```

Delete

Removes trust relationships between machines or networks.

Syntax

tag `Trustedhost Delete` *type host-spec*

where:

- *type* is one of those specified under `Trustedhost Add`.
- *host-spec* is the host name or IP address deleted from the trusted list.

Privilege Levels

Administrator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
6 Trustedhost Delete Mtaverify imr2.example.com
6 OK Completed
```

List

Displays the trust relationships between machines or networks.

Syntax

tag `Trustedhost List` *type pattern start count*

where:

- *type* is one of those specified under `Trustedhost Add`.
- *pattern* must be "" or *, both of which match everything.
- *start* is the number of the first item, and *count* is the number of items to list.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
5 Trustedhost List Mtaverify "" "" ""
* 5 imr1.my.domain ""
5 OK Completed
```

Listtypes

Lists the types available for use with the `Trustedhost Add` command.

Syntax

tag `Trustedhost Listtypes` *pattern start count*

where:

- *pattern* must be "" or *, both of which match everything.
- *start* is the number of the first item, and *count* is the number of items to list.

Privilege Levels

- Administrator
- Helpdesk administrator
- Backup operator

Domain Sensitivity

Allowed only when no delegated domain is current.

Example

```
6 Trustedhost Listtypes "" "" ""
* 6 mtaverify
* 6 mtagroup
6 OK Completed
```

66

The Uce Command

The Uce command manages the various available Antispam facilities, and lets you block messages from particular IP networks or DNS domains. These features help minimize the amount of unsolicited commercial email (UCE) that your systems and local-area network receive.

Antispam

Antispam scanning is enabled by license. Prepackaged spam identification rules on the system are updated with `Uce Update`. You can disable antispam scanning with `Conf Disable`. You can create various types of whitelists and a blacklist with the `Uce Addexception` command. Whitelists specify senders who can always deliver messages, even if content is identified as spam. Blacklists specify senders who can never deliver messages to you, no matter what. Whitelists override blacklists.

Principal Edition Antispam requires a license, often factory-installed on RazorGate. Signature Edition Rapid Antispam requires a different license, usually higher cost. Junk Mail Manager requires yet another license. Applying either Antispam license automatically `Conf`-enables inbound scanning. Antispam scanning can be provided for specific users only by means of the `Cos` command.

After applying either Antispam license, or after updating system software, run the `Uce Update` command to obtain the latest pattern files. See Update on page 645.

By default only “local” messages are scanned. Antispam scanning functions on a router if the recipients appear to be local (that is, if a recipient’s address is in a mail domain or delegated domain). You can enable “nonlocal” outbound scanning with the `Conf Enable Antispam-all` command.

If COS is not enabled, Antispam works for all users. With COS enabled, Antispam works for users with `antispam` listed in their `miService` LDAP attribute.

After licensing, all users acquire the “System Junk Mail Rule” that can be applied to messages labelled as junk mail by the Antispam facility. For details, see Filter Rules on page 240 concerning the `:UCE` attribute.

A header line `X-Junkmail: UCE(`*`score`*`)` is added to all messages identified as spam. The score is a number between 1 and 300. Messages are identified as junk mail if the score is above (default) 50; this threshold can be changed. Messages that match the whitelist lose this header and get a `X-Junkmail-Whitelist: Yes` line instead. `WhitelistIp` is treated similarly with header `X-Junkmail-IP-Whitelist`, and `WhitelistTo` is treated similarly with `X-Junkmail-Recipient-Whitelist`. The words (`by domain whitelist`) indicate domain-level whitelisting, while the words (`by user at` *`hostname`*) indicate personal whitelisting. Messages that match the blacklist get the `X-Junkmail: Blacklist` header line added.

Signature Edition Rapid Antispam adds the `X-Junkmail-SD-Raw` header showing reference ID and sender IP address.

In case of whitelisting, the `X-old-subject` header is consulted and used to restore the subject line as needed. If a message is whitelisted at either user or domain level, spam-related headers other than `X-Junkmail-Status` are stripped from a message.

In case of blacklisting, if LDAP attribute `miSpamInSubject` is set, "Spam" subject prefixing is performed; see `Uce Setoption Spamprolog` for details. In other words, blacklisting is handled very much like spam.

If `Smtp Set Ucesuspectlist` is On, Antispam scanning auto-manages a suspect list by adding the IP address of any message categorized as spam, if not there already. Also set `Mtaverify Set Checksuspectlist` On to enable MailHurdle consideration of the suspect list.

Rapid Antispam

Rapid Antispam requires knowledge of the remote IP address, which is not always the machine that hands off the message. It uses the "Received" headers to handle this as much as possible. Any address that is on a network associated with a local Ethernet interface is considered "local" and ignored, as are RFC private networks (10 net, 192.168, and 172.16.net).

Additionally, if an IP address or network is in the relay list, it is considered safe and is also ignored. This is done on the theory that if we trust hosts to relay, they should not be sending spam. This feature can help tune detection of remote IP addresses. The calculated IP address is available in the `X-Junkmail-SD-Raw` header. If this is a local or trusted machine, its address should be added to the relay list. Repeat this process until everything works correctly.

Category and intensity in the `X-Junkmail-SD-Raw` header are mapped into a score inserted into the `X-Junkmail-Status` header as follows: 0 not spam, 10 unknown, 35-45 suspect by intensity, 50-60 bulk by intensity, 300 definitely spam.

Note: It is not always possible to determine the IP address from the received header, because SMTP relays do not always pass along this information.

Networks

To specify a network from which to block all email messages, use a partial IP address. For example, if you specify `10.128`, the SMTP service will block all email from `10.128.0.1` and `10.128.3.1`, but will accept mail from `10.129.0.1`.

Subcommands

Add

Blocks email from the specified IP network or DNS domain. This is called reject list in the GUI. Even addresses in the relay list can be blocked. `Uce Add` does not require the Antispam license.

Syntax

```
tag Uce Add block
```

where *block* is the IP network or DNS domain from which the system will block all email.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Uce Add example.com
2 OK Completed
```

Addexception

Adds an address to a whitelist of any type, blacklist, or suspect list. The address may be either a domain name or a fully qualified email address, controlling scope. Use of whitelists is important to reduce false positives and to prevent internal company mail from being classified as spam. The size limit of each exception list is 1 MB of aggregate data. When you try to add an entry that would exceed this limit, an error results and the exception list remains unchanged.

As of release 3.5.5 some whitelist processing could occur before Antispam scanning; see type `(list=Whitelist)(type=MH)` below.

Syntax

```
tag Uce Addexception scope type address
```

where:

- *scope* indicates the user or domain to which this whitelist or blacklist exception applies. User and domain cannot both appear. Administrators must switch to a delegated domain with Domain Setcurrent before altering an exception list for a user in a delegated domain. Scope can take one of the following three forms:

- *username*—The user who owns this whitelist or blacklist.
- (domain=*dname*)—Specifies the domain where the whitelist or blacklist applies. A blank *dname* or keyword primary refers to the top-level domain. The pseudo-domains Local, Nonlocal, and Any are supported; for more information see the Filter command Add on page 244.
- (user=*username*)—Same as *username*, assumed to be the top-level domain user unless a delegated domain is current.

- *type* is a keyword indicating list type, listed below in order of precedence:
  - Whitelist—Messages from this address (an entire domain or one specific user) are always delivered. Use the at-sign (@) to specify one user@domain, or omit the user and specify just @domain for an entire domain. Whitelist is called "Allowed Senders" in the GUI.
  - WhitelistIp—Messages from this fully specified IP address are always delivered. Wildcards and subnet masks are not allowed.
  - WhitelistTo—Mail to this user or distribution list is always delivered. This is a recipient whitelist. Called "Allowed Mailing Lists" in the GUI. One whitelisted recipient causes everyone on the To/Cc list to be whitelisted also (this behavior may change in the future).
  - Blacklist—Messages from this user or domain are never delivered. Called "Blocked Senders" in the GUI.
  - Suspectlist—Messages from the specified IP address are added to the list of suspect sites that are exclusively subjected to MailHurdle processing. *Scope* must be (domain=any). The suspect list is automatically generated during Antispam scanning, if Smtp Set Ucesuspectlist is On, but you can also modify the list. MailHurdle consideration of the suspect list is turned On and Off by the Mtaverify Set Checksuspectlist command.
  - (list=Whitelist)(type=MH)—Apply the exception list, currently only Whitelist, during MailHurdle processing, but not during Antispam scanning. Scope must be (domain=any).

- *address* is one of the following to match an address in the **From** header field, or for Suspectlist, a numeric IP address:
  - user@domain—A normal email address.
  - @domain—Any address that is within this domain name.
  - user@*domain—Mild form of wildcarding that matches all subdomains. Note: user wildcarding is not supported: left of @, asterisk (*) is a literal.
  - @*domain—Any address that is within this domain or its subdomains.
  - @*—All addresses.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (for their own mailbox)

Domain Sensitivity

Administrator exceptions apply to the top-level domain, domain administrator exceptions apply to delegated domains, and users' exceptions apply to their inbox. The `Suspectlist` is not domain-sensitive.

Example

```
6 Uce Addexception "(domain=primary)" Whitelist @example.com
6 OK Completed
```

```
7 Uce Addexception juser Whitelist jokemail@elists.org
7 OK Completed
```

Count

Responds with the number of blocked IP networks and DNS domains.

Syntax

tag `Uce Count` *pattern*

where *pattern* is currently ignored.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
3 Uce Count ""
* 3 1
3 OK Completed
```

Countexception

Show number of addresses in a whitelist of any type, blacklist, or suspect list.

Syntax

tag `Uce Countexception` *scope type address*

where:

- *`scope`* indicates the user or domain to which this command applies. It can take one of the following forms; the null string ("") indicates primary domain.
  - *`username`*—The user who owns this whitelist or blacklist.
  - `(domain=`*`dname`*`)`—Specifies the domain where the whitelist or blacklist applies. A blank *`dname`* or keyword `primary` refers to the top-level domain.
  - `(user=`*`username`*`)`—Same as username.
- *`type`* is one of those listed under `Uce Addexception`.
- *`address`* may be as listed under `Uce Addexception`, or star (`*`) or null string ("") to match all addresses.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (for their own mailbox)

Domain Sensitivity

Administrator exceptions apply to the top-level domain, domain administrator exceptions apply to delegated domains, and users' exceptions apply to their inbox. The `Suspectlist` is not domain-sensitive.

Example

```
8 Uce Countexception "" Whitelist *
8 * 1
8 OK Completed
```

Delete

Unblocks email from the specified IP network or DNS domain.

Syntax

tag `Uce Delete` *block*

where *`block`* is the IP network or DNS domain from which you no longer want to block all email.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
5 Uce Delete example.com
5 OK Completed
```

Deleteexception

Delete the specified address from a whitelist of any type, blacklist, or suspect list.

Syntax

`tag Uce Deleteexception scope type address`

where:

- `scope` indicates the user or domain to which this command applies. It can take one of the following forms; the null string ("") indicates primary domain.
  - `username`—The user who owns this whitelist or blacklist.
  - `(domain=dname)`—Specifies the domain where the whitelist or blacklist applies. A blank `dname` or keyword `primary` refers to the top-level domain.
  - `(user=username)`—Same as username.
- `type` is one of those listed under `Uce Addexception`.
- `address` may be as listed under `Uce Addexception`, or in some cases star (*) or null string ("") to match all addresses.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (for their own mailbox)

Domain Sensitivity

Administrator exceptions apply to the top-level domain, domain administrator exceptions apply to delegated domains, and users' exceptions apply to their inbox. The `Suspectlist` is not domain-sensitive.

Example

```
10 Uce Deleteexception juser Whitelist jokemail@elists.org
10 OK Completed
```

Exportexceptions

Lists all exceptions of all types for the given scope. White and black lists are shown one per line with type and address, in UTF-8 representation including spaces.

Syntax

`tag Uce Exportexceptions scope`

where *`scope`* indicates the user or domain to which this command applies. It can take one of the three following forms:

- *`username`*—The user who is whitelisted or blacklisted.
- `(domain=`*`dname`*`)`—Specifies the domain that is whitelisted or blacklisted. A blank domain or keyword `primary` refers to the top-level domain.
- `(user=`*`username`*`)`—Same as username.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator (for the delegated domain)

Domain Sensitivity

Administrators can export all exceptions. Domain administrator can export only the exceptions for their delegated domain.

Example

```
13 Uce Exportexceptions "(domain=primary)"
* 13 {24}
whitelist,@example.com

13 OK Completed
```

Getoption

Retrieve setting of an antispam feature.

Syntax

`tag Uce Getoption option scope`

where *`option`* may be one of the following, and for all options *`scope`* must currently be `(domain=local)`. See `Uce Setoption` for details.

- `Headerinfo`—Whether to produce `X-junkmail-info` header line.
- `Multienginebulkonly`—Method of optimizing dual Antispam scanning.
- `Reporting`—Whether to send junk mail samples to Mirapoint.
- `Spamprolog`—Whether the spam prolog is turned on or off for junk mail.
- `Spamprologtext`—Text of spam prolog inserted into the subject line.
- `SuspectlistThreshold`—Spam scoring threshold, MailHurdle suspect list.
- `Threshold`—Spam scoring threshold, for deciding what is junk mail.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
18 Uce Getoption Spamprolog "(domain=local)"
* 18 On
18 OK Completed
```

```
19 Uce Getoption Spamprologtext "(domain=local)"
* 19 "=== spam? ==="
19 OK Completed
```

```
20 Uce Getoption Threshold "(domain=local)"
* 20 50
20 OK Completed
```

Importexceptions

Sets all exceptions of all types for the given scope. White and black lists are shown one per line with type and address, in UTF-8 representation including spaces.

Takes data in `Exportexceptions` format and adds it to the user or domain specified by scope. This overwrites any existing data, so it may lead to loss of data. All input is rejected if there is a format error in any line. The size limit of each exception list is 1 MB of aggregate data. Attempting to add entries that exceed this limit results in an error, and the exception list remains unchanged.

Syntax

`tag Uce Importexceptions scope {bytes+}`

where *scope* is specified under `Uce Exportexceptions` and *bytes* is the number of characters in the literal string input on the following lines.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator (for the delegated domain)

Domain Sensitivity

Administrators can import all exceptions. Domain administrator can import only the exceptions for their delegated domain.

Example

```
14 Uce Exportexceptions "(domain=primary)" {24+}
```

```
whitelist,@example.com
14 OK Completed
```

Junkmailsummaries

Starts, stops, and checks status of spam-report messages from Junk Mail Manager, which must be `Conf` enabled. You can control content of the spam-report messages with the `Message` command facility `MESSAGE.JUNKMAIL.SUMMARY`.

Syntax

tag `Uce Junkmailsummaries` *action*

where *action* is one of the following:

- `Start`—Initiates run sending junk mail summaries to all JMM users. Usually controlled from the `Schedule` command.
- `Status`—Returns status of the current run, including:
  - `State`—Is summarization currently running or stopped.
  - `Successes`—Number of summaries sent successfully.
  - `Failures`—Number of summaries attempted that failed to send.
  - `Totalprocessed`—Total number of users tried (successes + failures).
  - `Totalneeded`—Total number of users that should be processed.
- `Stop`—Halts a junkmail-summary run in progress. If you stop a run in the middle and later restart, users may get redundant summary messages.

Privilege Levels

Administrator

Domain Sensitivity

Returns error if run in a delegated domain.

Example

```
2 Uce Junkmailsummaries Status
* 2 state running
* 2 successes 25
* 2 failures 3
* 2 totalprocessed 28
* 2 totalneeded 1249
2 OK Completed
```

List

Responds with a list of all IP networks and DNS domains from which email is currently blocked.

Syntax

`tag Uce List pattern start count`

where:

- *`pattern`* is currently ignored.
- *`start`* is the first position in the list of networks and domains that you want to see. The empty string (`""`) implicitly means `0`.
- *`count`* is number of networks and domains that you want to see. The empty string (`""`) implicitly means all networks and domains. If *`count`* is greater than the total number of networks and domains, `list` returns as many networks and domains as possible.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
4 Uce List "" "" ""
* 4 example.com
4 OK Completed
```

Listexception

Display the addresses in a whitelist of any type, blacklist, or suspect list.

Syntax

`tag Uce Listexception scope type address start count`

where:

- *`scope`* indicates the user or domain to which this command applies. It can take one of the following forms; the null string ("") indicates primary domain.
  - *`username`*—The user who owns this whitelist or blacklist.
  - `(domain=`*`dname`*`)`—Specifies the domain where the whitelist or blacklist applies. A blank *`dname`* or keyword `primary` refers to the top-level domain.
  - `(user=`*`username`*`)`—Same as username.
- *`type`* is one of those listed under `Uce Addexception`.
- *`address`* may be as listed under `Uce Addexception`, or star (`*`) or null string ("") to match all addresses.
- *`start`* and *`count`* specify the beginning and ending items to list.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Users (for their own mailbox)

Domain Sensitivity

Administrator exceptions apply to the top-level domain, domain administrator exceptions apply to delegated domains, and users' exceptions apply to their inbox. The `Suspectlist` is not domain-sensitive.

Example

```
8 Uce Listexception "" Whitelist * * *
8 * "@example.com"
8 OK Completed

9 Uce Listexception juser Whitelist * * *
9 * "jokemail@elists.org"
9 OK Completed
```

Removerulegroup

Delete a spam rule group previously added or revised by `Uce Update`.

It is possible to remove all rulegroups on the system. This has the same effect as disabling Antispam scanning, but you can replace a rulegroup with `Uce Update`.

Syntax

```
tag Uce Removerulegroup rulegroup
```

where *rulegroup* indicates the name of a spam identification method.

Privilege Levels

Administrator

Domain Sensitivity

This command cannot be executed in a delegated domain.

Example

```
22 Uce Removerulegroup express
22 OK Completed
```

Report

Report a message as junk mail by copying it to the "`system junkmail.junkmail`" shared folder, or report a message as having legitimate content by copying it to the "`system junkmail.notjunkmail`" shared folder. These folders can be periodically transmitted to Mirapoint for rule-tuning; see the `Setoption` subcommand.

Syntax

```
tag Uce Report type content
```

where *`type`* may be either the keyword `junkmail` or the keyword `notjunkmail`, and *`content`* is the full RFC-2822 message being reported, as a counted string.

Privilege Levels

All administrators and users

Domain Sensitivity

None

Example

```
21 Uce Report junkmail {151+}
From: ube
Subject: Free air
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 7bit

to buyers of Space Ballz!!
21 OK Completed
```

Setoption

Change setting of antispam feature, such as threshold or spam prolog.

The spam prolog is useful for users with one-folder POP email clients. Text of the spam prolog is configurable. If `antispam` is COS-enabled, only users with an LDAP entry containing the following attribute get spam prolog prepended to the subject:

```
miSpamInSubject: On
```

To track scoring, the `X-Junkmail-Status` header is added to messages that were scored. This header should not be used for filtering; it just indicates that scanning was done. There should never be more than one `X-Junkmail-Status` per message. This header is set by the last Mirapoint system to check a message for spam. Header arguments *`score`* and *`thresh`* are the spam score of the message and the threshold that score was compared against; host *`name`* is the system that did the scoring:

```
X-Junkmail-Status: score=score/thresh, host=name
```

Syntax

```
tag Uce Setoption option scope setting
```

where *option* may be one of the following, and for all options *scope* must currently be `(domain=local)`.

- `Headerinfo`—If *setting* is `On`, insert `X-junkmail-Info` header line showing the reasons a message was scored as spam. Default is `Off`. For a list of reasons with explanation, see the http://support.mirapoint.com/products/antispam.html webpage. It is not possible to tune the weighting of test criteria.
- `Multienginebulkonly`—If both Signature Edition Rapid Antispam and Principal Edition Antispam are licensed and *setting* is turned `On`, the appliance runs the first (faster) scanner, then sends "bulk" categorized messages to the second scanner, which is better at separating bulk from junk mail. Default `Off`. If you run this command with only one Antispam scanner licensed, it has no effect until the second scanner is licensed. If two scanners are licensed but this option is `Off`, then the larger spam score wins, which can increase false positives. Running both scanners is resource intensive, but much less so with this option `On`.
- `Reporting`—Enable sending of periodic junk mail encapsulations to MrSpam at Mirapoint for continued rule-tuning. If *setting* is `On`, junk mail messages are subsampled, packaged, encrypted, and transmitted. Default is `On`.
- `Spamprolog`—Enable the spam prolog feature, which inserts "*Spam?*" into the subject line if the antispam software determines that a message is junk mail. The spam prolog *setting* is either `On` or `Off` (the default).
- `Spamprologtext`—Rather than inserting "*Spam?*" into the subject line, prepend the specified spam prolog *setting* instead. To avoid retagging spam, the original subject line is retained in a header field named `X-old-subject`, only one of which should exist. If the message is whitelisted by either the user or domain and the message subject can be restored from `X-old-subject` header, then the `X-old-subject` line is removed.
- `SuspectlistThreshold`—Sets the spam scoring threshold to *setting* for inserting IP addresses into the suspect list; see `Uce Addexception`.
- `Threshold`—Changes the spam scoring threshold to *setting*, which must be a number between 1 and 300. Any email with a junk-mail score higher than this is tagged as spam. Default threshold is 50 points.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
15 Uce Setoption Spamprolog (domain=local) On
15 OK Completed

16 Uce Setoption Spamprologtext (domain=local) "=== spam? ==="
16 OK Completed

17 Uce Setoption Threshold (domain=local) 60
```

```
17 OK Completed
```

```
18 Uce Setoption Multienginebulkonly (domain=local) On
18 OK Completed
```

Update

Revise the spam identification methods for a rule group. The usual rule group is `default` (`rpdengine` for Signature Edition Rapid Antispam); there may be others for various locales.

Rule groups may be installed with this command by specifying the name of a new rule group. Applying a new rule group restarts the SMTP service. Although rule groups expire, they continue working beyond their expiration date. When SMTP is started (but not restarted) a message appears in the system log for each expired rule group.

Syntax

```
tag Uce Update rulegroup
```

where *rulegroup* indicates the name of a spam identification method. If *rulegroup* is asterisk (*) then all installed rule groups will be checked and updated if needed.

Privilege Levels

Administrator

Domain Sensitivity

This command cannot be executed in a delegated domain.

Example

```
11 Uce Update *
11 OK Completed
```

```
12 Uce Update rdpasia
12 OK Completed
```

Version

Return the name, sequence number, and expiration date of spam rule groups.

Syntax

```
tag Uce Version
```

Privilege Levels

Administrator

Domain Sensitivity

This command cannot be executed in a delegated domain.

Example

```
13 Uce Version
13 * "default" "0000" "2003-08-01"
13 OK Completed
```

67 The Update Command

The `Update` command updates the Mirapoint system software over your network.

Subcommands

Install

Downloads and installs a Mirapoint update archive from the specified website URL (uniform resource locator).

A Mirapoint upgrades-allowed license is required to install system software releases ("R" updates), partial dot releases ("P" updates), and enhancements ("E" updates). A license is never required to install defect-correction patches ("D" updates).

System software releases starting with 3 require updates with `.mpu3` file extension, while system software releases starting with 4 require the `.mpu4` extension. It is not necessary to specify `mpu` for an update.

Syntax

`tag` `Update Install` *`update-url`*

where *`update-url`* is the URL of the update archive file. This URL can begin with:

- `ftp://`—Use File Transfer Protocol (FTP) to retrieve the update.
- `http://`—Use Hypertext Transfer Protocol (HTTP) to retrieve the update. This can be an HTTP proxy (unathenticated or authenticated) depending on the value of `Conf Set Httpproxy`.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

First the CLI, then in Protocol:

```
> Update Install ftp://ftp.mirapoint.com/pub/updates/E3_IMAP_uwnamespace
Note: Some updates can automatically reboot the system.  Continue?
(y/n) y
fetching update
UPDATEINFO 1004/1004 bytes received
validating update
installing update
UPDATE E3_IMAP_uwnamespace
OK Completed
```

```
3 Update Install ftp://ftp.mirapoint.com/pub/updates/R_3_8_3_GA
* 3 fetching update
* 3 UPDATEINFO 134672988/134672988 bytes received
* 3 validating update
* 3 installing update
* 3 UPDATE R_3_8_3_GA.mpu3
3 OK
```

List

Responds with a list of updates installed on your Mirapoint system.

Syntax

```
tag Update List
```

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
4 Update List
* 4 sample-1
4 OK
```

Uninstall

Removes the specified update from the system if the update is uninstallable. If the update cannot be uninstalled, `Update Uninstall` returns a NO response.

Syntax

```
tag Update Uninstall update-name
```

where *update-name* is the name of the update you want to remove. This must be one of the update names returned by `Update List` and must be uninstallable.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Update Uninstall sample-1
2 OK
```

68

The Url Command

The `Url` command lets you specify Uniform Resource Locators (URLs) that your Mirapoint system needs to access necessary information.

URL Classes and Instances

Each URL that you can specify must belong to a pre-defined class of URLs that the system requires. The `Addrbook` class allows you to specify multiple LDAP address books for each domain, and the `Groupcalendar` class tracks user preferences and settings for group calendars.

For each class of URLs, you can name and define multiple instances. Each instance of a class of URLs is denoted using the following syntax:

```
class:instance
```

where *`class`* may be `Addrbook`, `Groupcalendar`, or `Xmlcalendar`. With `Addrbook`, *`instance`* is an arbitrary name consisting of letters, numbers, and underscores (_), beginning with a letter or underscore. With class `Groupcalendar`, *`instance`* may be `Grouplookup`, `Userlookup`, `Matchgroup`, or `Matchuser`.

Subcommands

Add

Adds a named URL instance to the specified class.

Syntax

```
tag Url Add class:instance description url options
```

where:

- *`class`* and *`instance`* are as described in URL Classes and Instances on page 651. With `Addrbook` and `Xmlcalendar`, *`instance`* is some arbitrary name. With `Groupcalendar`, *`instance`* may be either `Userlookup`, `Grouplookup`, `Matchuser`, or `Matchgroup`.
- *`description`* is a text description of the URL limited to 256 characters.
- *`url`* is a legal URL as defined by RFC 1738. There are no restrictions on attributes accepted in the URL, except one must be used. Although Mirapoint

does not support (and actually ignores) server-side extensions, all three classes must use the LDAP URL format as defined by RFC 2255, which would be `ldap://`*`fqdn`*`:`*`port`*`/`*`basedn`*`?`*`attributes`*`?`*`scope`*`?`*`filter`*`?`*`extensions`* — for example:

```
ldap://ldap.example.com/ou=mktg,dc=example,dc=com??sub??
```

- *`options`* is a parenthesized list of the following optional arguments, where *`dn`* and *`passwd`* are the distinguished name and password used for authentication. The target LDAP server might support passwords in different formats, which might require *`passwd`* to contain a special prefix; see your LDAP server documentation for details.

  `"(binddn=`*`dn`*`)(bindpasswd=`*`password`*`)"`

Both class `Addrbook` and class `Groupcalendar` accept the `Binddn` and `Bindpasswd` options. For schema neutrality, class `Groupcalendar` (but not `Addrbook`) accepts the `(uidalias="`*`attribute`*`")` option with the correct attribute name for referencing a user's loginID. If `Uidalias` is not set, `Groupcalendar` takes `"uid"` as the default.

There are no restrictions on output attributes except at least one must be defined. After search is done, output attributes are displayed, comma-separated, in the same order as defined. Two examples follow:

```
Url Add Groupcalendar:Userlookup UL1 "ldap://ldap.example.com/o=corp?cn,uid?sub?
    (&(|(objectclass=person)(objectclass=inetorgperson)(objectclass=organizationalperson)
    )
    (|(uid=$(cn)*)(cn=*$(cn)*)(sn=$(cn)*)(givenname=$(cn)*)))" ""
Url Add Groupcalendar:Userlookup UL2 "ldap://ldap.example.com/o=corp?givenname,sn?sub?
    (&(|(objectclass=person)(objectclass=inetorgperson)(objectclass=organizationalperson)
    )
    (|(mail=$(cn)*)(cn=*$(cn)*)(sn=$(cn)*)(givenname=$(cn)*)))" "(uidalias=mail)"
```

To look up LDAP groups for calendar, instance `Grouplookup` accepts the same options as `Userlookup` except for the alias used to uniquely identify groups. The name for the alias is `"cnalias"` and if this option is not defined, the `"cn"` attribute is used to identify groups. Two examples follow:

```
Url Add Groupcalendar:Grouplookup GL1 "ldap://ldap.example.com/
    o=corp?cn?sub?(cn=*$(cn)*)" ""
Url Add Groupcalendar:grouplookup GL2 "ldap://ldap.example.com/
    o=corp?mail?sub?(mail=*$(cn)*)" "cnalias=mail"
```

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
0 Url Add Addrbook:tO TO "ldap://ldap.corp.com/ou=TO,corp=ex,dc=com??sub?" ""
0 OK Completed
1 Url Add Addrbook:AD "Active Directory" "ldap://adserver.example.com:3268/
    DC=adserver,DC=example,DC=com??sub?(&(&(cn=$(cn))(mail=$(mail)))(|(objectc
    lass=person)(objectclass=group)))"
"(binddn=CN=Administrator,CN=Users,DC=adserver,DC=example,DC=com)(bindpasswd=password)"
1 OK Completed
2 Url Add Groupcalendar:Userlookup GroupcalUserLookup "ldap://ldap.corp.com/
    o=corp?cn,uid?sub?(&(|(objectclass=person)(objectclass=inetorgperson)(obje
    ctclass=organizationalperson))(|(uid=$(cn)*)(cn=*$(cn)*)(sn=$(cn)*)(givenn
    ame=$(cn)*)))" ""
2 OK Completed
3 Url Add Groupcalendar:Grouplookup "GroupcalGroupLook" "ldap://ldap.corp.com/
    o=corp?mail?sub?(mail=*$(cn)*)" "cnalias=mail"
3 OK Completed
4 Url Add Groupcalendar:Matchuser MU
    "ldap://ldap.example.com/o=corp?cn?sub?(milogin=$(cn))" uidalias=milogin
5 Url Add Groupcalendar:Matchgroup MG
    "ldap://ldap.example.com/o=corp?cn?sub?(cn=$(cn))" ""
6 Url Add Xmlcalendar:beginswith filter1
    "ldap://ldap.example.com/o=corp?cn?sub?(uid=$(cn)*)" ""
7 Url Add Xmlcalendar:exactsearch filter2
    "ldap://ldap.example.com/o=corp?cn?sub?(uid=$(cn))" ""
```

Count

Responds with the number of URLs matching the specified pattern.

Syntax

tag `Url Count` *pattern*

where *pattern* is a string optionally containing wildcard characters (see Using Patterns on page 49.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
8 Url Count ""
```

```
* 8 1
8 OK Completed
```

Delete

Deletes the specified URL instance.

Syntax

```
tag Url Delete class:instance
```

where *class* and *instance* are as described in URL Classes and Instances on page 651.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
10 Url Delete addrbook:t0
10 OK Completed
```

List

Responds with a list of URLs.

Syntax

```
tag Url List pattern start count
```

where:

- *pattern* is a string optionally containing wildcard characters (see Using Patterns on page 49.
- *start* is the first position in the list of networks and domains that you want to see. The empty string (`""`) implicitly means `0`.
- *count* is number of networks and domains that you want to see. The empty string (`""`) implicitly means all networks and domains. If *count* is greater than the total number of networks and domains, `list` returns as many networks and domains as possible.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
9 Url List "" "" ""
* 9 addrbook:t0 T0 ldap://ldap.corp.com/ou=T0,dc=corp,dc=com??sub? ""
9 OK Completed
```

69 The User Command

+ABOUT+++

The `User` command lets you manage the user database on your Mirapoint system. You can list users, add and remove users, change users' passwords, and perform other user customization and administration tasks.

Periods in User and DL Names

Mirapoint user names cannot contain a period (.), although several work-arounds exist to let underscore (_) stand for the period. If user `john_smith` exists, email to `john.smith` is delivered to that user, and the user may log in as `john.smith`.

Distribution lists (DLs) may contain periods. The DL `john.smith` is distinct from the DL `john_smith`, which causes some confusion when DLs mix with user names. If both user `john_smith` and DL `john.smith` exist, when `john_smith` has autoreply or forwarding set, it applies to the email address `john.smith` (the DL) as well as to the user name `john_smith`, resulting in all mail sent to the DL being autoreplied to, or sent to the forwarding address set by `john_smith`.

Since Mirapoint tries to circumvent the lack of support for period (.) in user names, it is unsafe to have user names and DLs that collide this way.

When moving to or provisioning a Mirapoint system, if there is a collision either between two users, or between a DL and user, one should be renamed. Fortunately if LDAP is in use, an LDAP mapping can preserve email addresses and logins while allowing the actual user names to be sufficiently distinct.

+HEAD++

Note

This command has no effect on external databases used for login authentication, as on Lightweight Directory Access Protocol (LDAP) or Network Information Service (NIS) servers. To manage an external user database, see the vendor documentation.

Logins for the user `Administrator` are always authenticated using the local user database.

+HEAD++

Using Patterns

The User Count and User List commands allow you to specify a **pattern** for users' login names and full names. Patterns are case-insensitive and can contain these wildcard characters:

- `?`——matches any single character.
- `*`——matches zero or more characters of any kind.

For example, the pattern `ann?`, when passed to the `User List` command, might match these login names:

- `anna`
- `anne`

The pattern `jo*` might match these full names:

- Jo Grant
- Joe Morgan
- John Kennedy
- Jon Carroll
- Joseph Campbell
- Josephine Baker

Character [A-Z] and digit [0-9] ranges are not supported. For other examples, see `User Count` and `User List`.

Subcommands

+CMD+++

Add

Creates a user account.

User login names may include characters from the range `[A-Za-z 0-9_-]` but the following names are reserved: administrator, administrators, anonymous, anybody, anyone, and nobody. User names are case-insensitive.

When the number of users exceeds the licensed limit, this command returns No and does not create a new user. Even with the user-unlimited license, there is also a hard maximum, raised from 250,000 to 500,000 on larger appliances in release 3.5.

Syntax

tag `User Add` *username passarg fullname*

where:

- *username* is the unique username that the user will use to log in. This string must be no longer than 80 characters.
- *passarg* is one of:
  - A string of the form "(*passtype*=*password*)" where *password* is a string, the format of which is indicated by *passtype*, which is one of:
    - `Cryptpass`——indicates the user's password is `crypt()`-encoded.
    - `Encodedpass`——indicates the user's password is a Mirapoint-specific two-way encrypted version of the user's password (this is used only by the Messaging Infrastructure Manager).
    - `Pass`——indicates the user's password is plaintext.
  - Instead, a plaintext password for the user. Equivalent to `(Pass=`*password*`)`. This string must be no longer than 80 characters. It may contain non-ASCII characters, but no Unicode normalization occurs, so varying input methods could cause password incompatibility across platforms. In the CLI if you omit this argument, the interface prompts interactively for a password. You cannot add an unencrypted password containing parentheses, but you can do this with the User Set Pass command.
  - A string of the form "(`uuid=`*uniqueUserID*)" where *uniqueUserID* is a UUID derived from `User Get Uuid`, or a valid `miUuid` attribute value from LDAP (see `Conf Enable Ldapuuid`).

If you always use an external database such as NIS or LDAP for login authentication, you can specify any string for *password*, because it is ignored.

- *fullname* is the user's full name——you must enclose this in double quotes to allow a space between the user's first and last names. This string must be no longer than 80 characters.
- *fullname* is the user's full name—ordinarily, you must enclose this in double quotes to allow a space between the user's first and last names as shown in the example below, or use a literal string (see Literal Strings on page 48). This string must be no longer than 80 characters.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
2 User Add glenn glenn "Glenn Green"
2 OK Completed
```

+CMD+++

Count

Responds with the number of user accounts on your Mirapoint system.

Syntax

tag `User Count` *pattern*

where *pattern* has one of the following forms:

- "" (empty string)——equivalent to the wildcard `*`, meaning all login names.
- *value*——a pattern string, optionally containing wildcard characters, matching login names to be counted (see Help About UserUsing Patterns on page 49).
- `"*@*"`——a special pattern to count all users on the system, in all domains.
- `"(username=`*value*`)(fullname=`*value*`)"`——specifies patterns, optionally containing wildcard characters, for both login (user) names and full names. Only users whose login *and* full names match the specified values are counted. You may specify either or both key/value pairs.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain. The special pattern `*@*` applies to all domains but can be run only from the primary domain.

Example

```
14 User Count demo*
* 14 4
14 OK Completed
```

```
15 User Count "(username=demo*)(fullname=*Big*)"
* 15 1
15 OK Completed
```

+CMD+++

Delete

Deletes a user from your Mirapoint system.

Note: You cannot delete the last domain administrator for a delegated domain.

Syntax

```
tag User Delete username
```

where *username* identifies the user you want to delete.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
8 User Delete glenn
8 OK Completed
```

+CMD++

Get

Gets the value of the specified parameter for a particular user.

Syntax (Administrator-Authenticated)

```
tag User Get parameter username
```

Syntax (User-Authenticated)

```
tag User Get parameter
```

Syntax

```
tag User Get parameter username
```

where:

- *parameter* is one of:
  - `Auth`——the list of authentication schemes that the user may use for administration, POP, and IMAP logins. The value `Default` indicates use of the system default authentication schemes (see `Auth Set`Chapter 5, The Auth Command).
  - `Cryptpass`——the user's `crypt()`-encoded password; if the user's password is not `crypt()`-encoded, `User Get` returns a NO response.

- ❖ `Deliveryaddress`——the user's real email address with underscore (_) replacing dot (.) for use by `Conf Enable Keeptomailhost`.
- ❖ `Encodedpass`——a Mirapoint-specific two-way encrypted version of the user's password (used primarily for backup and restore).
- ❖ `Fullname`——the user's real name.
- ❖ `Junkmailsummary`——whether the user receives junkmail summary reports daily or whatever, and whether this convenience is disabled.
- ❖ `Junkmailsummaryemptylogs`——whether or not empty summaries appear.
- ❖ `Login`——whether this user's logins are currently enabled. See `User Set`.
- ❖ `Rights`——responds with a list of keywords identifying the operations the specified user can perform. These can be any of:
  - `antispam`——the user may participate in junk mail filtering
  - `autoreply`——the user may change automatic message reply settings
  - `changepass`——users may change their own password
  - `filter`——users may change message filters for their own mailbox
  - `forward`——the user may change message forwarding settings
  - `getmail`——user can retrieve messages from remote POP mailboxes
- ❖ `Uuid`——retrieves the specified or current user's UUID. This unique user ID is preceded by token LOCAL, LDAP, or UUID, depending on whether the value is present just locally, only in LDAP, or both are in agreement. If there are problems with the user's UUID, specific errors are printed as warnings. Running `Conf Enable Ldapuuid` causes UUIDs to be written to LDAP.
- ❖ `Wmaddrbook`—retrieves complete address book for the specified user, in LDIF format. To see data fields that are supported, create address book entries using WebMail Direct, then get them with this command.
- ❖ `Wmdictionary`—returns the WebMail personal spell checking dictionary for the specified user as a string literal.
- ❖ `Wmprefs`—retrieves WebMail Direct preferences for the specified user. These can also be stored in LDAP; see `Conf Enable Ldapwmprefs`.

- ◆ *`username`* (administrator-authenticated only) identifies the user for whom you want to get the value of *`parameter`*.

Privilege Levels

- ◆ Administrator
- ◆ Helpdesk administrator
- ◆ Domain administrator
- ◆ Backup operator
- ◆ Users (can access only their own accounts)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
4 User Get Fullname glenn
* 4 "Glenn Green"
4 OK Completed
```

```
5 User Get Auth glenn
* 5 "kerberos_v4:kerberos_v4"
5 OK Completed
```

```
6 User Get Uuid jsuser
* 6 WARNING "LOCAL and LDAP UUID mismatch"
* 6 LOCAL 0c4c0a3e-4fb9-1029-8df4-0007e94069cc
* 6 LDAP 58d753ae-4fb9-1029-8c85-0007e94069cc
6 OK Completed
```

```
17 User Get Wmaddrbook me
* 17 {411}
dn: uid=0,cn=Joseph User
uid: 0
cn: Joseph User
anniversarymonth: 0
birthyear: 1950
birthmonth: 1
birthday: 1
category: 3
primaryphone: 0
uuid: 15f4b8e2-5c5b-1025-83f5-00a0c9b4e306
sn: User
givenname: Joseph
mail: juser@example.com
o: Example Inc.
postaladdress: 12 Fullpress Court
l: Anytown
st: ME
postalcode: 00001
telephonenumber: 999-999-9999
displayname: Joseph User
objectclass: top
objectclass: person

17 OK Completed
```

```
18 User Get Wmprefs me
* 18 {32}
timezone = America/Los_Angeles;

18 OK Completed
```

+CMD+++

List

Responds with a list of user accounts on your Mirapoint system.

Syntax

tag `User List` *pattern start count*

where:

- *pattern* has one of the following forms:
  - "" (empty string)——equivalent to the wildcard character *, meaning all login names.
  - *value*——a pattern string, optionally containing wildcard characters, matching the login names you want to list (see Help About UserUsing Patterns on page 49).
  - "(username=*value*)(fullname=*value*)"—— specifies pattern strings, optionally containing wildcard characters, for both login names and full names. Only those users whose login names *and* fullnames match the specified values are listed.
- *start* is the number of the first user you want to see. The empty string ("") implicitly means 0.
- *count* is the number of users you want to see. The empty string ("") implicitly means all users. If *count* is greater than the total number of users, list returns as many users as possible.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
12 User List demo* "" ""
* 12 "demo" "demo"
* 12 "demo1" "Demo One"
* 12 "demo2" "Demo Two"
* 12 "demo3" "Demo \"The Big Guy\" Three"
12 OK Completed
```

```
13 User List "(username=demo*)(fullname=*Big*)" "" ""
* 13 "demo3" "Demo \"The Big Guy\" Three"
13 OK Completed
```

Rename

Changes the login name of the specified user and renames the users inbox.

In non-LDAP configurations, `Rename` also changes user names for Group Calendar. In LDAP configurations, user directory records must also be changed, for instance with `Dir ModifyLdif`. First change DN using the "`changetype: moddn`" keyword, then change user attributes using the "`changetype: modify`" keyword.

Syntax

```
tag User Rename login newlogin
```

where:

- ◆ *`login`* is the user's current login name. When the command completes, the user can no longer log in using this name.
- ◆ *`newlogin`* is the new login name you want assign to the user. When the command completes, the user must use this name to log in.

Privilege Levels

Administrator

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
16 User Rename demo1 demo4
16 OK Completed
```

+CMD++

Set

Sets the value of the specified parameter for a particular user. For example, the following is a "user disable" command for the `juser` account:

```
User Set Login juser Off
```

Syntax (Administrator-Authenticated)

```
tag User Set parameter username value
tag User Set Pass username
```

Syntax (User-Authenticated)

```
tag User Set parameter value
tag User Set Pass
```

Syntax

```
tag User Set parameter username value
```

where:

- *parameter* is one of:
  - Auth——the list of authentication schemes that the user may use for administration, POP, and IMAP logins. If you specify the value Default for this parameter, the user gets the system default authentication schemes (see Auth SetChapter 5, The Auth Command).

    You may specify only one PLAINTEXT authentication scheme in this list. Moreover, you may not specify any authentication scheme for the user Administrator except PLAINTEXT:LOCAL.
  - Cryptpass——a crypt()-encoded password.
  - Deliveryaddress——the user's real email address, with underscore (_) replacing dot (.) to circumvent dot-to-underscore mapping. Junk Mail Manager (JMM) sets this if the user's name (left hand side of address) contains a dot. This value is used by message send, JMM summaries, Conf Enable Keeptomailhost, and overquota messages via Keeptomailhost. Null string *value* (" ") resets.
  - Encodedpass——a two-way encrypted version of the user's password.
  - Fullname——the user's real name; you must enclose this string in double quotes to allow a space between the user's first and last names. This string must be no longer than 80 characters.
  - Fullname—the new full name for the user; you must enclose this string in double quotes (as shown in the example below) to allow a space between the user's first and last names, or use a literal string. This string must be no longer than 80 characters.
  - Junkmailsummary——how often and whether the user receives summary reports showing messages classified as spam stored by Junk Mail Manager. The *value* can be specified as Daily, Weekdays, Weekly, Off, or Always (always means as frequently as the administrator selects).
  - Junkmailsummaryemptylogs——whether a user receives summary reports with no entries. The default is On. Setting Off suppresses empty summaries.
  - Login——whether logins for this user are currently enabled (ON or OFF). If set to ON, the user may log into the system using any service, such as POP or IMAP. If set to OFF, the user is denied access. Changes in this setting take effect the next time the user tries to log in. Changing this setting for the user administrator is not allowed.
  - Pass——a plain-text password, which may contain only 7-bit ASCII characters and must be no longer than 80 characters.
  - Wmaddrbook—given data in LDIF format, replaces the address book for a specified user. The *value* is specified as a literal string; see Literal Strings on page 48. To see the supported data fields, create address book entries using WebMail Direct, then retrieve them with User Get Wmaddrbook. The maximum number of address book entries is 1000 by default; this can be changed in branding, but larger values have performance implications.
  - Wmdictionary—creates the WebMail personal spell checking dictionary for user *username*, specified as a string literal *value*.

- `Wmprefs`—given data in attribute=value format, modifies WebMail Direct preferences for the specified user. Table 6 shows LDAP attributes and the equivalent administration protocol attribute name, if different.

Table 6 LDAP Attributes for WmPrefs (CIS = Case Ignore String)

LDAP Attribute	Attribute	Type	Description
miWmprefCharset	charset	CIS	Character set
miWmprefColorTheme	colortheme	CIS	Background color scheme
miWmprefComposeHeight	composeheight	Integer	Height of compose window
miWmprefComposeWidth	composewidth	Integer	Width of compose window
miWmprefConfirmReadReceipt	confirm_read_receip t	Boolean	Ask before sending read receipt
miWmprefDraftFolder	draftfolder	CIS	Draft folder name
miWmprefEmailAddress	email	CIS	Preferred 'From' Address
miWmprefFullname	fullname	CIS	Preferred full name for user
miWmprefIncludeSignature	includesig	Boolean	To include signature or not
miWmprefMessageCount	header	Integer	Number of messages in header
miWmprefNewArrivalFirst	newarrivalfirst	Boolean	Show newest messages at top
miWmprefReplyOption	replyopt	CIS	Reply Option
miWmprefReplytoAddress	reply-to	CIS	Preferred 'Reply-To' Address
miWmprefReqDeliveryReceipt	req_delivery_receipt	Boolean	Request receipt for mail delivery
miWmprefReqReadReceipt	req_read_receipt	Boolean	Request receipt for reading mail
miWmprefSaveSent	savesent	Boolean	Whether to save sent messages
miWmprefSentFolder	sentfolder	CIS	Sent Folder name
miWmprefShowMessagePane	showmessagepane	Boolean	Corporate Edition three-pane
miWmprefSignature	signature	CIS	Signature
miWmprefTimezone	timezone	CIS	Timezone
miWmprefTrashFolder	trashfolder	CIS	Trash folder name
miWmprefUseHtml	usehtml	Boolean	Whether to use HTML compose
miWmprefUseTrashFolder	usetrash	Boolean	Whether to use trash folder
miWmprefVersion	version	CIS	WebMail Display version

- *`username`* (administrator-authenticated only) identifies the user for whom you want to set *`parameter`*.
- *`value`* is the value you want to assign to *`parameter`*.

The special command `User Set Pass` does not take a *`value`* parameter——instead, it prompts you to enter a new password.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator (except `Auth` parameter)
- Users (can access only their own accounts for `Login` and password settings)

Domain Sensitivity

This command applies to the current delegated domain. If no delegated domain is current, the command applies to the system's primary domain.

Example

```
7 User Set Fullname glenn "Glenn G. Green"
7 OK Completed

8 User Set Auth glenn "APOP:LDAP PLAINTEXT:LDAP"
8 OK Completed

19 User Set wmaddrbook me {411}
dn: uid=0,cn=Joseph User
uid: 0
cn: Joseph User
anniversarymonth: 0
birthyear: 1950
birthmonth: 1
birthday: 1
category: 3
primaryphone: 0
uuid: 15f4b8e2-5c5b-1025-83f5-00a0c9b4e306
sn: User
givenname: Joseph
mail: juser@example.com
o: Example Inc.
postaladdress: 12 Fullpress Court
l: Anytown
st: ME
postalcode: 00001
telephonenumber: 999-999-9999
displayname: Joseph User
objectclass: top
objectclass: person

19 OK Completed

20 User Set wmaddrbook me {32}
timezone = America/Los_Angeles;

20 OK Completed
```

70

The Webmail Command

The `Webmail` command configures behavior of the WebMail Direct service.

WebMail (and related services) require licenses, which may have been preinstalled at the factory. Licensing automatically enables and starts the service. You can halt and disallow WebMail with the `Service Stop` and `Service Disable` commands.

From a browser, URLs of WebMail Direct and Administration Suite are as follows:

```
http://miServer/wm/mail/window.html?sessionid=sesID&op=blank
http://miServer/cgi-bin/entry.cgi/entry/login.html?sessionid=sesID&op=entry
```

Subcommands

Get

Responds with the value of the specified parameter.

Syntax

```
tag Webmail Get parameter
```

where *parameter* must be one of those documented under `Webmail Set`.

Privilege Levels

- Administrator
- Backup operator

Domain Sensitivity

None

Example

```
1 Webmail Get Timeout
* 1 30
1 OK Completed
```

Set

Sets the value of the specified parameter.

Syntax

tag `Webmail Set` *parameter value*

where:

- *parameter* must be one of the following:
  - `Timeout`——specifies the idle-period timeout for WebMail sessions. WebMail service automatically logs out any user whose connection remains idle for *value* number of minutes.
  - `Omr`——changes the outbound message router (OMR), the SMTP server that WebMail contacts when sending messages. The hostname *value* specifies the OMR for sending mail messages composed in WebMail. Current restrictions are as follows: connections are effective only with Mirapoint servers; SSL is not supported; MX lookups are not done; no SMTP settings are honored except `Masq`; SMTP error messages are passed directly to the user and are not localized; there is no logging of `Omr`-sent messages; SMTP AUTH is not done (authentication information is passed by the ESMTP AUTH clause of the `Mail From` command).
  - `Maxnumrecips`—specifies the maximum number of recipients per sent email. Acceptable value range is 0 - 2147483647. The value pertains to the number of recipients before distribution-list expansion. This parameter, along with `Maxmsgrate` and `Maxreciprate`, helps prevent the sending of large volumes of email from compromised mail accounts. These parameters define limits on the amount of mail that can be generated from an account. If users try to send a message that exceeds these limits, WebMail displays a message informing them that they have exceeded their mail-sent quota and an alert is generated for the administrator.
  - `Maxmsgrate`—specifies the maximum number of messages per hour that can be sent for a given user. Acceptable *value* range is 0 - 2147483647.
  - `Maxreciprate`—specifies the maximum number of recipients to which mail can be sent in an hour, for a given user. Acceptable *value* range is 0 - 2147483647.
- *value* is the value you want to assign to *parameter*.

Privilege Levels

Administrator

Domain Sensitivity

None

Example

```
2 Webmail Set Timeout 30
2 OK Completed
```

```
3 Webmail Set Omr smtp.example.com
3 OK Completed

4 Webmail Set Maxnumrecips 250
4 OK Completed

5 Webmail Set Maxmsgrate 40
5 OK Completed

6 Webmail Set Maxreciprate 400
6 OK Completed
```

71

The Wordlist Command

The `Wordlist` command specifies custom wordlists, which can be used as criteria to filter messages containing obscene words, company-prohibited words, and so forth. The Administration Suite employs the wordlist facility for its **Corporate Word List** and **Objectionable Word List** pages. Wordlists can also be managed with the CLI.

You create a wordlist with the `Import` subcommand, supplying patterns separated by newlines, or specifying the URL of such a file. Each wordlist has a name and is associated with a domain. Once established, the wordlist can be invoked in a rule by the `Filter` command, as in this CLI example:

```
Filter Add "(domain=example.com)" 4LW redirect "admin@host" anyof stop
:body contains-wordlist "(domain=primary)(name=Four letter words)"
.
```

The filter attribute can be message `body` (as above), some header, or any attribute specified in Rules on page 240. For best performance, it is best to keep word lists as short as possible, especially with `:bodydecodedbinary`.

How Wordlists Work

Wordlists are processed as follows:

- A wordlist is imported for the current domain. Each line forms a pattern, which is UTF-8 normalized, as defined by the Unicode specification. Upper case letters are converted to lower case.
- Each pattern is parsed into words and delimiters. Words are composed of ASCII alphanumeric characters (hex 30-39, 41-5A, 61-7A) plus all characters above hex 80, the range of Unicode characters. Delimiters include spaces and ASCII punctuation marks (hex 20-2F, 3A-40, 5B-60, 7B-7E).
- The filter attribute (portion of an email message) is also UTF-8 normalized, unless it is a header address or `:bodydecodedbinary` attachment. This is because header addresses must be ASCII, and binary attachments are not Unicode.
- Implicit delimiters are placed at the beginning and end of both the pattern and the filter attribute.
- Filter rules starting with "`:attachmenttype`" or "`:attachmentfilename`" receive special treatment to make them easy to parse (see below).

For each pattern (line), wordlists are compared as follows:

- The search library understands UTF-8 and calculates the number of bytes forming each Unicode character, and compares Unicode codepoints, converting the filter attribute from upper to lower case.
- The pattern and filter attribute are compared word by word, moving forward one word at a time, words being separated by delimiters.
- Strings of delimiters are treated as a single delimiter. Any delimiter matches any other delimiter.
- If all the words match and we run out of words in the pattern, comparison returns `MATCH`, and searching terminates.
- Otherwise when the comparison reaches the end of the filter attribute without matching all words in the pattern, comparison returns `NOMATCH`, and searching continues with the next pattern (line) in the wordlist, if any.

The intention is that all the following should match the “end start” pattern:

```
Fragment end... start another!
Fragment end? Start another.
Fragment end!! Start another.
“Fragment end.” “Start another.”
```

Also, pattern “a@b.com” matches any of the following, but neither “a1@b.com” nor “abba@b.com” addresses (all these forms are in common use today):

```
a@b.com
"a"@b.com
<a@b.com>
<"a"@b.com>
```

Note that pattern “user1@example.com” matches input “user1????example.com” because all delimiters match each other, and multiple delimiters are compressed to a single delimiter. Hint: when filtering email addresses, it might be easier to use the `Regexmatches`, `Matches`, or `Contains` match type, not `Contains-wordlist`.

The `:attachmenttype` and `:attachmentfilename` rules are treated specially:

- If the wordlist pattern starts with a period (.) it is interpreted as a file extension. The filter attribute attachment is searched for that file extension, ensuring that the file-extension name ends with a space, semicolon (;) or slash (/).
- If the wordlist pattern starts with a slash (/) it is interpreted as a MIME type. The filter attribute attachment is searched for that MIME type, ensuring that the MIME type ends with a space, semicolon (;) or slash (/).
- Normal wordlist search continues if the above two steps fail to match.

Subcommands

Count

Returns the number of wordlists on the system.

Syntax

tag `Wordlist Count` *pattern*

where *pattern* must be "" or * to match all wordlists.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Wordlists are specific to a delegated domain or the top-level domain.

Example

```
5 Wordlist Count ""
* 5 2
5 OK Completed
```

Delete

Erases the specified wordlist from the system.

Syntax

tag `Wordlist Delete` *listname*

where *listname* is a wordlist as specified by `Wordlist Import`.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

Wordlists are specific to a delegated domain or the top-level domain.

Example

```
7 Wordlist Delete company
7 OK Completed
```

Export

Outputs the specified wordlist as a string literal.

Syntax

```
tag Wordlist Export listname
```

where *listname* is a wordlist as specified by `Wordlist Import`.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Wordlists are specific to a delegated domain or the top-level domain.

Example

```
4 Wordlist Export company
{19}
Beispiel
Ejemplo
4 OK Completed
```

Import

Creates a new wordlist given a string literal or URL. If a wordlist of that name already exists, it is replaced.

Syntax

```
tag Wordlist Import listname wordlist
```

where:

- *listname* is the (case-sensitive) name of a wordlist. Names beginning with the prefixes “`System:`” and “`SR:`” are reserved for use by Mirapoint. Any *listname* beginning with “(” is interpreted as a set of optional arguments for referring to lists and domains outside the current domain, including:
  - `(domain=`*domainname*`)`—where *domainname* is a delegated domain.
  - `(name=`*listname*`)`—where *listname* designates the wordlist name.
- *wordlist* is a string literal or a URL. If *wordlist* is a URL, it must begin with `http://`, must not contain a newline, and must be followed by nothing else.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator

Domain Sensitivity

Wordlists are specific to a delegated domain or the top-level domain.

Example

```
2 Wordlist Import obscene http://not.fcc.us.gov/bannedwordlist
2 OK Completed
```

```
3 Wordlist Import confidential {12+}
CMOS
DRAM

3 OK Completed
```

List

Returns names of wordlists on the system.

Syntax

```
tag Wordlist List pattern start count
```

where:

- `pattern` must be "" or * to match all wordlists.
- `start` is the first wordlist to return.
- `count` is number of wordlists to return.

Privilege Levels

- Administrator
- Helpdesk administrator
- Domain administrator
- Backup operator

Domain Sensitivity

Wordlists are specific to a delegated domain or the top-level domain.

Example

```
6 Wordlist List "" "" ""
* 6 company
* 6 obscene
6 OK Completed
```

Appendix A Sample CLI and Protocol Sessions

This appendix shows many administrator interactions using the CLI (command line interface) and the administration protocol. What you type appears in boldface.

Examples of CLI Use

These commands look up the A record for hostname www.example.com:

```
Dns Lookup www.example.com ""
A 64.124.80.66
Dns Lookup www.example.com type=A
A 64.124.80.66
```

This command looks up the PTR record for the given IP address.

```
Dns Lookup 64.124.80.67 type=PTR
PTR support.example.com
```

This command looks up all the MX records for domain example.com:

```
Dns Lookup example.com type=MX
MX 50 mercury.example.com
MX 20 venus.example.com
MX 10 mars.example.com
```

This command looks up CNAME alias(es) for the given hostname:

```
Dns Lookup news.example.com type=CNAME
CNAME corp.supernews.com
```

This command adds a secondary DNS server, always a good idea:

```
Dns List
10.0.0.254
Dns Add 10.10.0.254
```

This command tests for network connectivity of the given server:

```
Diag Ping mail.example.com 1
"PING 192.168.0.18: 56 data bytes"
"Received 64 bytes from 192.168.0.18: seq=0 ttl=253 time=1.14 ms"
"Statistics for 192.168.0.18:"
"Packets:  1 transmitted,  1 received,  0 (0%) lost"
"Round trip time: min=1.14 ms,  max= 1.14 ms,  ave= 1.14 ms"
```

These commands set timezone, add an NTP server, and synchronize time:

```
Ntp Set Zone America|Los_Angeles
Ntp Add ntp.example.com
Ntp Synchronize ntp.example.com
```

These commands show current licenses, if any, and retrieve the system licenses that your organization purchased from Mirapoint:

```
License List
License Fetch
```

These commands enable and start SMTP service, which is required on all Mirapoint servers after configuration, because SMTP is not started by default.

```
Service Enable Smtp
Service Start Smtp
```

These commands add an LDAP database, set the administrator password, index an important field, and import LDAP schema from the Mirapoint website:

```
Dir Adddb miratop
Dir Adddbsuffix miratop o=miratop
Dir Setdboption miratop RootDN uid=administrator,o=miratop
Dir Setdboption miratop RootPW !YourAdminPassword!
Dir Addindex "" miloginid eq
Dir Importldif o=miratop "c" http://www.mirapoint.com/support/allinone.ldif
```

These commands initiate directory service, and set local host as the LDAP server:

```
Service Enable Dir
Service Start Dir
Ldap Add ldap://127.0.0.1:389
```

These commands set up LDAP queries, so Mirapoint servers can correctly interpret LDAP information. The user:Publishedname query acts as a template for the next six queries, but not for mailgroup:Members.

```
Ldap Setquery user:Publishedname o=miratop (|(mail=$(login))(miloginid=$(login))) mail ""
          Ldap Setquery user:Mailhost "" "" mailhost ""
          Ldap Setquery user:Routingaddr "" "" mail ""
          Ldap Setquery user:Quota "" "" mimailquota ""
          Ldap Setquery user:Loginid "" "" miloginid ""
          Ldap Setquery user:Uuid "" "" miuuid ""
```

This is a query with 7 space-separated arguments; there should be no line break in the middle. The last two arguments are quoted because they contain a space:

```
Ldap Setquery mailgroup:members o=miratop (&(objectclass=mailgroup)(cn=$(group)))
          "mgrpRFC822MailMember uniqueMember" "direct indirect"
```

This command allows access to the o=miratop hierarchy in the LDAP database. The password here must match the one in `Dir Setdboption` above:

```
Ldap Addaccess o=miratop uid=administrator,o=miratop pass
Password: !YourAdminPassword!
```

This command enables LDAP authentication. User passwords are stored in LDAP, not on the local server. This is recommended for multi-tier deployments:

```
Auth Set Default Plaintext:ldap
```

This command enables the autoprovisioning of accounts, if they do not yet exist, from user records in the LDAP database:

```
Ldap Set Autoprovisioning On
```

The first command enables LDAP routing. The second command also enables LDAP routing and six other LDAP features, including password updates in LDAP, highly recommended. See `Conf Enable` for details.

```
Conf Enable Ldaprouting
Conf Enable Ldapall
```

These commands enable LDAP service proxying, so users login to a front-end that connects them to IMAP, POP, or HTTP (WebMail/WebCal) on their home server:

```
Imap Set Mode Ldapproxy
Pop Set Mode Ldapproxy
Http Set Mode Ldapproxy
```

These commands set the front-end default to display the login page for WebMail, and enable multi-language selection from the login-page footer:

```
Http Set Root Webmail
Locale Set Loginfooter On
```

These commands set up checking for Class of Service so that some users can get premium services unavailable to others:

```
Cos Enable Antispam
Cos Enable Antivirus
Cos Enable Autoreply
Cos Enable Calendar
Cos Enable Enterpriseui
Cos Enable Filter
Cos Enable Forward
Cos Enable Getmail
Cos Enable Groupcalendar
Cos Enable Imap
Cos Enable Msgexpiration
Cos Enable Msgundelete
Cos Enable Pop
Cos Enable Quota
Cos Enable Sender_as
Cos Enable Sender_av
Cos Enable Ssl
Cos Enable Webmail
```

These commands add a delegated domain administrator with ability to manage 50 users (increased from 20) in the “example.com” domain:

```
Domain Add example.com
Domain Setcurrent example.com
User Add adminname password "Firstname Lastname"
Role Add Admin adminname
Domain Set Userquota delegated.com 50
```

These commands set up NIC failover between ports 0 and 1:

```
Netif Addlogical "" Failover
Netif Bindlogical logical0 port0
Netif Bindlogical logical0 port1
Netif Bind logical0 10.0.11.18/16
Netif Setlogical Mode logical0 Activefailback
```

Adding User Accounts

The session below creates a user account with the login name `glenn`, sets up the account, then lists all user accounts. See Chapter 69, The User Command for more information about these commands.

Sample Session

```
* OK host.mycompany.com admind 1.2.4 server ready
1 Login Administrator admin
* 1 134273423
1 OK User logged in
2 User Add glenn glenn "Glenn Green"
2 OK Completed
3 User Set Pass glenn glenn
3 OK Completed
4 User Set Cryptpass glenn xcxhC/7vlpxkw
4 OK Completed
5 User Set Fullname glenn "Glenn G. Green"
5 OK Completed
6 User Get Fullname glenn
* 6 "Glenn G. Green"
6 OK Completed
7 User Count ""
* 7 5
7 OK Completed
8 User List "" "" ""
* 8 "Administrator" "Administrator"
* 8 "a0" "a0"
* 8 "a1" "a1"
* 8 "demo" "Demo User"
* 8 "glenn" "Glenn G. Green"
8 OK Completed
9 Logout
9 OK Completed
```

Adding Mailboxes

The session below creates an inbox `user.glenn` with a subfolder, manipulates access control lists, renames the subfolder, then lists all mailboxes. See Chapter 34, The Mailbox Command fir more information about these commands.

Sample Session

```
* OK host.mycompany.com admind 1.2.4 server ready
1 Login Administrator admin
* 1 134273473
1 OK User logged in
2 Mailbox Add user.glenn
2 OK Completed
3 Mailbox Add user.glenn.personal
3 OK Completed
4 Mailbox Count ""
* 4 7
4 OK Completed
5 Mailbox Getacl user.glenn.personal
* 5 user.glenn.personal glenn lrswipcda
5 OK Completed
6 Mailbox List user.g* "" ""
* 6 () "." user.glenn
* 6 () "." user.glenn.personal
6 OK Completed
7 Mailbox Setacl user.glenn.personal Administrator +d
7 OK Completed
8 Mailbox Rename user.glenn.personal user.glenn.private
8 OK Completed
9 Mailbox Setacl user.glenn.private glenn adilprsw
```

```
9 OK Completed
10 Mailbox Setacl user.glenn.private anyone -lrs
10 OK Completed
11 Mailbox Setacl user.glenn Administrator +d
11 OK Completed
12 logout
12 OK Completed
```

Creating Distribution Lists

The session below creates distribution lists by importing sendmail-style aliases, and then creating a single distribution list named `junk`. It then adds several entries to this distribution list, first using a bulk add, and then by adding a single entry.

See Chapter 16, The Dl Command to learn about the subcommands used here.

Sample Session

```
* OK host.mycompany.com admind 1.2.4 server ready
1 Login Administrator admin
* 1 134273464
1 OK User logged in
2 Dl Importsendmail {212+}
a1:
    fred,
    a2,
    a3,
    u1,
    u10,
    u1001
a3:
    a1,
    a2,
    a3,
    a4,
    fred,
    u1,
    u10,
    u1001
refuse:
    Administrator
betty:
    u1,
    u10,
    u100
fred:
    u100,
    u1000,
    u101,
    u102,
    u103,
    u104

2 OK Completed
3 Dl Count ""
* 3 6
3 OK Completed
4 Dl List "" "" ""
* 4 L a1
* 4 L a3
* 4 L betty
* 4 L daily-reports
* 4 L fred
* 4 L refuse
4 OK Completed
5 Dl Add junk
5 OK Completed
6 Dlentry Addbulk junk {35+}
fred
ralph
julie
steph
xerxes

6 OK Completed
7 Dlentry Add junk georges
7 OK Completed
8 Dlentry Count junk ""
* 8 6
8 OK Completed
9 Dlentry List junk "" "" ""
* 9 L fred
* 9   georges
* 9   julie
* 9   ralph
* 9   steph
* 9   xerxes
9 OK Completed
10 Logout
10 OK Completed
```

B

Automating Administration with Perl

This appendix documents the Perl `Net::MirapointAdmin` module, which provides a convenient scripting interface to the administration protocol.

Overview

The Mirapoint administration protocol was designed so that system administrators can automate administrative tasks using scripts. Examples of such tasks might be: creating and deleting users, managing quotas, or producing usage reports. The protocol supports the implementation of more ambitious applications, such as complete provisioning systems, report generation tools, complex distribution list management systems, or automated helpdesk attendants.

Perl is a popular programming language for writing system administration scripts and can be readily used to automate Mirapoint administration. The `Net::MirapointAdmin` Perl module provides both *high-level* and *low-level* interfaces to the administration protocol functionality. The high-level interface performs several useful functions, such as tag generation and stripping, correct handling of quoted and literal arguments, and optional response checking. The low-level interface provides direct access to basic functionality, but provides fewer automatic features.

Installing Net::MirapointAdmin

You can install the `Net::MirapointAdmin` module on UNIX-based systems and Windows-based systems. The current version of the module (v3.02) has been tested and runs successfully on Perl version 5.6.1 and higher. The `Net::MirapointAdmin` module is available from CPAN (Comprehensive Perl Archive Network).

UNIX-based Systems

The `Net::MirapointAdmin` module is a standard Perl module. Installation is the same for all UNIX-based systems, including Linux, FreeBSD, Solaris, HP-UX, and the Windows Cygwin UNIX emulation environment. You must perform installation as `root`.

To install the `Net::MirapointAdmin` module using the CPAN interface, type the following command in a root shell:

```
$ perl -MCPAN -e "install('Net::MirapointAdmin');"
```

To install the `Net::MirapointAdmin` module *without* using the CPAN interface, perform the following steps:

1. Download the current version of `Net::MirapointAdmin` from an approved CPAN repository. The most recent version is 3.02, so its downloadable file is named `Net-MirapointAdmin-3.02.tar.gz`.
2. Check that you have `gzip` installed and that you have write permission for the current directory. Unpack the compressed tar file with the following command:

   ```
   $ gzip -dc Net-MirapointAdmin-3.02.tar.gz | tar xvf -
   ```

   Unpacking the tar file creates a directory named `Net-MirapointAdmin-3.02` in the current directory.
3. To build the module, type the following commands:

   ```
   $ cd Net-MirapointAdmin-3.02
   $ perl Makefile.PL
   $ make
   ```
4. Root permission is required to install the module. If you are not using `sudo`, adjust the following instructions to conform to your site's procedures:

   ```
   $ sudo make install
   Password: password
   ```

   The `Net::MirapointAdmin` module is now installed and ready for use with your scripts.

Windows-based Systems

The two most common environments for using Perl on Windows-based systems are:

- the Cygwin UNIX emulation system
- the ActivePerl distribution from ActiveState

Cygwin

To install the `Net::MirapointAdmin` module under the Cygwin UNIX emulation system, follow the instructions for installation on UNIX systems. See UNIX-based Systems on page 685

ActiveState

You can download ActiveState ActivePerl without charge from the ActiveState web site. The Perl Package Manager (PPM), included with every ActivePerl release, is a tool that enables you to manage Perl CPAN modules with ActivePerl. To download and install the `Net::MirapointAdmin` module using PPM, type the following commands:

```
> ppm
ppm> install Net-MirapointAdmin
ppm> exit
>
```

Note that the ActivePerl distribution does *not* support low-level SSL connections. The `Net::MirapointAdmin` module does not support SSL connections when used in the ActivePerl environment.

Method Reference

The `Net::MirapointAdmin` module provides a convenient Perl interface to the Mirapoint administration protocol. The methods provided in this module can be considered either *high-level* or *low-level*, depending upon the degree to which they relieve the programmer of responsibility for managing the handling of tags, arguments, and command responses.

This section assumes you are thoroughly familiar with the syntax of the administration protocol commands, arguments, and responses, as presented in About the Protocol on page 45.

High-Level Interface

The following methods are considered part of the high-level interface:

- `new()`
- `login()`
- `send_command()`
- `get_response()`
- `command()`
- `command_ok()`
- `command_no()`
- other methods

The new() Method

The `new()` method creates a new `MirapointAdmin` object and establishes a TCP connection to the Mirapoint server's administration service. If the method call fails, the method sets the `$!` (`$OS_ERROR`) variable and returns `undef`. The `new()` method has the following syntax:

```
$mp = Net::MirapointAdmin->new( host=>hostname, port=>portnumber,
    exception=>code_ref, debugfunc=>code_ref, debug=>[0|1], ssl=>[0|1] );
```

where:

- `host` is the name of Mirapoint server. The `host` argument is required.
- `port` is the port number of the specified host. Values for the `port` argument are:
  - 10143—clear text connection
  - 10243—SSL connection

The default value is 10143.

- `exception` is a code reference to an exception handler provided by the programmer. The default exception handler simply prints the error message and exits.
- `debug` is a flag to turn TCP trace information on or off. Values for the `debug` argument are:
  - 0 (zero)—Turn TCP trace information *off*. This is the default.
  - 1 (one)—Turn TCP trace information *on*.
- `ssl` is a flag to specify an SSL connection. Values for the `ssl` argument are:
  - 0 (zero)—Request a clear text (non-SSL) connection. This is the default.
  - 1 (one)—Request an SSL connection. The `new()` method returns `undef` if an SSL connection is requested but is not available.

Only the `host` argument to the `new()` method is required. All other arguments are optional.

The login() Method

After establishing the connection to the administration service with `new()`, you must authenticate yourself by logging in. The `login()` method has the following syntax:

```
$mp->login( username, password );
```

where:

- *`username`* is a valid user name
- *`password`* is the password for this user

If the `login()` method detects a dropped connection, it raises an exception, which by default causes the script to exit after printing an error message to standard error. If the `login()` method fails (because of an invalid login name or password, for example), it returns `undef`.

The send_command() Method

The `send_command()` method sends its arguments as a command to the administration protocol. If the command succeeds, the method returns the command tag for use with a subsequent call to the `get_response()` method. The method returns `undef` upon failure. The `send_command()` method has the following syntax:

```
$tag = $mp->send_command( arg1, arg2, ..., argN );
```

The `send_command()` method performs command tagging, and argument packing.

- *Tagging*: Each administration protocol command requires a tag. The `send_command()` method automatically creates a unique protocol command tag and returns that tag for use in collecting the command response.
- *Argument packing*: Arguments that contain whitespace (space, tab) or parentheses must be enclosed in double quotation marks. Arguments that

contain newlines must be converted to IMAP-style counted literals. (See Literal Strings on page 48) The `send_command()` method performs the appropriate quoting and conversion to counted literals. If the only argument to `send_command()` is a single scalar, it is neither quoted nor sent as a literal.

The get_response() Method

The `get_response()` method has the following syntax:

```
$mp->get_response( tag );
```

where *tag* is the tag returned by the last successfully completed command. The response format depends upon whether the `get_response()` method was called in *scalar* or *array* context:

- *Scalar context*: The (possibly multi-line) command response is stripped of tags and returned as a single string, with embedded newline characters delineating individual lines. Counted literals are converted to scalars, but the response is not "dequoted".
- *Array context*: The command response is returned as an array of arrays, indexed by line and by field. (See Responses on page 692 for additional discussion.) Responses are "dequoted," and counted literals are converted to scalars.

In the event of an error, the `get_response()` method saves the cause of the error in the `MirapointAdmin` object and raises an exception. In the absence of a user-defined exception handler, the default behavior is to exit after printing a diagnostic message to standard error.

The command() Method

The `command()` method issues an administration protocol command and returns the command response. Calling the `command()` method is the equivalent of calling the `send_command()` method followed by the `get_response()` method. The `command()` method has the following syntax:

```
@response = $mp->command( arg1, arg2, ..., argN ); # array context
$response = $mp->command( arg1, arg2, ..., argN ); # scalar context
```

The response format depends upon whether the `command()` method was called in scalar or array context. See the discussion of the `get_response()` method for additional information. In the event of an error, the `command()` method raises an exception, which by default causes the script to exit after printing a diagnostic message to standard error.

The command_ok() Method

The `command_ok()` method issues an administration protocol command and performs an automatic check of the success or failure of that command. If the protocol command succeeds, the `command_ok()` method returns the command response; if the protocol command fails, the `command_ok()` method raises an exception, which by default causes the script to exit after printing a diagnostic message to standard error. The `command_ok()` method has the following syntax:

```
@response = $mp->command_ok( arg1, arg2, ..., argN ); # array context
$response = $mp->command_ok( arg1, arg2, ..., argN ); # scalar context
```

The response format depends upon whether the `command_ok()` method was called in scalar or array context. See the discussion of the `get_response()` method for additional information.

The command_no() Method

The `command_no()` method issues an administration protocol command and performs an automatic check of the success or failure of that command. If the protocol command succeeds, the `command_no()` method returns the command response. If the protocol command fails, the `command_no()` method checks to see whether the command response matches the Perl regular expression ("regexp") supplied as the first argument to the method. If the response matches the regular expression, the `command_no()` method returns the command response; if the response does not match the regular expression, the `command_no()` method raises an exception, which by default causes the script to exit after printing a diagnostic message to standard error. The `command_no()` method has the following syntax:

```
@response = $mp->command_no( regexp, arg2, arg3, ..., argN ); # array context
$response = $mp->command_no( regexp, arg2, arg3, ..., argN ); # scalar context
```

The response format depends upon whether the `command_no()` method was called in scalar or array context. See the discussion of the `get_response()` method for additional information.

Other High-Level Interface Methods

The following high-level methods provide access to data stored in the `MirapointAdmin` object:

- `connected()`—The `connected()` method returns 1 (one) if the script is connected to a host, and 0 (zero) otherwise.
- `loggedin()`—The `loggedin()` method returns 1 (one) if the script is successfully logged in and is authenticated.
- `hostname()`—The `hostname()` method returns the name of the host to which the script is connected.
- `reported_hostname()`—The `reported_hostname()` method returns the name of the host to which the script is connected as reported by the Mirapoint system.
- `version()`—The `version()` method returns the version of the Mirapoint protocol running on the host to which the script is connected.
- `mos_version()`—The `mos_version()` method returns the version of the Mirapoint protocol encoded as a hexadecimal number.
- `okno()`—The `okno()` method returns the "OK/NO" response of the last protocol command executed.
- `error()`—The `error()` method returns the last error message generated by the `MirapointAdmin` object.

- `lasttag()`—The `lasttag()` method returns the last tag generated by the object.
- `ssl()`—The `ssl()` method returns 1 (one) if the current connection uses SSL. If the current connection does not use SSL, the `ssl()` method returns 0 (zero).

Low-Level Interface

The following methods are provided for compatibility with an older version of the `MirapointAdmin` module. Most new scripts should use the high-level interface. It is not possible to establish an SSL connection using the low-level interface.

- `xmit()`
- `getbuf()`

The xmit() Method

The `xmit()` method writes a command to the TCP connection, terminated by a CRLF (`\r\n`) sequence. The command is a single string argument comprising an administration protocol command, which must include the required administration protocol *tag*. The `xmit()` method has the following syntax:

```
$mp->xmit( command );
```

If the command is successfully written to the TCP connection, the `xmit()` method returns the number of bytes written to the connection: the length in bytes of *command*, plus two bytes for the terminating CRLF sequence. If the TCP connection no longer exists, the `xmit()` method returns `undef`. If the write to the connection fails, the `xmit()` method raises an exception, which by default causes the script to exit after printing an error message on standard error.

The getbuf() Method

The `getbuf()` method returns one line of the host's response to the last command sent to the host with the `xmit()` method. The `getbuf()` method has the following syntax:

```
$buffer = $mp->getbuf();
```

The `getbuf()` method returns `undef` if it is unable to communicate with the host.

The `getbuf()` method performs no dequoting of the command response, and the return value may not contain the full output of the command executed with the `xmit()` method.

Examples

This section provides sample code for useful tasks using the `Net::MirapointAdmin` module.

Connection and Authentication

The following code establishes an SSL connection to *host* as *user*:

```
$mp = Net::MirapointAdmin->new( host=>host , ssl=>1 );
$mp->login( user, password );
```

To logout, terminate the connection, and destroy the `MirapointAdmin` object (`$mp`), use the command:

```
undef $mp;
```

Commands and Responses

The following section assumes a working knowledge of the syntax of administration protocol commands and responses.

Commands

Issuing administration protocol commands from a Perl script is straightforward with the methods provided by the `Net::MirapointAdmin` module. These methods automatically handle tags, argument quoting, IMAP literals, and other requirements of the administration protocol.

By convention, administration protocol commands and subcommands are enclosed with the Perl "quote word" operator: `qw//`. For example, to obtain a list of users on your Mirapoint system, you would use the `User List` command, which has the following syntax:

```
tag User List pattern start count
```

Assuming that you want to list all users (that is, the the *pattern*, *start*, and *count* arguments are null), you could use the following code:

```
@users = $mp->command( qw/User List/, "", "", "" );
```

Because the `User List` command will succeed in most cases, you might prefer to perform minimal error checking by using the `command_ok()` method:

```
@users = $mp->command_ok( qw/User List/, "", "", "" );
```

Responses

The format of command responses depends upon whether the method was called in *scalar context* or *array context*.

In *scalar* context, the command response is returned as a single string, with multiple-line responses separated by newline characters (`\n`). Response tags are removed, and counted literals are converted to scalars. For example, the following code returns the command response in scalar context:

```
$users = $mp->command( qw/User List/, "", "", "" );
```

If there are two users, "John Smith" (login name "jsmith") and "Jane Doe" (login name "jdoe"), then the value of string `$users` is:

```
"jsmith" "John Smith"\n"jdoe" "Jane Doe"\n
```

In *array* context, each line of the response is converted to an array, and the complete command response is returned as an array of references to these arrays (that is, an "array of arrays"). Response tags are removed, counted literals are converted to scalars, and quoted entities are dequoted. The following code returns the command response in array context:

```
@users = $mp->command( qw/User List/, "", "", "" );
```

(Note the difference between `$users` and `@users`: `$users` is a scalar value, while `@users` is an array.) Assuming the same users for the system as in the previous example, the value of `@users` is:

```
( [ "jsmith", "John Smith" ], [ "jdoe", "Jane Doe" ] )
```

Although the array-of-arrays data structure for command responses might seem complicated at first, this structure facilitates subsequent processing of command responses. The Perl functions `map` and `grep` can be quite useful for such processing. The `map` function performs an operation on each element of a list and returns a list of the transformed elements; the `grep` function evaluates an expression or block for each element of a list and returns a list of those elements for which the evaluation is true. For example, the statement

```
@login_names = map { $$_[0] } @users;
```

would assign the login names `jsmith` and `jdoe` to the array `@login_names`: the expression `$$_[0]` means "select the first element of the array pointed to by the reference in `$_`". The statement

```
@full_names = grep { $$_[1] =~ /D/ } @users;
```

would assign the full name "`Jane Doe`" to the array `@full_names`: the expression `$$_[1] =~ /D/` means "select those second elements of the array pointed to by the reference in `$_` that contain the letter 'D'".

In most cases, familiarity with the expected format of the command response will enable you to manage the returned data successfully. You can use the `Data::Dumper` module (now a standard module in Perl installations) to examine the command response data structure more formally should you need to do so.

Low-Level Interface

Certain tasks require the use of the low-level interface. For example, when obtaining log information on a running system, there is no discrete command "end point" that defines the command response. In such cases, it will be necessary to use the low-level send (`xmit`) and receive (`getbuf`) functions to communicate with the connection.

Because there is no "end point" when monitoring log information, you need to create one to properly terminate the `Log` command in the event of a signal. The following code defines a subroutine named `endit` that terminates the `Log` command in the event of an interrupt or `die` command:

```
sub endit {
    $mp->xmit('Done\r\n');
    die @_;
}

$SIG(__DIE__) = \&endit;
$SIG('INT') = \&endit;
```

The following code runs the `Log Watch` command using the `*.LOGOUT`, `MTA.MESSAGE.RECEIVED`, and `MTA.MESSAGE.LOCAL` routes:

```
@routes = qw(*.LOGOUT MTA.MESSAGE.RECEIVED MTA.MESSAGE.LOCAL);
$mp->send_command(qw/Log Watch/, join(' ', @routes), $last_seqno);
$buf = $mp->getbuf;
if ($buf !~ /^\+idling/) {
    die "Invalid IDLING response from host: $buf\n";
}
```

With the `Log Watch` command initialized, the following loop processes log events:

```
while(defined($buf = $mp->getbuf)) {
    $buf =~ s/^\* //;         # strip off the common start of the line
    $buf =~ s/[\r\n]*$/;      # strip off any line terminators
    ($segno, $host, $origtime, $id, @evt) = split(/\t/, $buf);
}
```

To terminate the script, type control-C. Be sure to save the sequence number (`$seqno`) of the last event processed in order to correctly initialize the next run of the script.

Exceptions

When the methods in the `Net::MirapointAdmin` module encounter an unrecoverable error such as a connection timeout, or a `NO` response from the server, the methods raise an exception. The default exception handler provided by the `Net::MirapointAdmin` module causes the script to exit after printing a diagnostic message to standard error. Because this default behavior might not be the most appropriate action (you might want to attempt reconnection rather than terminating the script, for example), the `Net::MirapointAdmin` module provides the capability of specifying an alternate exception handler that provides the functionality you need.

For example, assume that you have defined a custom exception handler named `&my_handler()`. You instruct the `Net::MirapointAdmin` object to use your custom exception handler by passing a code reference to `&my_handler()` to the `Net::MirapointAdmin` object. This can be done when the object is created using the `new()` method:

```
$mp = Net::MirapointAdmin->new( host => hostname, exception => \&my_handler);
```

You can specify an alternate exception handler at any time with a call to the `exception()` method:

```
$mp->exception(\&my_handler);
```

Debugging

The `Net::MirapointAdmin` module provides a complete protocol trace system to assist in debugging scripts. When the trace functionality is activated, commands from the client are prefixed with "`C: `" and responses from the server are prefixed with "`S: `". By default, trace data is printed to the standard error, although this can be changed by providing a custom debugging function to redirect the data. The trace data for the previous `User List` command

```
@users = $mp->command( qw/User List/, "", "", "" );
```

would appear as follows:

```
C: 2 User List "" "" ""
S: * 2 "jsmith" "John Smith"
S: * 2 "jdoe" "Jane Doe"
S: 2 OK Completed
```

You turn on tracing by calling the `debug()` method with an argument of 1 (one). Although debugging can be specified when the `Net::MirapointAdmin` object is created with the `new()` method, it is recommended that tracing not be turned on until after authentication has taken place, because the protocol trace will show the password sent to the Mirapoint host in the trace log. The following command turns tracing on:

```
$mp->debug(1);
```

To turn tracing off, call the `debug()` method with an argument of 0 (zero). The following command turns tracing off:

```
$mp->debug(0);
```

By default, the debugging function prints the trace data on the standard error. You can specify an alternative debugging function by passing a code reference to the new function as an argument to the `debugfunc()` method. For example, the function defined below, `&my_debug`, prints trace data to a file named `protocol.txt`:

```
$debugfh = IO::File->new("> protocol.txt");
sub my_debug {
    my $msg = join(' ', @_);
    $msg .= "\n" if ($msg !~ /\n$/);
    print $debugfh scalar localtime, ": ", $msg;
}
```

To specify the alternate debugging function, pass a code reference to `&my_debug` as an argument to the `debugfunc()` method:

```
$mp->debugfunc(\&my_debug);
```

You can also specify the alternate debugging function when the `Net::MirapointAdmin` object is created with the `new()` method:

```
$mp = Net::MirapointAdmin->new( host => hostname, debugfunc => \&my_debug );
```

Single Sign-On Portal Access

C

Single sign-on (SSO) allows users to authenticate only once and thereafter access multiple webpages, all of which would otherwise require repeated login. SSO is often used to simplify “portal” entry from an organization’s central webpage to WebMail and WebCal.

With one message server, SSO can be implemented with the unofficial **_Cgi** protocol. In more complex multi-tier environments, it is preferable to use Mirapoint’s XML interface (license required) for SSO, because the **_Cgi** protocol does not support LDAP proxying.

Many examples presented below work in conjunction with the **curl** command, which is available for both Unix and Windows. Curl sends HTTP requests, and makes more sense if spelled cURL.

Single Message Server

In a single-tier environment developers have three options:

1. The administration protocol command **_Cgi Get Sessionid** fetches a session ID for the currently logged-in user account. The returned session ID is effective for subsequent WebMail or WebCal requests sent directly back to that system (no proxy or redirect).
2. The administration protocol command **_Cgi Getsessionidfor *User*** fetches a session ID for the given ***User*** account name. This is called “proxy login” or “three-part login”. Only the administrator, or an authenticated user with the Admin role, is allowed to fetch the session ID on behalf of another user. The returned session ID permits the given ***User*** to sign on, and is effective for subsequent WebMail or WebCal sessions.
3. The **login.xml** command in Mirapoint’s XML interface returns a session ID for the given user and password combination that is supplied in parameters. The returned session ID is effective for subsequent WebMail or WebCal sessions. It is also possible to use **login.xml** for three-part login (proxy login), if security constraints prevent you from obtaining user passwords.

   The following Curl command fetches an XML response for login by a user named **test21** with password **secret89** on system mail1.example.com:

```
curl -d "user=test21&password=secret89" http://mail1.example.com/mc/xml/v1/login.xml
```

   Here is the XML response. You will need to write software to parse the output and obtain the session ID for use in the URL.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE sid SYSTEM "http://mail1.example.com/mc/xml/v1/login.dtd">
<sid>g13af6547c6aac7891f90a7a9be455b121ifjdp5pfl</sid>
```

Here are URLs that your software would send to open a standard WebMail, WebCal, or Corporate Edition session for that user, where ***SID*** is the session ID resulting from **login.xml**—everything between <sid> and </sid> above.

```
http://mailsrv/wm/login.html?sessionid=SID&op=entry&locale=
http://mailsrv/mc/login.html?sessionid=SID&op=entry&locale=
http://mailsrv/wm/eml/login.html?sessionid=SID&op=entry&locale=
```

Of course, the requirement here is that you have access to the user's plaintext password, which is not always the case. Keep reading for information on how to use XML for three-part login.

Multi-Tier Environments

In multi-tier environments, with a layer of one or more RazorGate appliances routing to one or more message servers, things get more complex. There are two possibilities for multi-tier SSO:

1. RazorGate doing LDAP redirect (**Http Set Mode Ldapredirect**)

   When a RazorGate appliance is in LDAP redirect mode, message servers must be exposed to the network because users connecting to a RazorGate have their web sessions redirected to the message server with an HTTP 303 URL redirect. Users touch the RazorGate only once, for authentication. After that, subsequent requests go directly to and from the message server.

   This is the easier multi-tier case, because developers can still get to the message server and use either of the **_Cgi** commands described above for single-tier. You just need to determine on which message server the user account resides, based on the LDAP record. And you must generate the session ID by running the **_Cgi** command on that message server, otherwise it will fail.

2. RazorGate doing LDAP proxy (**Http Set Mode Ldapproxy**)

   In a typical multi-tier environment, RazorGate appliances are running in LDAP proxy mode so all the message servers can be concealed behind a firewall or within a private network.

   When RazorGate appliances are running in LDAP proxy mode, the only way to generate a session ID for SSO is by using the XML interface, connecting to the edge RazorGate appliances.

   Note: The session ID given out by the message server does not work on the RazorGate because in LDAP proxy mode, the RazorGate rewrites the message server's session ID, adding information to it. The RazorGate then uses this extended session ID for proxying. This has several implications:

   - It reinforces the need to employ the RazorGate layer for portal access session ID fetch. You cannot fetch session IDs from a message server and expect to work if the RazorGate layer is in LDAP proxy mode.
   - You don't need to know on which message server a user resides, because you can fetch the session ID using any RazorGate in LDAP proxy mode with XML licensed.

- RazorGate appliances build some useful information into the session ID that affords great functionality. Specifically, if you are behind a load balancer, you needn't worry about keeping all sessions routing through to the same edge RazorGate for every request. Instead, Mirapoint-based session requests work with any type of load balancer.

The **login.xml** command in Mirapoint's XML interface returns a three-part login (proxy login), when given a valid login/password combination of a user with Admin role, and a normal user (called **caluser**) for whom to obtain the session ID.

Implementing Multi-Tier SSO

The following procedure requires XML licenses on both RazorGate and message servers, and WebCal licenses on message servers.

Using XML, the following Curl command obtains a session ID for user **joe** on the message server **mail2**. Note that "caluser" is the keyword accepted by **login.xml** for WebCal group calendar XML. The Admin role has been added for a hypothetical user named **sso2**, whose password is **xyz** (bad password choice):

```
curl -d "user=sso2&password=xyz&caluser=joe" http://mail2.example.com/mc/xcal/v1/login.xml
```

The user for authentication, in this case **sso2**, should exist in LDAP (LDAP lookup must work for this user) and on the message server. Admin role means to run the **Role Add Admin sso2** command after creating the user. The regular Administrator should not be used for SSO management because the password here is semi-exposed by HTTP; using a name like **sso2** lends obscurity.

The RazorGate LDAP proxy is not able to resolve the **sso2** user locally, so it performs an LDAP lookup to get the mailhost for this user. In this case this is **mail2** (let's say mail2.example.com is the fully qualified host name). After LDAP lookup is successful, the RazorGate proxies the login request to the message server.

The message server receives the **login.xml** request and authenticates the **sso2** user successfully from LDAP. Because this user has the Admin role, XML perform a three-part proxy login for **caluser** Joe.

The message server generates a session ID, which it returns to the RazorGate proxy. This session ID contains additional information required for the proxy to route further requests.

Be sure to observe the following configuration requirements:

- All participating appliances should have default authentication set to **plaintext:ldap**, not local or Kerberos.
- Do not use Administrator on the RazorGate for authentication. Three-part proxy login must be authenticated by an LDAP-listed user with Admin role on the mailhost (message server).
- The proxied **caluser**, in this case Joe, must also be LDAP listed.
- LDAP should be available at both layers, RazorGate and mailhost.
- On RazorGate appliances, run the **Http Set Mode Ldapproxy** command. HTTP cookies should be disabled, which they are by default.

- On message servers, the **Http Get Mode** command should return **Normal**, not **Ldapproxy**. Cookies should be disabled. The **Role List** command should return Administrator and the SSO management user, in this example **sso2**.
- XML and Group Calendar must be licensed on the message servers.

The Curl command below, in bold, returns the HTTP response given below, not in bold:

```
curl -d "user=sso2&password=xyz&caluser=joe" http://mail2.example.com/mc/xcal/v1/login.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE login SYSTEM "http://mail2.example.com/dtd/xcal/v1/login.dtd">
<login>
  <ok>OK</ok>
  <sid>g43770cc5f811221d661c6119ccfad7cf6ifjdp5oal</sid>
</login>
```

The returned session ID contains routing information, which the proxy uses to locate the mailhost (message server).

Now your software can construct the appropriate URL to which your portal entry point should redirect the user. Substitute the session ID returned from **login.xml** (everything between <sid> and </sid> in the example above) for **$SID** in the URLs below.

For Corporate Edition WebMail/WebCal:

```
http://mail2/wm/eml/login.html?sessionid=$SID&op=entry&locale=
```

For WebMail standard edition:

```
http://mail2/wm/login.html?sessionid=$SID&op=entry&locale=
```

For WebCal standard edition group calendar:

```
http://mail2/mc/login.html?sessionid=$SID&op=entry&locale=
```

The **op=entry** and **locale=** parameters are designed so that the setting for cookies and prevailing locale can be automatically negotiated.

A
AcceptPercent 540
AcceptUucp 540
Activebackup 446
Activefailback 446
Admin 512
Adminmessage 64
AllowedEntryLifetime 416
AllowEntireIP 416
AllowMisbehavingMailers 417
AllowNullFrom 417
AllowRelays 417
Antispam 122, 133
Antivirus 133
Antivirus-Fsav 122
Antivirus-Rapid 122
Antivirus-Sophos 122
Arpmon 446
Arpmonaddr 446
Attachenabled 108
Attachquota 108
Auth 466, 666
Autoprovision 321
Autoreplies 222
Autoreply 133
B
Backup 512
Bannerdelay 540
Bannerdelaytime 540
Bbstatus 612
Bbwritethru 612, 618
Bbwritethruthresh 612, 618
BindDN 187
BindPW 187
BlackholeDuration 444
Broadcast 371
C
Cachetimeout 322
Calendar 133
Calendar.maxnumevents 222
Calendar.removeafter 222
Calendar.timeout 222
Catchall 222
Chaining 166
ChangerAddress 143
Checksuspectlist 417

Cleartext 58, 274, 278, 467
Cleartextout 58, 274, 278, 467
Color 263
Compatv1routing 322
Connectport 540
Consolelogin 122
Controller 612
Cookies 272
Cryptpass 666
Customercare 122
Cyrus 277
D
Dataxferelementaddr 143
Date 457
Daygranularity 108
Default 338
Defaultsendoverquotamessage 472
Degraded 441
Deliveryaddress 666
Denylogin, an miException 124
DerivedomainIP 122
Derivedomainurl 123
Detach 607
Dfltcalflags 108
Dfltdayend 108
Dfltdaystart 108
Dfltuiflags 108
Dfltview 109
Dironly 426
Discard 245
Disclaimer 245
Diskquota 222
Diskvendor 612
Dls 222
Dma 426
Dsndisable 541
Dsnlimit 541
Dsnretrycnt 541
E
Empty 441
Encodedpass 666
EnterpriseUi 123, 133
Exclude 187
Exemptonspf 417
Expandlun 607
Externalmaxnum 109

Externalmaxsize 109
F
Failed 441
Failover 446
Fastpath 541
Fileinto 245
Filter 133
Filtering 123, 263
Forward 133
Fullname 666
Fwdexcerpt 245
Fwds 222
G
Getmail 123, 133
Groupcalendar 133
Groupcalmode 109
Grow 607
Growforce 607
H
Headerinfo 644
Helpdesk 512
History 358
Httpproxy 123
I
Idecache 612, 618
IdleTimeout 166
Ignorerecipients 417
Imap 133
Imaplistonlyinbox 123
Imodemail 133
InboundOnly 417
Init 607
Initforce 607
InitialEntryLifetime 417
InitialTimeout 417
J
Junkmailmanager 123
Junkmailsummary 666
Junkmailsummaryemptylogs 666
K
Keep 245
Keeptomailhost 123
Knownzones 457
L
Ldapall 123
Ldapautoreply 123

Ldapexception 123
Ldapforward 124
Ldapgui 124
Ldapgui-jmm 124
Ldapmasqsender 542
Ldapmaxcount 109
Ldapmxrouting 542
Ldappassupdate 124
Ldapproxy 277
Ldaprouteurls 272
Ldaprouting 542
Ldaptimeout 109
Ldapuuid 124
Ldapwmprefs 124
Ldif 322
Leaveonserver 263
LimitTcpConnectCount 444
LimitTcpConnectRate 444
Linkdowndelay 446
Linkmon 446
Linkupdelay 446
Listenport 543
Lmr 543
Localcostable 322
Login 666
LoginbeforeSmtp 543
LoginbeforeSmtpServer 543
LoginbeforeSmtpTime 543
Loginfooter 338
Logprotocol 426
M
Mailbox 263
Mailreturn 543
Mailwarn 543
Markinterval 358
Masq 543
masquerade, definition 543
Master 187
Maxconnections 272
Maxdirectorytraversals 272
Maxheaderlength 272
Maxmsg 543
Maxmsgcnt 544
Maxnumevents 109
Maxrecip 544
Maxrequestbodysize 273

MaxTcpConnectCount 444
MaxTcpConnectRate 444
Maxtotalheadersize 273
Media 444
Merged 277
Minpoll 269, 467
Mirabase 322
Mode 273, 277, 467
Modspamscore 245
Monthlysummaryday 109
Monthlysummarytime 109
Msgexpiration 133
Msgundelete 134
Mtaverify 544
Multienginebulkonly 644
N
Namespace 277
Net::MirapointAdmin (Perl module) 685–695
Newmessagesonly 264
Nomasquerade 544
Normal 277
Notifyadmin 64
Notifyrecipient 64
Notifysender 64
O
Omr 544
Optimal 441
Overquota 471
Overquotamessage 472
P
Pagerreminderdiff 109
Partial 441
Pass 666
Password-hash 157
Pop 134
Port 264, 427
Portwwn 612, 614
PreferredIp 427
Preservecnames 544
Primaryport 444
Proxyauth 166
Proxycapability 278
Proxytimeout 273
Q
Quarantine 245, 512
Quarantineaddress 64

Quarantinedelay 65
Quota 134
Quotawarn 278
R
Rbl 544
Rblhandling 544
RdnIntegrity 166
ReadOnly 166
Realm 284, 291
Reattach 607
Rebuildrate 612, 619
Receivedforhdr 544
Recipientcheck 545
Recipientmessage 65
Redirect 245
Referral 166
RefOnly 166
Reject 245
Reminderdiff 109
Removeafter 109
Reporting 644
Reputation 124
Resourcegroup 109
ReverseMx 418
Root 273
RootDN 166
RootPW 166
S
Scanlist 65
Schedule 187
Schedulemode 109
SchemaCheck 166
Secret 476
Secureauthonly 278, 467, 545
Security 58, 157, 273, 278, 467, 545
Sender_as 134
Sender_av 134
Sendercheck 545
Senderisauth 545
Senderisvalidrecipient 546
Sendermessage 65
Sendoverquotamessage 472
Sendusernameastyped 476
Signature 222
SizeLimit 167
Smtpauth 546

Softwareversion 424
Spamprolog 644
Spamprologtext 644
Spf 546
Srvtab 284, 291
Ssh 58
Ssl 58, 134, 274, 278, 467, 545
Sslout 58, 274, 278, 467, 545
Standby 607
Starttls 278, 467
Starttlsout 278, 467
Storage 471
Storetype 612
Subdomainroutinglevel 322
Summarytimeemail 109
Summarytimepager 110
SuspectlistThreshold 644
Syncinterval 359
Syscontact 557
Syslocation 557
T
TapeAddress 143
TapeCompression 143, 146
TapeCompressionRatio 143
Threshold 644
TimeLimit 167
Timeout 58, 110, 279, 468, 476, 670
Trigger 188
Trustfarmmembers 547
U
UceBlacklist 547
UceSuspectlist 547
UceWhitelist 547
UceWhitelistIp 547
Unannounced 338
Undeletequota 372
Uninitialized 441
Userdn 323
UserIDtoDN 157
Userquota 222
UW namespace, IMAP 277
V
Verifyreboot 124
Version 427
version 1.2.4 protocol 43
version 1.2.5 protocol 43

version 1.3 protocol 43
version 1.4 protocol 43
version 1.5 protocol 42
version 1.6 protocol 42
version 2.0 protocol 42
version 2.5 protocol 41
version 2.7 protocol 40
version 2.8 protocol 40
version 2.8.1 protocol 40
version 2.9 protocol 39
version 2.9.1 protocol 39
version 2.9.2 protocol 39
version 2.9.3 protocol 38
version 3.0 protocol 39
version 3.1.0 protocol 38
version 3.1.1 protocol 38
version 3.2 protocol 37
version 3.3 protocol 37
version 3.4 protocol 36
version 3.5 protocol 35
version 3.6 protocol 34
version 3.7 protocol 34
version 3.8 protocol 33
version 3.9 protocol 32
Virusaction 65
W
Wapcalendar 134
Wapmail 134
Webmail 134
Weeklysummaryday 110
Weeklysummarytime 110
Wiretap 245
Wmaddrbook 666
Wmdictionary 666
Wmprefs 667
Z
Zone 457