Alcatel·Lucent

VitalQIP® DNS/DHCP & IP Management Software

VitalQIP® (QIP) | Release 7.2

Command Line Interface User's Guide

190-409-044R7.2
July 2009
Issue 4

Alcatel, Lucent, Alcatel-Lucent and the Alcatel-Lucent logo are trademarks of Alcatel-Lucent. All other trademarks are the property of their respective owners..

The information presented is subject to change without notice. Alcatel-Lucent assumes no responsibility for inaccuracies contained herein.

Copyright © 2009 Alcatel-Lucent. All Rights Reserved.

Trademarks

All trademarks and service marks specified herein are owned by their respective companies.

License statement

Refer to Appendix C, “Third party software license statements” in the VitalQIP Release 7.2 Installation Guide (190-409-043R7.2) for a complete description of all software licenses used to develop this product.

Contents

About this document

Purpose xv

Reason for reissue xv

How to use this information product xx

Conventions used xxi

Related information xxi

Technical support xxi

How to order xxii

How to comment xxii

1 VitalQIP database functions

Issuing CLI commands 1-3

Notation key 1-4

Common arguments/parameters 1-5

Using the CLI commands

qip-active 1-8

qip-admin 1-12

qip-altersubnet 1-29

qip-bootpgen 1-30

qip-changepassword 1-31

qip-check 1-32

qip-checkobjname 1-33

qip-clear 1-34

qip-crypt 1-35

qip-dbinit 1-36

qip-del 1-38

qip-delacltemplate 1-41

qip-deladminrole 1-42

qip-delpolicy 1-43

qip-dhcpdebuglog 1-44

qip-dhcpdump 1-45
qip-dhcpgen 1-46
qip-dhcpsync 1-48
qip-dnsgen 1-50
qip-dnsupdate 1-52
qip-folder 1-54
qip-genddnsconfs 1-56
qip-getacltemplate 1-57
qip-getactiveobjectlst (for use with AutoDiscovery) 1-59
qip-getaddrrangelst 1-63
qip-getadminrole 1-65
qip-getapplst 1-66
qip-getbtpsvrlst 1-67
qip-getclientclass 1-68
qip-getcontactlst 1-69
qip-getdebuglevel 1-72
qip-getdecnetaddr 1-73
qip-getdhcpscopes 1-74
qip-getdhcpsubnetlst 1-76
qip-getdhcpsvrlst 1-77
qip-getdnsrr 1-78
qip-getdnssvrlst 1-80
qip-getdomnlst 1-82
qip-getfolderlst 1-83
qip-getfreesubnetlst 1-84
qip-gethublst 1-86
qip-gethubport 1-87
qip-gethubslotlst 1-88
qip-getipaddr 1-89
qip-getlocalsvrlst 1-90
qip-getloclst 1-91
qip-getmaclst 1-93
qip-getmacmodellst 1-94

qip-getmacpools 1-96
qip-getnetlst 1-97
qip-getnissvrlst 1-98
qip-getnmdnsserver 1-99
qip-getobjectlst 1-101
qip-getobjectprof 1-103
qip-getobjname 1-110
qip-getorganization 1-111
qip-getospflst 1-112
qip-getospfprof 1-113
qip-getpolicy 1-115
qip-getprimdnssvrlst 1-116
qip-getrevzonelst 1-117
qip-getrtrlst 1-118
qip-getsecdnssvrlst 1-119
qip-getsnaddr 1-120
qip-getsnorglst 1-121
qip-getsnorgprof 1-122
qip-getsubnetlst 1-124
qip-getsubnetprof 1-126
qip-gettemplate 1-130
qip-gettemplst 1-132
qip-gettimesvrlst 1-133
qip-getudflst 1-134
qip-getudfnamelst 1-137
qip-getuser 1-138
qip-getuseraddrlst 1-140
qip-getzoneext 1-141
qip-getzoneprof 1-144
qip-globalmacpool 1-148
qip-hndbgen 1-150
qip-hostgen 1-151
qip-leasefilegen 1-152

qip-mcancel 1-154
qip-move 1-156
qip-msextract 1-158
qip-namingpolicy 1-159
qip-objectclass 1-162
qip-postallocation 1-165
qip-qdhcplease 1-170
qip-reclaim 1-171
qip-report 1-174
qip-rot13 1-178
qip-rptadminaudit 1-179
qip-rptadminrole 1-180
qip-rptaudithistory 1-181
qip-rptdhcp 1-183
qip-rptfreesubnet 1-187
qip-rptinquire 1-188
qip-rptmanaged 1-190
qip-rptobjectlst 1-191
qip-rptrole 1-194
qip-rptzonerr 1-195
qip-scope 1-197
qip-search 1-200
qip-searchacltemplates 1-205
qip-setacltemplate 1-206
qip-setadminrole 1-207
qip-setclientclass 1-209
qip-setcontact 1-211
qip-setdnsrr 1-213
qip-setdomainfolder 1-216
qip-setlocation 1-217
qip-setmacpools 1-219
qip-setnmdnsserver 1-220
qip-setobject 1-223

qip-setospfprof 1-233
qip-setpolicy 1-234
qip-setreclaimschedule 1-236
qip-setsnorgprof 1-238
qip-setsubnet 1-240
qip-setudf 1-246
qip-setuser 1-248
qip-setzonednsoptions 1-250
qip-setzoneext 1-256
qip-setzoneprof 1-259
qip-setzoneserveroptions 1-264
qip-sitegen 1-266
qip-siteimport 1-267
qip-splitmergeenum 1-269
qip-splitrevzone 1-271
qip-subnetmacpool 1-272
qip-syncexternal 1-274
qip-template 1-277
qip-tombstonepurge 1-281
qip-udf2uda 1-282
qip-ungetdecnetaddr 1-284
qip-ungethubport 1-285
qip-ungetipaddr 1-286
qip-unlock 1-287
qip-util 1-288
vercheck 1-293

2 Export and import whole databases

qip-export 2-2
qip-import 2-4

3 Exporting and importing DNS and Bootptab files

Exporting DNS files 3-2
qip-dnscsv 3-5
Using the “Rules” file with qip-dnscsv 3-8

qip-dnscsv output files 3-13
Importing DNS files 3-18
Exporting Bootptab files 3-19
qip-bootptabcsv 3-20
Using qip-bootptabcsv 3-21
qip-bootptabcsv output files 3-22
Importing Bootptab files 3-26

4 Exporting and importing with CLIs

Importing files

Order of importing 4-2
enteraddrrange 4-3
enteralias 4-5
entercontact 4-7
enterdnsobj 4-9
enterdnsrr 4-11
enterdomain 4-15
enterlocalobj 4-22
enterlocation 4-24
entermanufacturer 4-26
enternetwork 4-28
enterorganization 4-30
enterospf 4-32
enterreversezone 4-34
enterserver 4-42
entersimpleobj 4-47
entersubnet 4-52
entersubnetorg 4-55
enterudf 4-59
enteruser 4-61
enterusergrp 4-63
enterzoneext 4-64
qipbulkload 4-66
qiploadsndomn 4-72

Exporting files with CLIs

exportaddrrange 4-75
exportcontact 4-77
exportdnsrr 4-78
exportdomain 4-80
exportlocation 4-82
exportmanufacturer 4-83
exportnetwork 4-84
exportorganization 4-85
exportospf 4-86
exportreversezone 4-87
exportserver 4-88
exportsimpleobj 4-89
exportsubnet 4-91
exportsubnetorg 4-93
exportudf 4-94
exportuser 4-95
exportusergrp 4-96
exportzoneext 4-97

5 Web Client CLIs

About Web Client CLIs 5-2
Address Management and Address Allocation 5-12
acdevice, add 5-13
acdevice, modify 5-16
acdevice, delete 5-18
acdevice, get 5-19
acdevice, search 5-21
acsubscriber, add 5-24
acsubscriber, modify 5-27
acsubscriber, delete 5-29
acsubscriber, get 5-30
acsubscriber, list 5-32
acsubscriber, search 5-34

addressrange,add 5-37
addressrange,delete 5-39
addressrange,get 5-40
addressrange,modify 5-43
amseedblock,add 5-45
amseedblock,delete 5-47
amseedblock,get 5-48
block,addtopool 5-50
block,allocate 5-52
block,deletefrompool 5-55
block,expand 5-56
block,free 5-57
block,freepending 5-58
block,get 5-60
block,merge 5-63
block,modify 5-64
block,move 5-66
block,recover 5-67
block,renumber 5-68
block,return 5-70
block,search 5-71
block,split 5-78
cdnssec, add 5-79
cdnssec, delete 5-81
cdnssec, modify 5-82
cdnssec, get 5-84
cserver, copy 5-86
dnsacltemplate,add 5-89
dnsacltemplate,copy 5-91
dnsacltemplate,delete 5-92
dnsacltemplate,get 5-93
dnsacltemplate,modify 5-95
dnsacltemplate,search 5-96

dnsview,add 5-98
dnsview,delete 5-99
dnsview,get 5-100
dnsview,list 5-103
dnsview,modify 5-105
dnsview,optionreport 5-106
dnsview,search 5-109
dnsview,sequence 5-112
info,dnsoptions 5-114
info,get 5-115
job,cancel 5-119
job,delete 5-120
job,get 5-121
job,search 5-124
list,search 5-128
maintainer,add 5-131
maintainer,delete 5-133
maintainer,get 5-134
maintainer,modify 5-137
maintainer,search 5-139
managedfile,add 5-142
managedfile,associate 5-146
managedfile,delete 5-147
managedfile,get 5-148
managedfile,modify 5-150
managedfile,search 5-153
node,add 5-157
node,delete 5-160
node,get 5-162
node,link 5-166
node,modify 5-167
node,move 5-169
node,search 5-171

node,unlink 5-179
object,fqdnadd 5-180
object,fqdndelete 5-182
object,get 5-183
object,modify 5-185
pool,add 5-186
pool,delete 5-188
pool,get 5-189
pool,merge 5-193
pool,modify 5-194
pool,search 5-196
registry,get 5-202
registry,modify 5-205
rule,add 5-207
rule,copy 5-209
rule,delete 5-210
rule,get 5-211
rule,modify 5-214
rule,search 5-216
subnet,add 5-219
subnet,delete 5-222
subnet,get 5-223
subnet,merge 5-227
subnet,modify 5-228
subnet,renumber 5-230
subnet,search 5-232
subnet,split 5-239
template,add 5-241
template,copy 5-243
template,delete 5-244
template,get 5-245
template,modify 5-248
template,search 5-250

uda, add 5-253
uda,associate 5-255
uda, countassociations 5-256
uda, delete 5-257
uda, dissociate 5-258
uda, findassociations 5-259
uda, get 5-260
uda, list 5-262
uda,modify 5-264
uda,search 5-266
uda, sequence 5-267
uda,set 5-269
udagroup, add 5-274
udagroup,associate 5-276
udagroup, countassociations 5-277
udagroup, delete 5-278
udagroup, dissociate 5-279
udagroup, findassociations 5-280
udagroup, get 5-281
udagroup,modify 5-283
udagroup,search 5-285
zone,add 5-288
zone,delete 5-290
zone,get 5-292
zone,modify 5-294
zone,optionreport 5-296
zone,search 5-300
zone,share 5-304
zone,unshare 5-306
zoneserver,add 5-308
zoneserver,delete 5-309
zoneserver,get 5-310
zoneserver,modify 5-312

zoneserver,optionreport 5-313
zoneserver,search 5-317

IN Index

About this document

Purpose

Welcome to VitalQIP® – a powerful IP name and address management tool. VitalQIP simplifies the assignment and allocation of IP addresses and services, such as DHCP and DNS. This product is a comprehensive collection of management tools and user interfaces. Each management tool and user interface provides the ability to plan, manage, and locally administer IP addresses and services across UNIX, Linux, and Windows platforms. VitalQIP works with directory services and RDBMS database configurations.

Reason for reissue

The following table lists the changes to the VitalQIP GUI that required the *Command Line Interface User's Guide* to be reissued.

Issue	Feature Name	Description	Feature Impact
4	Client ID information	The **`qip -active`** CLI now displays information about the Client ID.	"Output example 1" (p. 1-9)
4		Formatted header levels. Fixes VQIP00020477.	• "qip-admin input file" (p. 1-15) • "Comma delimited file format" (p. 1-20) • "Name value pairs file format" (p. 1-21) • "To add a managed list" (p. 1-26)

Issue	Feature Name	Description	Feature Impact
4	**-e** option to **qip-del** CLI	Documented the **-e** option for the **qip-del** CLI. Fixes VQIP00020491 and VQIP00020625.	"qip-del" (p. 1-38)
4		The **qip-del** CLI now handles the deletion of domain controller server type.	Table 1-6, "Owner types" (p. 1-39)
4		Added a note for the qip-export CLI. Fixes VQIP00020581.	Note on page 2-3
4	Updates to **qip -leasegen** CLI	The Client id is displayed in the tenth column of the output.	"Output example for DHCP 5.5 server" (p. 1-152) and "Output example for DHCP 5.6 server" (p. 1-152)
4		Added a note for the **-b** option of the **qip-dnscsv** CLI. Fixes VQIP00022153.	"-b 9" (p. 3-7)
4	enterserver support for Lucent DHCP 5.6	Added LUCENT DHCP 5.6 server type and deleted LUCENT DNS 3.X and BIND-8.X DNS server types.	"enterserver" (p. 4-42)
4	exportserver support for Lucent DHCP 5.6	Removed LUCENT DNS 3.X from the server types.	"exportserver " (p. 4-88)
4		Modified the date format for the qip-cli command wherever the "--when" option was specified. Fixes VQIP00021933.	"block,renumber" (p. 5-68)
4	CLIs to add, modify, delete, get, and search AC devices	New CLI commands are available for adding, modifying, deleting, getting, and searching AC devices.	• "acdevice, add" (p. 5-13) • "acdevice, modify" (p. 5-16) • "acdevice, delete" (p. 5-18) • "acdevice, get" (p. 5-19) • "acdevice, search" (p. 5-21)

Issue	Feature Name	Description	Feature Impact
4	CLIs to add, modify, delete, get, list, and search AC subscribers	New CLI commands are available for adding, modifying, deleting, getting, listing, and searching AC subscribers.	• "acsubscriber, add" (p. 5-24) • "acsubscriber, modify" (p. 5-27) • "acsubscriber, delete" (p. 5-29) • "acsubscriber, get" (p. 5-30) • "acsubscriber, list" (p. 5-32) • "acsubscriber, search" (p. 5-34)
4	CLI to clone DNS and DHCP servers	A new CLI command is now available to clone DNS and DHCP servers.	"cserver, copy" (p. 5-86)
4	CLIs to work with the DNSSEC feature, that provides data origin authentication, data integrity, and authenticated denial of existence for DNS data by securing it with public and private keys and establishing chain of trust.	New CLIs are now available to: • Add the key for the zone name specified in the inputfile to VitalQIP database. • Delete the keys specified in the input file from the VitalQIP database. • Modify the key data for the key name given in the input file or command line option. • Retrieve the DNSSEC key for given keynames.	• "cdnssec, add" (p. 5-79) • "cdnssec, delete" (p. 5-81) • "cdnssec, modify" (p. 5-82) • "cdnssec, get" (p. 5-84)
4	CLIs to associate UDA and UDA groups	New CLIs are now available to associate UDA and UDA groups with different infrastructure types.	• "uda,associate" (p. 5-255) • "udagroup,associate" (p. 5-276)
4	CLIs to dissociate UDA and UDA groups	New CLIs are now available to dissociate UDA and UDA groups from different infrastructure types.	• "uda, dissociate" (p. 5-258) • "udagroup, dissociate" (p. 5-279)

Issue	Feature Name	Description	Feature Impact
4	CLIs to view the infrastructure types associated with UDA and UDA groups	New CLIs are now available to view infrastructure types associated with UDA and UDA groups.	• "uda, countassociations" (p. 5-256) • "udagroup, countassociations" (p. 5-277)
4	CLIs to view the infrastructure types associated with UDA and UDA groups using the output scheme LIST_REC	New CLIs are now available to view the infrastructure types associated with UDA and UDA groups using the output scheme LIST_REC.	• "uda, findassociations" (p. 5-259) • "udagroup, findassociations" (p. 5-280)
4	CLIs to: • View the UDA_DEFINITION_LIST_REC containing a list of UDA and UDA groups associated with the specified infrastructure types. • Impose a specific ordering to UDAs and UDA Groups already associated with an infrastructure type.	New CLIs are now available to: • View the UDA_DEFINITION_LIST_REC containing a list of UDA and UDA groups associated with the specified infrastructure types. The output format is only in XML • Impose a specific ordering to UDAs and UDA Groups already associated with an infrastructure type. The CLIs read the input file containing UDA_DEFINITION_LIST_REC with the new sequence of UDA and UDA groups already associated with the infrastructure type.	• "uda, list" (p. 5-262) • "uda, sequence" (p. 5-267)
4	CLI to set specific UDA values for a variety of QIP infrastructure types	A new CLI is now available to set specific UDA values for a variety of QIP infrastructure types.	"uda,set" (p. 5-269)
4		Added information for the "Input file name" and documented the steps to execute the **entersubnetorg** CLI. Fixes VQIP00019731.	"entersubnetorg " (p. 4-55)

Issue	Feature Name	Description	Feature Impact
3	AFRINIC RIR support	AFRINIC support has been added to all CLIs that reference Internet Registries. The CLIs listed in the Feature Impact column have the AFRINIC registry as a parameter value of their syntax.	"maintainer,delete" (p. 5-133), "maintainer,get" (p. 5-134), "maintainer,search" (p. 5-139), "registry,get" (p. 5-202)
3	qip-setsubnet	New -t parameter added to qip-setsubnet	"qip-setsubnet" (p. 1-240)
3	DHCP 5.5 Support	VitalQIP now suports DHCP 5.5 servers.	"enterserver" (p. 4-42)
3	block,recover	New CLI	"block,recover" (p. 5-67)
2	Editorial Changes	Minor editorial changes.	Throughout document.
1	qip-setzoneserveroptions	New CLI	"qip-setzoneserveroptions" (p. 1-264)
1	qip-udf2uda	New CLI	"qip-udf2uda" (p. 1-282)
1	Web Client CLIs	New and enhanced descriptions of Web Client CLIs.	"Automated CLI Options" (p. 5-4), "XML input and output" (p. 5-5), "XML File Structure" (p. 5-6)
1	DNS ACL Templates	New CLIs to control ACL Templates	"dnsacltemplate,add" (p. 5-89), "dnsacltemplate,search" (p. 5-96)
1	DNS View	New CLIs to control DNS Views	"dnsview,add" (p. 5-98) to "dnsview,sequence" (p. 5-112)
1	Managed Files	New CLIs to control managed files.	"managedfile,add" (p. 5-142) to "managedfile,search" (p. 5-153)

Issue	Feature Name	Description	Feature Impact
1	Objects	New CLIs to control objects	"object,fqdnadd" (p. 5-180) to "object,modify" (p. 5-185)
1	Zones and Zoneservers	New CLIs to control zones and zone servers	"zone,add" (p. 5-288) to "zoneserver,search" (p. 5-317)
1	qip-setreclaimschedule	Revised example	"qip-setreclaimschedule" (p. 1-236)

How to use this information product

This manual is organized as follows:

Chapter 1, "VitalQIP database functions"	This chapter outlines CLI command usage and explains the parameters that are common to all CLI commands. It also explains the notation used in explanations of the command parameters. Chapter 1 also explains all the CLI commands (in alphabetic sequence) except those used for exporting and importing information.
Chapter 2, "Export and import whole databases"	This chapter explains how to use qip-export and qip-import to export and import entire VitalQIP databases. These CLI commands are used primarily for upgrades or backup/recovery purposes.
Chapter 3, "Exporting and importing DNS and Bootptab files"	This chapter explains how to use the export/import utilities to extract current BIND 4.9.x, BIND 8.x, or bootp data, and import it into your VitalQIP system. These utilities can be used for upgrades or new installations
Chapter 4, "Exporting and importing with CLIs"	This chapter explains how to use the VitalQIP CLI commands to export and import data to/from the VitalQIP database. The commands allow you to export data for domains, OSPF areas, subnet organizations, subnets, MAC address pools, or objects. They also allow you to import data for domains, OSPF areas, subnet organizations, subnets, MAC address pools, or objects. These CLI commands can be used for upgrades or new installations.

Chapter 5, "Web Client CLIs"	This chaper describes how to use Web Client CLIs. These CLIs allow you to perform functions that you typically perform with the VitalQIP Web Client.

Conventions used

Refer to "Notation key" (p. 1-4) for the typographical conventions used in chapters 1 through 4 of this manual. Refer to "Typographical Conventions" (p. 5-10) for the typographical conventions used for Web Client CLIs, which are located in chapter 5.

Related information

The following documents are referenced in this manual:

- *VitalQIP Administrator Reference Manual* (part number: 190-409-042R7.2)

  This guide describes planning and configuring your network, information about the VitalQIP interface, advanced DNS and DHCP configurations, and troubleshooting.
- *VitalQIP Installation Guide* (part number: 190-409-043R7.2)

  This guide describes how to install the VitalQIP product.
- *VitalQIP Web Client User's Guide* (part number: 190-409-079R7.2)

  This guide describes how to use the web client interface.

Training Support

Alcatel-Lucent University offers cost-effective educational programs that support the VitalQIP product. Our offerings also include courses on the underlying technology for the VitalQIP products (for example, DNS and DHCP). Our classes blend presentation, discussion, and hands-on exercises to reinforce learning. Students acquire in-depth knowledge and gain expertise by practicing with our products in a controlled, instructor-facilitated setting. If you have any questions, please contact us at 1 888 LUCENT8, option 2, option 2.

Technical support

If you need assistance with VitalQIP, you can contact the Technical Assistance Center for your region. Contact information is provided in the following table.

Table 1 Technical support information

Region	Address	Contact information
North, Central, and South America	Alcatel-Lucent 400 Lapp Road Malvern, PA 19355 USA	Phone: 1-866-LUCENT8 (582-3688) Option 1, Option 2 Web: **https://support.lucent.com**

Region	Address	Contact information
Europe, Middle East, Africa, and China	Alcatel-Lucent Chiltern House Sterling Court Broad Lane Bracknell, RG12 9GU UK	Phone: 00 800 00 LUCENT or +353 1 692 4579 E-mail: *emeacallcenter@alcatel-lucent.com* Web: **https://support.lucent.com**
Central and South America	Alcatel-Lucent Calle 10, No. 145 San Pedro de los Pinos, 01180 Ciudad de Mexico Mexico	Mexico 01 800 123 8705 or (52) 55 5278 7235 Brazil 0800 89 19325 or (55) 193707 7900 Argentina 0800 666 1687 Venezuela 0 800 1004136 Costa Rica 0800-012-2222 or 1800 58 58877 For other local CALA numbers, consult the web site **https://support.lucent.com** or contact your local sales representative.
Asia Pacific	Alcatel-Lucent Australia 280 Botany Road Alexandria NSW 2015 Australia	Phone: 1800-458-236 (toll free from within Australia) (IDD) 800-5823-6888 (toll free from Asia Pacific - Hong Kong, Indonesia, South Korea, Malaysia, New Zealand, Philippines, Singapore, Taiwan, and Thailand) (613) 9614-8530 (toll call from any country) E-mail: *apactss@alcatel-lucent.com*

How to order

Customers can order additional VitalQIP manuals online at **http://www.lucentdocs.com/cgi-bin/CIC_store.cgi**. Select **VitalQIP** from the Product Line list and click **Go**.

How to comment

To comment on this , go to the Online Comment Form (**http://www.lucent-info.com/comments/**) or e-mail your comments to the Comments Hotline (**comments@alcatel-lucent.com**).

1 VitalQIP database functions

Overview

The VitalQIP Command Line Interface (CLI) provides another alternative process to using the VitalQIP Graphical User Interface (GUI). Commands permit the use of a prompt to carry out functions.

Note: Only the CLIs in Chapter 5, "Web Client CLIs" can be used with IPv6 objects and subnets. All other CLIs are for use only with IPv4 objects and subnets.

The CLI provides commands that allow you to do the following:

- Assign, reserve, dynamically allocate IP addresses
- Assign network objects to subnets
- Assign and maintain users and administrators
- Create and modify templates
- Manage objects
- Delete objects and de-allocate previously reserved addresses
- Move objects from one subnet to another, immediately or on a schedule
- Cancel scheduled moves
- Create, modify, delete, and get AC devices and AC subscribers
- List and search AC subscribers
- Reclaim unused IP addresses
- Generate reports from VitalQIP
- Create DNS configuration and data files for a domain, and distribute them to the primary DNS server
- Generate the Bootptab file
- Add, modify, and delete the key for the zone name specified in the inputfile to VitalQIP database, and retrieve the DNSSEC key for given keynames.
- Generate DHCP configuration files

- Define a DHCP scope
- Show active leases for DHCP
- Synchronize VitalQIP with DHCP
- Clone DNS and DHCP servers
- Export data from the database and import data into the database
- Clear entries in the audit table before a certain date
- Unlock selected and Planned Use addresses

Issuing CLI commands

The CLI commands are executable from various directories, depending on the platform. Refer to Table 1-1 to determine where you should execute the CLI commands.

Table 1-1 Location for execution of CLI commands

Platform	Executable location
Windows	*%QIPHOME%\cli*
UNIX enterprise	*$QIPHOME/usr/bin*

A few things to keep in mind

- All data lines must end with a carriage return, or they are not imported.
- If you do not specify the [**-s** ***servername***] [**-u** ***username***] [**-p** ***password***] parameters, the CLI looks in the *qip.pcy* file for the values. If you do not want to use these values taken from the *qip.pcy* file, you must specify them on the command line.
- The [**-o** ***organization***] parameter is only applicable if you are running the CLI as a master administrator. The [**-o** ***organization***] parameter is case sensitive for all CLI commands.
- For all CLI commands, the date and year format must be `mm/dd/yyyy`. If you use `mm/dd/yy`, the input data is rejected and an error message explaining that the date is invalid displays.
- All information passed on the command line must be surrounded with quotes if it contains a space. For example, passing the **-i** of *My File.txt* must be surrounded with quotes because it contains a space, therefore it would appear as **-i "My File.txt"**.
- Field names in input files are case-insensitive. Values, however, continue to be case-sensitive. For example,

```
Name=user1
name=user1
NAME=user1
```

are all the same. However,

```
Name=user1
Name=USER1
Name=User1
```

are not the same.

Notation key

To make it easier for you to code the parameters, this manual uses the notations described in Table 1-2 when explaining the synopsis of each CLI command:

Table 1-2 Synopsis notation conventions

Notation	Description
Bold	Used for directories, filenames, commands, and parameters. Type the boldface term as it appears in the Synopsis. **Example:** Type **`qip-dbinit`** as: **`qip-dbinit`**
Italics	Used to show generic arguments and options; replace text in italics with your own values. **Example:** Type **`-i`** ***`input_filename`*** as: **`-i input1.txt`**
[ ]	Used to indicate optional elements in a description of syntax. Do not type the brackets themselves. **Example:** Type **`[-m]`** as: **`-m`**
|	Used in syntax descriptions to separate items for which only one alternative can be chosen at a time. **Example:** Type **`-t active|expired|all`** as: **`-t active`** or **`-t expired`** or **`-t all`**
	Used to eliminate new lines in multiline inputs
`Constant width`	Used to show the contents of files or the output from commands.

Common arguments/parameters

Table 1-3 lists parameters that are valid for many CLI commands.

Table 1-3 CLI command parameters

Parameter	Explanation	Important notes
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.	
-s ***qip_dataserver***	Specifies the server on which to operate.	Optional if the appropriate value is defined in your policy file (*qip.pcy*) – refer to note following table.
-u ***username***	Specifies the user ID for the database.	Optional if the appropriate value is defined in your policy file (*qip.pcy*) – refer to note following table.
-p ***password***	Specifies the user password for the database.	Optional if the appropriate value is defined in your policy file (*qip.pcy*) – refer to note following table.
-f ***input_filename***	Specifies the name of the input file that contains the command parameters or the import information.	
-FO ***Formatted Output***	Adds line feed and carriage return characters to raw XML output to improve legibility.	This parameter is not currently functional. Support for this feature will be added in a future VitalQIP release.
-? or **-h**	Displays the syntax of the command's parameters (to assist you in coding the command).	Enter **-?** or **-h** to display the syntax, without processing the command.
-v	Displays version information for the CLI command.	

Parameter	Explanation	Important notes
-o *organization*	Specifies the user's organization. This parameter is ignored unless the user has system privileges (for example, qipman). The default is "VitalQIP Organization".	This parameter is only applicable if you are running the CLI as a master administrator. Also, this parameter is case sensitive for all CLI commands.
-ac *action*	Defines what action to take. You can add, modify, delete or retrieve information.	You can only perform one action at a time. You cannot add and modify in the same call, nor can you combine additions and modifications in one input file.

Note: A value must be provided. If the appropriate value is not defined in your policy file (*qip.pcy*), specify it in this parameter. If this parameter is omitted, the CLI looks for the value in the *qip.pcy* file.

Using the CLI commands

This section explains (in alphabetical sequence) the CLI commands used for functions other than exporting and importing information. The CLI commands for importing and exporting information are explained in other chapters of this document.

Note: In some tables in this chapter, the following notation is used to indicate whether input values are required or optional.

[M] = Mandatory
[O] = Optional

qip-active

qip-active displays active leases for the specified DHCP server or deletes an active lease. The Active Lease Report lists IP addresses in ascending order. The Object and Domain Name that appear in the listing are determined, as follows:

- If the Object Name or Domain Name exists in the DHCP server's database, that name is displayed.
- If the Object Name or Domain Name does not exist in the DHCP server's database, the system uses the name stored within VitalQIP if the object is defined. However, the Object Name and/or Domain Name is prefixed with an asterisk (*) to indicate that the name was taken from VitalQIP.
- If the Object Name or Domain Name does not exist in the DHCP server's database and no Object Name or Domain Name exists within VitalQIP (for example, the object has not been defined), the Object Name and/or Domain Name is blank.

Synopsis

```
qip-active -n dhcp_svr [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization]
   -b subnet_address|-j object_address
   [-r parsed_option82|unparsed_option82]
   [-t active|expired|all|delete]
```

Parameters

qip-active recognizes the following parameters:

-n ***dhcp_svr***	Specifies the fully-qualified name of the DHCP server for which the lease information is to be retrieved.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies VitalQIP organization (corporation) name.
-b ***subnet_address*** | **-j** ***object_address***	Specifies the Subnet or Object IP address for which lease information is retrieved.

-r *parsed_option82*| *unparsed_option82*	***Parsed_option82*** - Specifies that the Option 82 information is output in user-friendly format, with 4 additional fields: Circuit ID, Remote ID, Device Class, and Subnet Selection. ***Unparsed_option82*** - Specifies that the Option 82 information is output in unparsed format.
-t active|expired|all|delete	Specifies the type of retrieved lease information.

Command line input examples

- To view all active leases:

```
qip-active -u qipman -p passwd -n dhcpsvr.quadritek.com -t all
```

- To view all active leases with a specific subnet address:

```
qip-active -u qipman -p passwd -n dhcpserver.quadritek.com -t all
   -b 198.200.138.0
```

- To delete an active lease:

```
qip-active -u qipman -p passwd -n dhcpserver.quadritek.com -t delete
   -b 198.200.138.3
```

Output example 1

A typical output file example follows. Refer to Table 1-4, “Output field definition” (p. 1-10) for a definition of the output fields.

```
IP_Address  MAC_Addr. Exp. DHCP_Svr Lease_Granted
   Lease_Expired Ven. Class Client ID Host_Name Domain
-------------------------------------------------------
-------------------------------------------------------
-------------------------------------------------------
10.10.0.15   ac15dea1048f   No   10.30.40.9      05/11/2008
   01:11:31  07/21/2008 07:51:31  MSFT 5.0   deadbeefcafebabe
   dhcpclient01   tech.com
```

Output example 2

A typical output file example generated using the **-r** parameter follows. The Option 82 information is displayed with 4 additional fields: Circuit ID, Remote ID, Device Class, and Subnet Selection. Refer to Table 1-4, "Output field definition" (p. 1-10) for a definition of the output fields.

```
IP_Address MAC_Address Expired DHCP_Server
Lease_Granted Lease_Expired Last_Trans Client_ID Host_Name
Domain_Name Circuit_ID Remote_ID Device_Class
Subnet_Selection
------------------------------------------------------------
------------------------------------------------------------
------------------------------------------------------------
------------------------------------------------------------
144.144.144.01 300:60:97:40:ac:62 No 198.200.138.207
03/25/2003 02:22 05/14/2003 23:52 03/25/2003 02:22 MDLAPTOP
quadritek.com 80010006 0ac1578dc3ab4e CCCM 198.200.138.0
```

Table 1-4 Output field definition

Field	Description
IP_Address	The IP address of the object.
MAC_Address	The MAC or hardware address of the object.
Expired	**Yes**-Lease has expired. **No**-Lease has not expired.
DHCP_Server	The IP address of the DHCP Server.
Lease_Granted	The date and time of when the lease was granted.
Lease_Expired	The date and time of when the lease expires.
Last_Trans.	The date and time of when the last transaction occurred against this object.
Client_ID	The client identifier.
Host_Name	The host name of the Object.
Domain_Name	The domain name of the Object.
Circuit_ID	The agent-local identifier of the circuit from which a client-to-server packet was received. It is intended for use by relay agents in forwarding DHCP responses back to the proper circuit. For more information, see RFC 3046.

Field	Description
Remote_ID	The identifier that identifies the remote host end of the circuit to a DHCP relay agent that terminates a switched or permanent circuit. For more information, see RFC 3046.
Device_Class	Provides DOCSIS cable modem device attributes. For more information, see RFC 3256.
Subnet_Selection	The subnet/link IP address requested by a DHCP relay agent. For more information, see RFC 3527.

qip-admin

qip-admin adds, modifies, deletes, and retrieves administrators and their managed organizations in VitalQIP.

Synopsis

qip-admin [**-n** ***admin_name***] [**-mo** ***managed_organization***] [**-details**] [**-nc** ***add_new_contact***] [**-g** ***loginserver***] [**-s** ***dataserver***] [**-u** ***username***] [**-p** ***password***] [**-f** ***output_file***] [**-i** ***input_file***] **-ac** ***action*** **-df** ***data_format*** **-a** ***<new_admin_name>*** **-np** ***<new_admin_password>***

Parameters

qip-admin recognizes the following parameters:

-n ***admin_name***	Specifies the name of the administrator.
-mo ***managed_organization***	Specifies the name of the managed organization.
-details	Retrieves an entire profile, including administrator information, access information, and customization information. If an administrator is specified, customization options are also displayed. For an explanation of profile details, refer to "Administrators" in Chapter 6 of the *VitalQIP User's Guide*. **Note:** The **-df c** parameter (comma delimited data format) cannot be used with **-details**.
-nc ***add_new_contact***	Adds a new contact. Only used in conjunction with the **-ac add** parameter when you are adding a new administrator.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***dataserver***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-f ***output_file***	Specifies the directory and filename where the output data is to be placed.

-i ***input_file***	Specifies the directory and filename where the input data is to be placed.
-a ***new_admin_name***	Adds a new administrator name. Only used in conjunction with the **-ac clone** parameter when you are cloning an existing administrator record.
-np ***new_admin_password***	Adds a new administrator password. Only used in conjunction with the **-ac clone** parameter when you are cloning an existing administrator record.

-ac ***action***

Defines what action to take. Only one action may be performed at a time. You cannot add and modify in the same call, nor can you combine additions and modifications in one input file.

The ***action*** variable can have the following values:

add – adds an administrator. Use the **-i** option to specify the input file that contains the data. Use the **-df** option to specify the format of the input file. Use **-nc** to add a new contact.

mod – modifies an administrator. Use the **-i** option to specify the input file that contains the data. Use the **-df** option to specify the format of the input file.

clone – clones an administrator. Use the **-n** option to specify the existing administrator. Use the **-a** option to specify the new administrator name and **-np** to specify the password for the new administrator.

del – deletes an administrator. The **-n** parameter is also required. The administrator specified is removed from the database and all accounts created by the deleted administrator are reassigned to the administrator who performed the deletion.

get - retrieves a specific administrator's profile or a list of all administrators:

- To retrieve a specific administrator's default profile, use the **-n** parameter with the administrator's name. The default profile contains the administrator's profile information, contact information, default system privileges, and GUI preferences. If the password field is included in the input file, it will not appear in the output.
- To retrieve a list of all administrators in the database, omit the **-n** parameter. The output for a list includes only the Login Name and Administrator Type.

Note: Managed organizations are not retrieved by default. If you need to retrieve an entire profile including managed organizations, also specify the **-details** parameter.

addorg – adds an organization. Use the **-i** option to specify the input file that contains the data. Use the **-df** option to specify the format of the input file.

modorg – modifies an organization. Use the **-i** option to specify the input file that contains the data. Use the **-df** option to specify the format of the input file.

getorg – retrieves a specific organization or a list of all organizations managed by an administrator:

- To retrieve a managed organization for an administrator, specify the administrator with the **-n** parameter and the managed organization with the **-mo** parameter.
- To retrieve a list of all managed organizations for an administrator, specify the administrator with the **-n** parameter and omit the **-mo** parameter. The output for a list includes only the Organization Name, Managed Type, and Read Only flag.

delorg – deletes an organization. Both the **-n** parameter and the **-mo** parameter are required. The organization specified in the **-mo** parameter is removed from the administrator specified in the **-n** parameter.

-df ***data_format***

Determines the output format. The ***data_format*** variable has two values: **c** and **n**. Use the **c** variable to produce comma delimited output. Use the **n** variable to produce name value pairs output. If the **-df** parameter is omitted from the command line, the default is name value pairs output.

The output data format is identical to the input format shown in "Comma delimited file format" below, and "Name value pairs file format" (p. 1-21).

Note: The comma delimited data format cannot be used with the **-details** flag.

qip-admin input file

Default system privileges

If any default system privileges are specified in the file, the 'Default System Privileges' and the 'End Default System Privileges' placeholder columns are required (data is never present in these fields).

The fields between these tags may vary based on the setting of other fields. For example, the ‘Create Billing Infrastructure’, ‘Create Infrastructure’, and ‘Create Resource Records’ fields are applicable if the ‘Highest Level GUI Mode’ is set to a value other than ‘Basic Mode’. If these fields are specified but the ‘Highest Level GUI Mode’ is set to ‘Basic Mode’, these fields are ignored. Consequently, if the ‘Highest Level GUI Mode’ is to be set, it should come before the children fields (or the children fields will be ignored). The best way to understand which fields are applicable is to check the GUI, where fields that are not applicable are not displayed.

Bit representation of menu preferences

The input file can also contain a section that establishes the specific menus that are enabled for an administrator. In the following preference fields,

```
FilePref=1111
HelpPref=111
InfrastructurePref=111111111111
ManagementPref=11111111111
NetworkPref=1111111
PolicyPref=110100011100
ReportPref=111111111111
ViewPref=11111
ImportPref=1111111
UIQuickViewPref=000
```

“1” and “0” are bit representations that indicate whether the menu option is on or off respectively. These menu settings correspond to the settings on the Customize menu in the Administrator Profile GUI. Refer to Table 1-5 to determine the menu preference by bit representation order.

Table 1-5 Bit representation of menu preferences

Preference field name	Order	Bit representation (1 equals on and 0 equals off)
FilePref	1	Administrator Password
	2	Change Organization
	3	Change Server/Administrator
	4	Exit
HelpPref	1	About VitalQIP
	2	Contents
	3	Technical Support

Preference field name	Order	Bit representation (1 equals on and 0 equals off)
InfrastructurePref	1	Application Profile
	2	Domain Profile
	3	OSPF Profile
	4	User Group Profile
	5	Non-Managed DNS Server Profile
	6	Network/Reverse Zone Profile
	7	Organization Profile
	8	Server Profile
	9	Not Applicable
	10	Subnet Organization Profile
	11	Administrative Role Profile. **Note:** Set this to 0 if the Administrator Type is set to “Normal”, since only Master and Organization administrator types can create Administrative Roles.
	12	Administrator Profile
ManagementPref	1	Go To Search|Objects by Contact
	2	Go To Search|Objects by Location
	3	Go To Search|Objects
	4	Global MAC Address Pool
	5	Object Management|All Subnets
	6	Object Management|Used Subnets
	7	Object Management|Unused Subnets
	8	QIP Hierarchy
	9	Reclaim Addresses
	10	User Management|User Group
	11	User Management|User Profile

Preference field name	Order	Bit representation (1 equals on and 0 equals off)
NetworkPref	1	Bootptab File Generation
	2	DNS Generation
	3	DHCP Generation
	4	Local Host Generation
	5	NIS Generation
	6	View Active Leases
	7	Windows 2000 DC Generation
PolicyPref	1	ACL Template
	2	Contact
	3	Class/Option Setup
	4	Option Template
	5	Policy Template
	6	Global Policies
	7	Location Profile
	8	Manufacturer Profile
	9	Object Class
	10	Naming Policy
	11	User-Defined Fields
	12	Client Class

Preference field name	Order	Bit representation (1 equals on and 0 equals off)
ReportPref	1	Administrator Audit History
	2	Management Reports|Administrator Profile
	3	Management Reports|Free Subnet
	4	Management Reports|DHCP Report
	5	Management Reports|Inquire Report
	6	Management Reports|Object by Location Report
	7	Management Reports|Object by Address Range Report
	8	Management Reports|Zone Resource Record Report
	9	Management Reports|Object by Administrator Profile Report
	10	Management Reports|Object by Administrative Role Report
	11	Management Reports|Object by Application Report
	12	Object Audit History
ViewPref	1	Advanced Mode
	2	Basic Mode
	3	Hierarchy Legend
	4	Standard Mode
	5	Toolbar
ImportPref	1	Import|Domain
	2	Import|OSPF
	3	Import|MAC Addresses
	4	Import|Network
	5	Import|Subnet Organizations
	6	Import|Objects
	7	Import|Subnets
UIQuickViewPref	1	Display Domain Folders
	2	Network Quick View
	3	Subnet Quick View

Comma delimited file format

The first line of a comma delimited list file contains a layout of the items that will be entered. The columns can be in any order and any optional columns can be excluded.

The next lines encountered will be the data lines. Multiple data lines may exist, each separated by a line feed.

Note: The comma delimited data format cannot be used with **-details**.

Command line input examples

For a description of the fields in any of the examples for the **qip-admin** CLI, refer to "Administrators" in Chapter 6 of the *VitalQIP User's Guide*.

Add an administrator example

To add an administrator in a comma delimited file, enter the following. The sample input file illustrates the contents of *c:\input.txt*.

```
qip-admin -u qipman -p passwd -ac add -i c:\input.txt -df c
```

Sample input file

```
Login Name, Password, Type,First Name,Last
   Name,Email,Phone,Pager,Business Unit ID,Default Printer,Default
   System Privileges,Global Privileges,Create/Update
   Administrator,Organization Defaults,Access Address
   Allocation,Maintain Blocks,Allocate Blocks,Rule Level,Expand
   Blocks,Explicitly Create Blocks,Free Blocks,Move
   Blocks,Renumber Blocks,Split/Merge Blocks,Maintain Object
   Templates,Maintain Pools,Maintain Reverse Zone
   Templates,Maintain Rules,Maintain Seed Pool,Maintain Subnet
   Templates,Rule Override,Write New Infrastructure to Managed
   List,Highest Level GUI Mode,Create Billing
   Infrastructure,Create Infrastructure,Create Resource
   Records,Allow User Selection,Delete Confirmation
   Warning,Require Alias,Require Contact Name,Require
   Location,Require MAC Address,Require Manufacturer
   Information,Restrict CNAME,Restrict Subnet,Unique Name
   Warning,End Default System
   Privileges,FilePref,HelpPref,InfrastructurePref,ManagementPref
   ,
   NetworkPref,PolicyPref,ReportPref,ViewPref,ImportPref,
   UIQuickViewPref
admin3,MyPassword,Normal,Joe,Sample,jsample@lucent.com,2123663344
   ,
   2125551111,ID10T,QAPrint,,,False,,True,True,True,Normal,False,
   False,False,False,False,False,False,False,False,False,Fa
   lse,False,
   True,Advanced
   Mode,False,False,True,True,True,False,True,True,False,
```

```
False,False,False,False,True,1111,111,111111111111,11111111111
,
1111111,111111111111,11111111111,11111,1111111,000
```

Add a managed organization to a managed list example

To add a Managed Organization to an access list for an administrator named Bob, enter the following.

`qip-admin -u qipman -p passwd -ac addorg -n Bob -i c:\input.txt -df c`

The details of the Managed Organization are contained in the comma delimited sample input file (*c:\input.txt)*.

```
Organization Name,Managed Type,Read Only, System Privileges,Use
   Default,Highest Level GUI Mode,Create Billing
   Infrastructure,Create Infrastructure,Create Resource
   Records,Create/Update Administrator,Allow User Selection,Delete
   Confirmation Warning,Network Allocator Admin,Require
   Alias,Require Contact Name,Require Location,Require MAC
   Address,Require Manufacturer Information,Restrict
   CNAME,Restrict Subnet,Unique Name Warning,End Default System
   Privileges
Org2,Normal,False,,False,Advanced
   Mode,False,False,False,False,False,
   False,False,False,False,False,False,False,False,False,False,
```

Name value pairs file format

Each line of the file starts with a field name. The valid field names are identical to the column names for the comma delimited file format. The fields can be listed in order. More than one object may be specified in an input file. Objects should be separated by a blank line.

> **Note:** The same logic regarding default system privileges applies to the name value pairs format. Refer to "Default system privileges" (p. 1-15).

Command line input examples

For a description of the fields in any of the examples for the **`qip-admin`** CLI, refer to "Administrators" in Chapter 6 of the *VitalQIP User's Guide*.

Add administrator example

To add an administrator in a name value pairs file (the default), enter the following.

`qip-admin -u qipman -p passwd -ac add -i c:\input.txt`

The input file example illustrates the contents of *c:\input.txt*.

```
Login Name=admin1
```

```
Password=***********
Type=Normal
First Name=Joe
Last Name=Sample
Email=jsample@lucent.com
Phone=2123663344
Pager=2125551111
Business Unit ID=ID10T
Default Printer=QAPrint
Default System Privileges
        Global Privileges=
                Create/Update Administrator=False
          Allow Password Expiration=True
        Organization Defaults=
                Access Address Allocation=True
                        Maintain Blocks=True
                                Allocate Blocks=True
                                        Rule Level=Normal
                                Expand Blocks=False
                                Explicitly Create Blocks=False
                                Free Blocks=False
                                Move Blocks=False
                                Renumber Blocks=False
                                Split/Merge Blocks=False
                        Maintain Object Templates=False
                        Maintain Pools=False
                        Maintain Reverse Zone Templates=False
                        Maintain Rules=False
                        Maintain Seed Pool=False
                        Maintain Subnet Templates=False
                        Rule Override=False
                        Write New Infrastructure to Managed
   List=True
                Highest Level GUI Mode=Advanced Mode
                        Create Billing Infrastructure=False
                        Create Infrastructure=False
                        Create Resource Records=True
                Allow User Selection=True
                        Require User=False
                Delete Confirmation Warning=True
                Require Alias=False
                Require Contact Name=True
                Require Location=True
                Require MAC Address=False
                Require Manufacturer Information=False
```

```
                Restrict CNAME=False
                Restrict Subnet=False
                Unique Name Warning=False
```

Add managed organization example

To add a Managed Organization to an access list for an administrator named Bob, enter the following.

qip-admin -u qipman -p passwd -ac addorg -n Bob -i c:\input.txt -df n

The details of the Managed Organization are contained in the name value pairs input file (*c:\input.txt)*.

```
Organization Name=Org2
  Managed Type=Normal
  Read Only=False
  Default System Privileges
    Use Default=False
    Allow Password Expiration=True
      Highest Level GUI Mode=Advanced Mode
        Create Billing Infrastructure=False
        Create Infrastructure=False
        Create Resource Records=False
        Create/Update Administrator=False
      Allow User Selection=False
      Delete Confirmation Warning=False
      Network Allocator Admin=False
      Require Alias=False
      Require Contact Name=False
      Require Location=False
      Require MAC Address=False
      Require Manufacturer Information=False
      Restrict CNAME=False
      Restrict Subnet=False
      Unique Name Warning=False
  End Default System Privileges
```

Add administrator with contact information example

The following example shows sample input to add an administrator and provide new contact information:

qip-admin -u qipman -password <passwd> -ac add -n Joe -i c:\input.txt -df n -nc

Sample input file

```
Login Name=Joe
Password=*******
Type=Master
First Name=Barbara
Last Name=Rogers
Email=brogers@bigcompany.com
Phone=123-555-1212
Pager=
Business Unit ID=
Default Printer=
Default System Privileges
Allow Password Expiration=True
Delete Confirmation=False
Require Alias=True
Require Contact Name=False
Require Location=False
Require MAC Address=False
Require Manufacturer Information=False
Require User=False
Unique Name Warning=True
End Default System Privileges
FilePref=1111
HelpPref=111
InfrastructurePref=111111111111
ManagementPref=11111111111
NetworkPref=1111111
PolicyPref=110100011100
ReportPref=111111111111
ViewPref=11111
ImportPref=1111111
UIQuickViewPref=000
```

Delete administrator example

To delete the existing administrator “adminD”:

qip-admin –u qipman –p passwd –ac del –n adminD

Modify administrator example

To modify the administrator “Bob” with the name value pairs file *c:\input.txt*.

qip-admin -u qipman -p passwd -ac mod -n Bob -i c:\input.txt

Retrieve a list of managed organizations example

To retrieve a list of all of Administrator “Bob”’s Managed Organizations. The list will contain each Managed Organization's name, Managed Type, and Read/Write access.

```
qip-admin -u qipman -p passwd -ac getorg -n Bob
```

Retrieve specific managed organization example

To retrieve Administrator “Bob”’s Managed Organization “VitalQIP Organization”.

```
qip-admin -u qipman -p passwd -ac getorg -n Bob -mo "VitalQIP
   Organization"
```

Retrieve administrator information in name value file example

The following command line retrieves information for administrator “Bob” into a name value pairs file. The content of *c:\output.txt* is shown below.

```
qip-admin -u qipman -p passwd -ac get -n Bob -f c:\output.txt -df
   n
```

The following example shows sample output in *c:\output.txt*:

```
Login Name=Bob
Type=Master
First Name=
Last Name=
Email=
Phone=
Pager=
Business Unit ID=
Default Printer=
Default System Privileges
  Allow Password Expiration=True
  Delete Confirmation=False
  Require Alias=True
  Require Contact Name=False
  Require Location=False
  Require MAC Address=False
  Require Manufacturer Information=False
  Require User=False
  Unique Name Warning=True
End Default System Privileges
FilePref=1111
HelpPref=111
InfrastructurePref=111111111111
ManagementPref=11111111111
NetworkPref=1111111
PolicyPref=110100011100
ReportPref=111111111111
```

```
ViewPref=11111
ImportPref=1111111
UIQuickViewPref=000
```

To add a managed list

Purpose

Use this procedure to add a managed list to a new administrator.

Procedure

To add a managed list, follow these steps:

1 Clone an existing administrator profile, as follows:

`qip-admin -ac get -n normal -details -f admin.txt`

2 Modify the input file (*admin.txt*) and change the administrator name and password. Assuming the new administrator and password were set to "`normal3`", the file would look like this:

```
Login Name=normal3
Password=normal3
Type=Normal
First Name=
Last Name=
Email=
Phone=
Pager=
Business Unit ID=
Default Printer=
Default System Privileges
        Global Privileges=
         Allow Password Expiration=True
         Create/Update Administrator=False
        Organization Defaults=
                Access Address Allocation=True
                        Maintain Blocks=True
                                Allocate Blocks=True
                                        Rule Level=Normal
                                Expand Blocks=False
                                Explicitly Create Blocks=False
                                Free Blocks=False
                                Move Blocks=False
```

```
                                        Renumber Blocks=False
                                        Split/Merge Blocks=False
                                Maintain Object Templates=False
                                Maintain Pools=False
                                Maintain Reverse Zone Templates=False
                                Maintain Rules=False
                                Maintain Seed Pool=False
                                Maintain Subnet Templates=False
                                Rule Override=False
                                Write New Infrastructure to Managed
   List=True
                        Highest Level GUI Mode=Advanced Mode
                                Create Billing Infrastructure=False
                                Create Infrastructure=False
                                Create Resource Records=True
                        Allow User Selection=True
                                Require User=False
                        Delete Confirmation Warning=True
                        Require Alias=False
                        Require Contact Name=True
                        Require Location=True
                        Require MAC Address=False
                        Require Manufacturer Information=False
                        Restrict CNAME=False
                        Restrict Subnet=False
                        Unique Name Warning=False
            End Default System Privileges
            FilePref=1111
            HelpPref=111
            InfrastructurePref=111111111101
            ManagementPref=11111111111
            NetworkPref=1111111
            PolicyPref=111111111111
            ReportPref=111111111111
            ViewPref=11111
            ImportPref=1111111
            UIQuickViewPref=000
            Managed Organizations
            Organization Name=VitalQIP Organization
            Managed Type=Normal
            Read Only=False
            System Privileges
            Use Default=True
            End System Privileges
            Managed Range
```

```
Subnet;Customer1;10.59.0.0;;False
End Managed Range
End Managed Organizations
```

3 Run **qip-admin** to create the administrator `normal3` using the following command. This command creates the administrator profile, but it lacks managed range information.

```
qip-admin -ac add -i admin.txt
```

4 Edit the managed range section in *admin.txt* and change the managed range items to be name,value pairs, as follows.

```
Managed Organizations
  Organization Name=CLI_Automation_Org
    Managed Type=Normal
    Read Only=False
    System Privileges
      Use Default=True
    End System Privileges
    Managed Range
      Subnet;;100.200.80.0;;False
      Domain;qtek.com;;;False
      Network;CLI-NETNAME2;100.100.80.0;;False
      Address Range;;100.200.80.1;100.200.80.60;False
    End Managed Range
End Managed Organizations
```

5 Run **qip-admin** again using the following command:

```
qip-admin -ac addorg -i admin.txt -n normal3
```

6 To add additional managed range items, run **qip-admin** with the following command:

```
qip-admin -ac modorg -i admin2.txt -n normal3
```

END OF STEPS

qip-altersubnet

qip-altersubnet provides the capability to split or join subnets.

Synopsis

```
qip-altersubnet [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] -f fromSubnets -t toSubnets
   -m newMask
```

Parameters

qip-altersubnet recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *fromSubnets*	Specifies an existing subnet address (to split) or addresses (to join), separated by a comma.
-t *toSubnets*	Specifies the resulting subnet address (after join) or addresses (after split), separated by a comma.
-m *newMask*	Specifies the new mask to be applied.

Command line input examples

- To join subnet addresses:

```
qip-altersubnet –u qipman –p passwd –f 144.144.32.0,144.144.48.0
   ,144.144.64.0 –t 144.144.32.0 –m 255.255.224.0
```

- To split subnet addresses:

```
qip-altersubnet –u qipman –p passwd –f 135.135.0.0
   -t 135.135.0.0,135.135.0.128,135.135.1.0,135.135.1.128 -m 255.255.255.128
```

qip-bootpgen

qip-bootpgen generates the Bootptab file for the specified Bootp server. The file is then copied and output to the specified file.

Synopsis

qip-bootpgen -n *server_name* [**-g *loginserver***] [**-s *servername***] [**-u *username***] [**-p *password***] [**-o *organization***] [**-f *filename***]

Parameters

qip-bootpgen recognizes the following parameters:

-n *server_name*	Specifies the fully-qualified Bootp server name for which the Bootptab file is generated.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the filename to which the Bootptab file is output.

Note: If the **-f** parameter is omitted, the Bootptab file will be generated based on the setting of parameter 'Bootptab Filename' in server profile. If the server profile 'Bootptab Filename' does not include the full path, the file will be generated in *QIPHOME/etc*.

Command line input example

qip-bootpgen –u qipman –p passwd-n bootpserver.quadritek.com –f /etc/bootptab

Output example

```
Defaults:\
   :ds=130.63.168.21 130.63.237.99 130.251.1.2:\
   :sm=255.255.255.0:\
   :to=18000:
Subnet243:\
   :tc=Global_Defaults:\
   :ht=ethernet:\
   :gw=130.63.243.1:
```

qip-changepassword

qip-changepassword changes the password of an administrator. For more information on passwords, refer to Chapter 1 of the *VitalQIP User's Guide*.

Synopsis

```
qip-changepassword -u loginname [-s servername] [-g loginserver]
   [-o org_name] -p password -n new_password -r retype_password
```

Parameters

qip-changepassword recognizes the following parameters:

-u ***loginname***	Specifies the VitalQIP login name for which you are changing the password.
-s servername	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-o ***org_name***	Specifies the VitalQIP organization (corporation) name.
-p ***password***	Specifies the existing password for the associated administrator account.
-n ***new_password***	Specifies the new password to replace the existing password.
-r ***retype_password***	Specifies the new password to replace the existing password. The value you assign to this option must match the **-n** option.

Command line input example

```
qip-changepassword –u qipman –p passwd –n qipmanxyz
```

qip-check

qip-check checks the current date with the date of all reserved addresses. If the reserved addresses have expired, it sets the status of the object to “unused”.

Synopsis

```
qip-check [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization]
```

Parameters

qip-check recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server’s IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.

Command line input example

```
qip-check –u qipman –p passwd
```

qip-checkobjname

qip-checkobjname shows all addresses associated with the specified host name or alias name.

Synopsis

```
qip-checkobjname [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] -n object|alias_name
```

Parameters

qip-checkobjname recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-n *object|alias_name*	Specifies the object's fully-qualified name or alias.

Command line input examples

```
qip-checkobjname –u qipman –p passwd –n myobject
qip-checkobjname -u qipman –p passwd –n www
```

Output examples

The following objects are using the same name:

```
10.98.4.12
100.97.4.12
```

qip-clear

qip-clear deletes entries in the audit table *before* the date specified. To clear all audit data, enter **-d ALL**.

Synopsis

```
qip-clear [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] -d mm/dd/yyyy|ALL
```

Parameters

qip-clear recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-d *mm/dd/yyyy*|ALL	***Required***. Specifies the date (in mm/dd/yyyy format) before which entries are cleared. To clear all audit data, enter **-d ALL** instead of a date.

Command line input example

```
qip-clear –u qipman –g <loginservername> -s <servername> –p passwd
   –d 12/30/1996
```

qip-crypt

qip-crypt allows you to encrypt a password. **qip-crypt** takes the password as the first argument and sends a hex-string encrypted password to *STDOUT*. The **qip-crypt** CLI command must be run again and the new password placed in the *qip.pcy* file if the qipman's password, the qipadmin's password, the Schedule Password, or the Update Password is changed.

Note: **qip-crypt** is only intended for use with a password in the *qip.pcy* file. It should not be used to encrypt the database login. Database logins can be encrypted using third-party tools; refer to your system administrator for more information relevant to your database.

Synopsis

qip-crypt [***password***]

Parameters

qip-crypt recognizes the following parameters:

password Specifies your current password.

Note: The qipadmin password must be alphanumeric with no special characters. Although **qip-crypt** will accept $, &, and so on, Sybase's sp_password will not.

qip-dbinit

qip-dbinit initializes the VitalQIP *or* Audit Manager database for the Sybase and Oracle database servers. For **qip-dbinit** details, refer to Chapter 9, "Database Administration" in the *VitalQIP Administrator Reference Manual.*

When using **qip-dbinit**, the full path must be specified (for example, enter *$QIPHOME/script/qip-dbinit*), or you must execute **qip-dbinit** from the *$QIPHOME/script* directory.

Note: Make sure that you back up your database and have shut down your VitalQIP services before running this CLI.

Synopsis

```
qip-dbinit [-t qip_dbase] [-s servername] [-u username]
   [-p password] [-l log_file] [-q output_file] [-i script_path]
   [-b db_name] [-a] [-k] [-z encrypted_password]
```

Parameters

qip-dbinit recognizes the following parameters:

-t ***qip_dbase***	Specifies the database server type - either **Oracle** or **Sybase**.
-s ***servername***	Specifies the name of the database server. Note the following: • The database server name must match the Sybase server name or the Oracle database alias name. • This parameter is optional if the QIPDATASERVER environment variable is set. • The command line argument overrides the environment variable. For Audit Manager, the Audit Manager database server name must be specified.
-u ***username***	Specifies the VitalQIP or Audit Manager administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-l ***log_file***	Specifies the name of the log file.
-q ***output_file***	Quiet Mode. If this parameter is omitted, the output is sent to *STDOUT*.
-i ***script_path***	Specifies the directory where the SQL scripts reside.
-b ***db_name***	Specifies the database name; QIP *or* LAM. (You cannot specify both databases at the same time.) The default is QIP.
-a	Appends to the log file. The default is to overwrite the log file.
-k	Skips the prompt.

-z encrypted_ password	Allows commands to be executed with the encrypted password.

qip-del

qip-del removes objects and unallocates addresses that have been reserved. This function can also be used to delete objects of various "Owner Types" (refer to Table 1-6 following).

Note: An Object Range deleted by using **qip-del** does not delete objects associated with the Object Range. However, all ties with the Object Range are broken.

Synopsis

```
qip-del [-g loginserver] [-s servername] [-u username]
   [-p password] -n name|-a address [-o organization] [-r]
   [-l appl_name] [-m] [-t owner_type] [-c class] [-e]
```

Parameters

qip-del recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-n ***name*** |**-a** ***address***	Specifies the fully-qualified name of the object, domain, contact ID, location ID or subnet IP of the object where the reserved object resides (**-n**), or the address ranges (**-a**). For the removal of a non-managed DNS server, both name and address are required: **-n** ***managed_server_name*** **-a** ***non_managed_server_address***
-r	Unallocates previously reserved addresses on the subnet specified in the **-a** option.
-l ***appl_name***	The application name for which the addresses were reserved.

-m	Deletes all names if the fully-qualified name is used and more than one object/subnet exists with that name. This parameter is only applicable when you specify the **-n** parameter.
-t *owner_type*	Refer to Table 1-6 following.
-c *class*	Specified policy or user-defined field class. Only valid for **-t** Policy and **-t** Userfield.
-e	Forces a delete of an object specified in the **-t** option, if the object is a server. Also forces a delete of either a DNS Server or DHCP Server. If the user attempts to delete an object that is either a DNS Server or DHCP Server without specifying the "-e" option, the following error is displayed: `Can not delete server objects - use -e to override this check`

Table 1-6 Owner types

Owner_Type	Description
Object	Deletes a used object or a reserved object.
DC_server	Deletes the domain controller.
Dhcp_server	Deletes the dhcp server by name
Dns_server	Deletes the dns server by name.
Object_range	Deletes the object range.
Domain	Deletes the domain and detaches all associations.
Reverse_zone	Deletes the zone and detaches all associations.
Network	Deletes the network.
Subnet	Deletes all objects in the subnet.
Subnet_organization	Deletes the subnet organization.
Address_range	Deletes the administrator address range.
User	Deletes the user via loginID.
Contact	Deletes a contact. Pass the ContactID in the **-n** parameter. You can determine the contact ID by using the **qip-getcontactlst** command.
Location	Deletes a location. Pass the LocationID in the **-n** parameter.

Owner_Type	Description
Policy	Deletes a Global Policy. Pass the Policy Class in the **-c** parameter (for example, General, Billing, DYNDNS, Oracle, Reports). Pass the Policy Name in the **-n** parameter.
Userfield	Deletes a User-Defined Field (UDF). Pass the UDF type in the **-c** parameter (for example, Domain, Object, Organization, Subnet, Reverse Zone, User). Pass the UDF Name in the **-n** parameter.
Access_Nonmanaged_Server	This owner type allows you to delete non-managed DNS servers. When you specify this owner type, you must also specify values for both **-n *<name>*** and **-a *<address>*** parameters.
Client_Class	Deletes a client class. Pass the client class name with the -n parameter.
Dhcp_Template	Deletes a DHCP Template. Pass the dhcp template name with the -n parameter

Command line input examples

- To delete all reserved objects within a subnet associated with application **appl2** (two examples):

```
qip-del -a 144.144.144.0 -r -1 appl2 -s srv1 -u qipman -p passwd
qip-del –n qtek.com –r -l appl2 -m -s svr1 -u qipman -p passwd
```

- To delete domains:

```
qip-del –n qtek.com -t domain -s srv1 -u qipman -p passwd
```

- To delete a domain controller:

```
qip-del -g 10.10.0.23 -s MARS -o “VitalQIP Organization” -t
   DC_SERVER -n 55DC.statefarm.org
```

- To delete object 198.200.138.123:

```
qip-del –u qipman –p passwd –a 198.200.138.123 - t object
```

- To delete all reserved objects in the subnet 198.200.138.192:

```
qip-del –u qipman –p passwd –a 198.200.138.192 -t subnet –r
```

- To delete an address range:

```
qip-del -u qipman -p passwd -t Address_Range -a 199.199.199.10-
   199.199.199.120
```

- The command to delete a non-managed server is:

```
qip-del -n 'test' -a 'address' -t Access_Nonmanaged_Server
```

qip-delacltemplate

qip-delacltemplate deletes an ACL Template. Once called, the CLI queries the database for zone option associations. If there are any zones associated with the template, an error message tells the user that there are zones associated with the template. If the user still wants to delete the template, the -sm (suppress message) flag suppresses the message and turns off the checking in the database. It deletes the template and all associations automatically.

Synopsis

```
qip-delacltemplate [-g loginserver] [-s servername] [-e errorfile]
   [-o organization] [-u username] [-p password] -tn template_name
   -sm
```

Parameters

qip-delacltemplate recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-tn *template_name*	Specifies the name of the template being retrieved.
-sm	Specifies that messages are suppressed when this CLI runs.
-e *error_file*	Specifies the file that contains errors generated by this CLI.

qip-deladminrole

qip-deladminrole deletes an administrative role for the VitalQIP enterprise server.

Synopsis

```
qip-deladminrole -n role_name [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization] -c [-f filename]
```

Parameters

qip-deladminrole recognizes the following parameters:

-n ***role_name***	Specifies the administrative role name you want to delete.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-c	Request that the output is in CSV format.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-deladminrole -n "admin role" -u qipman -p passwd
```

qip-delpolicy

qip-delpolicy deletes a policy entry from the VitalQIP enterprise server.

Note: When you delete a policy, you are resetting the policy back to its default value. The policy still displays in the graphical user interface in the Global Policies. (For more information on the policies, refer to the "Global Policies" section in Chapter 2 of the *VitalQIP User's Guide.*)

Synopsis

```
qip-delpolicy -c class -n policy_name [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
```

Parameters

qip-delpolicy recognizes the following parameters:

-c *class*	Specifies the class name associated with the policy you want to delete (for example, DNS).
-n *policy_name*	Specifies the policy name you want to delete (for example, PING_DELAY). For details about class lists and policy, refer to the "Define a DHCP/Bootp Template option" section in Chapter 2 of the *VitalQIP User's Guide*.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.

Command line input example

- To delete the **GENERAL PING_DELAY** policy from the VitalQIP enterprise server:

```
qip-delpolicy -s QIPSYBASE -u qipman -p passwd -c GENERAL -n
   PING_DELAY
```

qip-dhcpdebuglog

qip-dhcpdebuglog clears the debug log, changes the debug level, or stops the debug logging for a specified server, without restarting the server.

Synopsis

```
qip-dhcpdebuglog -n DHCP Server -c|-d new debug level|
   -t stop debug level [-g login_server] [-s dataserver]
   [-u username] [-p password] [-o organization] [-f output_file]
```

Parameters

qip-dhcpdebuglog recognizes the following parameters:

-n *DHCP Server* Specifies the DHCP server being debugged.

Note: One of the following three parameters, **-c**, **-d** or **-t** must be specified along with the **-n** parameter above.

-c Clears the debug log

-d *new debug level* Specifies the debug level for the server. Allowable values are:

- LevelCritical
- LevelError/
- LevelWarning
- LevelInfo
- LevelDebug

-t *stop debug level* Stops debugging the server.

-g *loginserver* Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.

-s *servername* Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.

-u *username* Specifies the VitalQIP administrator account to be used in establishing the database connection.

-p *password* Specifies the password for the associated administrator account.

-o *organization* Specifies the VitalQIP organization (corporation) name.

-f *output_file* Specifies the directory and filename where the output data is to be placed. The default is STDOUT.

qip-dhcpdump

qip-dhcpdump reads the VitalQIP Lease Disk file and outputs all active lease information to a "|" delimited output file.

Synopsis

qip-dhcpdump -i *input_filename* -f *output_filename*
[**-t active|expired|all**]

Parameters

qip-dhcpdump recognizes the following parameters:

-i *input_filename*
Specifies the VitalQIP Active Lease disk file path and name (for example, */$QIPHOME/usr/bin/dhcp.db*).

In UNIX the default path for the *dhcp.db* file is in the directory of the **$QDHCPCONFIG** environment variable.

In Windows the default path for the *dhcp.db* file is in *%QIPHOME%/dhcp*.

Note: The *dhcp.db* file can be placed in a directory other than the default directory.

For information on converting Active Lease files from previous versions of VitalQIP, refer to **qip-qdhcplease**.

-f *output_filename*
Specifies the directory and filename where the output data is to be placed. The default filename is the path you are running the process from and is named *dhcpdump.dat* in the current directory.

-t active|expired|all
Specifies whether you want "active" lease information, "expired" lease information, or "all" lease information.

Command line input example

```
qip-dhcpdump  -i /QIPHOME/dhcp/dhcp.db -f /mydhcpdumplist.txt -t expired
```

Output example

```
144.144.144.14|00:80:5f:01:84:f4|ws_support_3|quadritek.com|19970307
   13:12|19970307 13:42|Expired
144.144.144.15|00:80:5f:61:c4:91|AGRABIL|quadritek.com|19970311
   05:34|19970311 06:04|Expired
135.16.53.13|00:40:97:40:ca:62|LAPTOP23|quadritek.com|19970314
   12:36|19970314 13:06|Expired
135.16.53.14|00:40:4f:01:84:f4|ws_support_6|quadritek.com|19970307
   13:12|19970307 13:42|Expired
135.16.53.15|00:40:4f:61:c4:91|ws_support_5|quadritek.com|19970311
   05:34|19970311 06:04|Expired
```

qip-dhcpgen

qip-dhcpgen generates a DHCP configuration file for the specified DHCP server.

Synopsis

qip-dhcpgen -n *dhcp_svr* [**-d *directory***] [**-g *loginserver***] [**-s *servername***] [**-u *username***] [**-p *password***] [**-o *organization***] [**-l**] **[-f]**

Parameters

qip-dhcpgen recognizes the following parameters:

-n *dhcp_svr*	Specifies the fully-qualified DHCP server name for which the DHCP configuration file is generated.
-d *directory*	Specifies the directory name to which the DHCP configuration file is generated. If this parameter is omitted, it is written to the directory defined for the DHCP server.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-l	Generates the DHCP files on the local server.
-f	Forces a DHCP push to occur.

Command line input example

qip-dhcpgen –u qipman –p passwd –n dhcpserver.qtek.com –d c:\qip\dhcp

Output example

```
Server-identifier quad.qtek.com;
# Name: sol2
  shared-network 198_200_130_0 {
    subnet 198.200.1380.0 netmask 255.255.255.0 {
      manual-dhcp range 02-03-04-05-06-07 198.200.138.9 {
        option subnet-mask 255.255.255.0;
        option domain-name "qtek.com";
        option domain-name-servers
198.200.138.174;
        option routers 198.200.138.1;
```

```
            option dhcp-lease-time 4294967295;
        }
    }
}
```

qip-dhcpsync

qip-dhcpsync synchronizes VitalQIP with the true client-suggested names and their IP address bindings that are located in the DHCP server's Active Lease database. The process is illustrated in Figure 1-1. This command checks the DHCP server and obtains all active ***and*** expired leases known to the server.

Note: **qip-dhcpsync** is only used for recovery purposes and does not need to be run on a regular basis.

Figure 1-1 Name synchronization

When an Expired Object is updated in the VitalQIP database, the MAC address is removed and the host name reverts to the prior name. Likewise, when an Active Object is updated in the VitalQIP database, the MAC address, Object Name and Domain Name are updated according to the following rules:

1. If the Object Name or Domain Name exists in the DHCP database, and the "Accept Client Names" parameter in the Server Profile is set to True, the Object Name or Domain Name from the DHCP is used to update VitalQIP. The MAC address is also modified.
2. The Object ***must*** exist as a Dynamic Object; otherwise, no changes are made to VitalQIP.

If the Object Name or Domain Name do not exist in the DHCP database the Object Name or Domain Name is not changed within VitalQIP, and the "Accept Client Name" parameter in the Server Profile is set to False, only the MAC address is modified.

Synopsis

```
qip-dhcpsync -n dhcp_svr [-a subnet_address][-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-r reject_file]
```

Parameters

qip-dhcpsync recognizes the following parameters:

-n *dhcp_svr*	Specifies the fully-qualified name of the DHCP server you want to synchronize.
-a *subnet_address*	Specifies a subnet address; omit the parameter to synchronize all subnets.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-r *reject_file*	Specifies the file with the records that were not synchronized.

Command line input example

```
qip-dhcpsync –u qipman –p passwd –n dhcpserver.qtek.com
```

qip-dnsgen

qip-dnsgen creates and distributes DNS configuration and data files for a DNS server. For Windows 2000 DNS servers, this CLI creates and distributes DNS files of "All Records." This CLI will either accept an additional argument that specifies a comma separated list of zones or a file containing the zone names for Lucent DNS 4.x servers.

Synopsis

```
qip-dnsgen -n dns_svr [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-d directory]
   [-l] [-z] [-f] [-t zone_filename]
```

Parameters

qip-dnsgen recognizes the following parameters:

-n ***dns_svr***	Specifies the fully-qualified primary DNS server name.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-d ***directory***	Specifies the directory name where the configuration and data files are stored. If this option is not specified, the files/data are written to the default directory for the DNS server specified. **Note:** **UNIX users**. By default, the nameservers look for the DNS files under */etc*. Therefore, if you wish to generate the DNS files to a directory other than */etc*, you must set up a symbolic link from */etc* to the working DNS directory.
-l	Generates files locally.
-z	Generates files for the changed zones only. If the **–z** parameter is not specified, all zones are generated.
-f	Forces a push to the DNS server if the DNS server was locked previously. If it is still locked, an error message is returned, indicating it is "push locked".
-t ***zone_filename***	This file contains the zones that *qip-dnsgen* uses to generate DNS files. This parameter is only valid for Lucent DNS 4.x.

Command line input examples

```
qip-dnsgen –u qipman –p passwd –n dnssvr.qtek.com –d /usr/etc
qip-dnsgen -u qipman -p passwd -n dnssvr.qtek.com -d /usr/etc -z
```

Zone file example

The following is an example of a forward zone and a reverse zone as they would appear in a zone file:

```
abc.lucent.com
200.200.200.0/24
```

qip-dnsupdate

qip-dnsupdate creates the DNS configuration and data files, increases the serial number in the SOA record, and restarts the server. For Lucent DNS 4.x servers, this CLI creates and distributes DNS files of "Changed Records Only."

Synopsis

```
qip-dnsupdate -n dns_svr [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization] [-d directory]
   [-l] [-z] [-f] [-m] [-t zone_filename]
```

Parameters

qip-dnsupdate recognizes the following parameters:

-n ***dns_svr***	Specifies the fully-qualified name of the primary DNS server.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-d ***directory***	Specifies the directory name where the configuration and data files are stored. If this option is not specified, the files/data are written to the default directory for the DNS server specified. **Note: UNIX Users.** By default, the nameserver looks for the DNS files under */etc*. Therefore, if you wish to generate the DNS files to a directory other than */etc*, you must set up a symbolic link from */etc* to the working DNS directory.
-l	Generates files locally.
-z	Generates files for the changed zones only. If the **–z** parameter is not specified, all zones are generated.
-f	Forces a push to the DNS server if the DNS server was locked previously. If it is still locked, an error message is returned, indicating it is "push locked".
-m	When doing a local push, updates the database as if a server push was specified.
-t ***zone_filename***	This file contains the zones that *qip-dnsgen* uses to generate DNS files. This parameter is only valid for Lucent DNS 4.x.

Command line input example

```
qip-dnsupdate -u qipman -p passwd -n dnsserver.quadritek.com
```

Zone file example

The following is an example of a forward zone and a reverse zone as they would appear in a zone file:

```
abc.lucent.com
200.200.200.0/24
```

qip-folder

qip-folder adds, renames, or deletes folders used for grouping the domains in the Hierarchy of the GUI.

Synopsis

```
qip-folder -ac add|rename|del -n full_path_folder_name
   [-r new_folder_name] [-e errorfile] [-g loginserver]
   [-s servername] [-o organization] [-u username] [-p password]
   [-h] [-v]
```

Parameters

qip-folder recognizes the following parameters:

-ac add|rename|del	The standard command line parameter, **-ac**, can have the following values: **add**, **rename** and **del**. Only one action is allowed per execution. Adding and modifying cannot be performed in the same call. When the value for **-ac** is **add** or **del**, the full path folder name must be provided. When the value for **-ac** is **rename**, both the **-n** full path folder name and the **-r** new folder name must be provided.
-n *full_path_folder_name*	Enter the full path name of the folder to be added, renamed, or deleted.
-r *new_folder_name*	Only use this parameter if the -ac rename action is to be entered. This parameter establishes the new name of the folder.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-e *error_file*	Specifies the file that contains errors generated by this CLI.

Command line input examples

- To create a new folder:

```
qip-folder -ac add -s server1 -u qipman -p passwd -o QIPOrg
   -n topfolder/nextfolder/bottomfolder
```

- To rename the “bottomfolder” to “newbottomfolder”:

```
qip-folder -ac rename -s server1 -u qipman -p passwd -o QIPOrg
   -n topfolder/nextfolder/bottomfolder -r newbottomfolder
```

- To delete an existing folder (the folder must be empty):

```
qip-folder -ac del -s server1 -u qipman -p passwd -o QIPOrg
   -n topfolder/nextfolder/bottomfolder
```

qip-genddnsconfs

qip-genddnsconfs generates the *<org>.DDNS.conf* files for organizations. It is called periodically by the DNS Update Service when the service is on the enterprise server. It accepts the command line parameter **–R** to restart the DNS Update Service when it has completed generating the files. It is installed on the enterprise server.

Synopsis

```
qip-genddnsconfs [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-R]
```

Parameters

qip-genddnsconfs recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-R	Restarts the DNS Update Service when the service is on the enterprise server.

Command line input example

- To generate ddns.conf files for organizations:

```
qip-genddnsconfs
```

- To generate the ddns.conf files for organization 1:

```
qip-genddsnconfs -o 1
```

- To restart the DNS Update Service after generating *ddns.conf* files for organizations:

```
qip-genddnsconfs -R
```

qip-getacltemplate

qip-getacltemplate lists the values for a specific template.

Synopsis

```
qip-getacltemplate [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-f output_file]
   -tn template_name
```

Parameters

qip-getacltemplate recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *output_file*	Specifies the file where the report is stored. The default is *STDOUT*.
-tn *template_name*	Specifies the name of the template being retrieved.

Input file example

This is the input file for **qip-setacltemplate** on page 206. It contains the following elements:

Field	Description
Name	Name of the ACL template
Template Components	Template components include any combination of the following: – IP Addresses – Network Addresses – Localhost

Example

```
testname
135.119.106.0
145.0.0.0/10
```

```
localhost
```

qip-getactiveobjectlst (for use with AutoDiscovery)

qip-getactiveobjectlst constructs a list of all objects matching the Request Profile's list of subnets, and extracts object data for the specified organization and object list as follows:

- Unused objects will be excluded.
- Reserved and Selected objects will be excluded.
- Tombstoned objects will be excluded.
- All other objects with status = Static will be included.
- All other objects with status = Dynamic that are associated with a DHCP server (MAC Address is populated and object has a non-expired lease) will be included.
- Manual BOOTP objects will be included.

Synopsis

```
qip-getactiveobjectlst [-g loginserver] [-s servername]
   -o organization [-u username] [-p password] -i input_file
   [-f output_file]
```

Parameters

qip-getactiveobjectlst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. If not provided, the value of the VitalQIP LOGIN environment variable will be used.
-s *servername*	Specifies the VitalQIP database server. If not provided, the value of the VitalQIP QIPDATASERVER environment variable is used.
-u *username*	Specifies the VitalQIP administrator login. This parameter is required, but is optional if the appropriate value is defined in your policy file (*qip.pcy*)
-p *password*	Specifies the VitalQIP administrator login password. This parameter is required, but is optional if the appropriate value is defined in your policy file (*qip.pcy*).
-o *organization*	***Required.*** Specifies the VitalQIP organization (corporation) name.
-i *input_file*	***Required.*** A directory and filename of the input data. The input data is a list of one or more subnets, one per line, that the CLI will extract from the VitalQIP database. Entries represent network address blocks and bits, for example: 135.222.146.0/24.
-f *output_file*	A directory and file name for the CLI output. If no filename is provided, the output will be to *STDOUT*.

The following information is included for each object in the CLI output:

- ObjectAddress
- MACAddress (formatted with colons)
- ObjectName
- DomainName
- ContactLastName
- ContactFirstName
- ContactEmail
- ContactPhone
- ContactPager
- RoomID
- Street1
- Street2
- City
- State
- Zip
- Country
- Objectclass
- ObjectType-a combination of Allocation Type and Dynamic Type, as follows:
  - Static
  - Dynamic_none
  - Manual_Bootp with non-expired lease
  - Manual_DHCP
  - Automatic_Bootp
  - Automatic_DHCP
  - Dynamic_DHCP
- ObjectDescription

Note: The output file is in XML format.

Command line input example

```
qip-getactiveobjectlst -u qipman -p passwd -i infile.txt -f outfile.txt
```

Input File Example

An input file consists of a list of one or more subnets, one per line, for example:

172.16.7.0/26
172.16.7.64/26
172.16.8.0/26
172.16.8.64/26
172.16.8.128/26

Output example

The following is an output file example for

qip-getactiveobjectlst -o "Active Objects Organization" -u qipman -p passwd -i /import/infile.txt:

```
<qapi-active-obj-list>
<qapi-active-obj  ip-address="172.16.7.65" name="udp000070uds"
   domain="activeobj.net" object-class="Undefined" object-
   type="Dynamic_DHCP">
</qapi-active-obj>
<qapi-active-obj  ip-address="172.16.7.66" name="udp000071uds"
   domain="activeobj.net" object-class="Undefined" object-
   type="Dynamic_DHCP">
</qapi-active-obj>
<qapi-active-obj  ip-address="172.16.7.79" name="udp000084uds"
   domain="activeobj.net" object-class="Undefined" object-
   type="Automatic_DHCP">
</qapi-active-obj>
<qapi-active-obj  ip-address="172.16.7.80" name="udp000085uds"
   domain="activeobj.net" object-class="Undefined" object-
   type="Automatic_DHCP">
</qapi-active-obj>
<qapi-active-obj  ip-address="172.16.7.81" name="udp000086uds"
   domain="activeobj.net" object-class="Undefined" object-
   type="Manual_DHCP">
<mac>87:de:0b:26:00:00</mac>
</qapi-active-obj>
<qapi-active-obj  ip-address="172.16.7.91" name="udp000094uds"
   domain="activeobj.net" object-class="Undefined" object-
   type="Automatic_Bootp">
</qapi-active-obj>
<qapi-active-obj  ip-address="172.16.7.1" name="qipsfv1204"
   domain="activeobj.net" object-class="Server" object-
   type="Static">
<mac>00:03:ba:7a:c9:b7</mac>
<description>Server</description>
<room>2C-335</room>
<contact-first-name>Admin</contact-first-name>
<contact-last-name>Master</contact-last-name>
<contact-email>masteradmin@example.com</contact-email>
<contact-phone>555-8153</contact-phone>
```

```
<contact-pager>12345@example.com</contact-pager>
<street1>600 Somewhere Lane</street1>
<street2>Room A</street2>
<city>Somewhereville</city>
<state>PA</state>
<zip>19355</zip>
<country>USA</country>
</qapi-active-obj>
<qapi-active-obj  ip-address="172.16.7.90" name="udp000093uds"
   domain="activeobj.net" object-class="Undefined" object-
   type="Manual_Bootp">
<mac>87:de:0b:26:00:13</mac>
<description>Manual BOOTP Object</description>
<contact-first-name>Admin</contact-first-name>
<contact-last-name>Master</contact-last-name>
<contact-email>masteradmin@example.com</contact-email>
<contact-phone>555-8153</contact-phone>
<contact-pager>12345@example.com </contact-pager>
</qapi-active-obj>
</qapi-active-obj-list>
```

qip-getaddrrangelst

qip-getaddrrangelst retrieves and lists the defined network address ranges in the VitalQIP database. This CLI command is part of the **Infrastructure|Network** option, and can be executed through the **Address Range** tab of the Network Profile. (For details, refer to Chapter 4 of the *VitalQIP User's Guide*.) The output is stored in the specified (or default) file.

Synopsis

```
qip-getaddrrangelst -t address_type [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-n login_name|-a ip_address] [-f filename]
```

Parameters

qip-getaddrrangelst recognizes the following parameters:

-t ***address_type***	Specifies the address range type (network or subnet).
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-n ***login_name*** **|** **-a** ***ip_address***	Specifies either **-n** (to limit the retrieval to what is managed by a specific administrator) or **-a** (to limit the retrieval to a specific network).
-f ***filename***	Specifies the file where the report is stored. The default is *STDOUT*.

Command line input examples

- To obtain a list of address ranges an administrator manages:

```
qip-getaddrrangelst –u qipman –p passwd –s qipserver –n admin1 -t network
```

- To list the existing address ranges on network "10.10.0.0":

```
qip-getaddrrangelst –u qipman –p passwd –s qipserver –a 10.10.0.0 -t network
```

- To list all address ranges in VitalQIP database and write it to "*addrRange.dat*":

```
qip-getaddrrangelst –u qipman –p passwd  -s qipserver -f addrRange.dat -t
   subnet
```

Output example

```
10.10.0.0,10.10.1.10,10.10.2.200
```

qip-getadminrole

qip-getadminrole displays an administrative role.

Synopsis

qip-getadminrole [**-g** ***loginserver***] [**-s** ***servername***] [**-u** ***username***] [**-p** ***password***] [**-o** ***organization***] **-n** ***role_name*** [**-f filename**]

Parameters

qip-getadminrole recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-n ***role_name***	Specifies the administrative role name you want to display.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getadminrole -u qipman -p passwd -n adminrole -f c:\temp\output.txt
```

Output example

```
RoleName=test role
RoleDescription=This is a test role
Managed List:
Application,Sample Application,True
Subnet Organization,psl_suborg1,True
Domain,usa.world.com,True
Network,net_73,73.0.0.0,True
Server,DHCP-DHCP1,name.com,0.0.0.0,True
Server,DOMAIN CONTROLLER-dc.domain.com,0.0.0.0,True
AddressRange,170.10.0.0-170.10.20.0,False
ObjectRange,14.14.14.1-14.14.14.100,False
```

qip-getapplst

qip-getapplst retrieves the entire list of existing application(s) from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

```
qip-getapplst [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-f filename]
```

Parameters

qip-getapplst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getapplst -u qipman -p passwd -f c:\temp\output.txt
```

Output example

```
"Engineering"
"Computer Club"
```

qip-getbtpsvrlst

qip-getbtpsvrlst retrieves the entire list of fully-qualified Bootp server(s) from the VitalQIP database for the organization specified (if any). The output is stored in the specified (or default) file.

Synopsis

qip-getbtpsvrlst [**-g** ***loginserver***] [**-s** ***servername***] [**-u** ***username***] [**-p** ***password***] [**-o** ***organization***] [**-f** ***filename***]

Parameters

qip-getbtpsvrlst recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getbtpsvrlst –u qipman –p passwd –f c:\temp\output.txt
```

Output example

```
qbootp.qtek.com
ibootp.qtek.com
```

qip-getclientclass

Use the **qip-getclientclass** CLI to obtain a list of client classes and their associated vendor classes or user classes.

Synopsis

```
qip-getclientclass [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization]
   [-n client_class] -f filename
```

Parameters

qip-getclientclass recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used for establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the directory and filename of the input data. If the name contains spaces, it must be enclosed by quotation marks.
-n *client class*	Specifies the client class information.

Output file content example

For a description of the fields in the output file example, refer to Table 1-29, "Input file field definitions" (p. 1-210).

```
ClientClassName=new client class
ClassType=USER
AttachedClass=qa support
OptionTemplate=general
ClientClassPolicyTemplate=policyTemp
```

qip-getcontactlst

qip-getcontactlst retrieves a list of the contact information for all contacts within the VitalQIP database, or for the contacts that match the search criteria specified in the *search_tokens* file. The output is stored in the specified (or default) file.

Synopsis

```
qip-getcontactlst [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-f filename]
   [-t search_tokens_filename]
```

Parameters

qip-getcontactlst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file to contain the retrieved information. The default is *STDOUT*.
-t *search_tokens_filename*	Specifies the name of the file that contains the search criteria to match. If this file is present, the system returns contact information for the contacts that match the search criteria. Otherwise, contact information for *all* contacts are returned. Refer to Table 1-7 for sample search criteria.

Table 1-7 Search token criteria

Search_Tokens **file fields**	**Description**
FirstName	The first name of the contact. **Note:** An implied "Begins with" search is performed. For example, enter **Jo** to return contact information for all contacts having first names that start with "Jo" (such as Joseph, Joanne, Joan).
LastName	The last name of the contact. Note that an implied "Begins with" search is performed. For example, enter **Sm** to return contact information for all contacts having last names that start with "Sm" (such as Small, Smedley, Smith).

The Input and Output examples following are produced by using the **-t** *allcontact.txt* parameter specification. The *allcontact.txt* file contains the following search criteria:

```
LastName=pal
```

Command line input example

```
qip-getcontactlst -u qipman -p passwd -t allcontact.txt
```

Output example

For a description of the fields in the following output file, refer to Table 1-8 following.

```
"2""pal"  "pal"  "pal@qtek.com"  "111-0033"  "1-800-111-3300"
"3""pal1"  "pal,jr I"  "pal1@qtek.com"  "111-3333"  "1-800-111-
  3311"
"4""pal2"  "pal,jr II"  "pal2@qtek.com"  "222-3333"  "1-800-111-
  3322"
"5""pal3"  "pal3"  "pal3@qtek.com"  "333-3333"  "1-800-111-3333"
"7""pal5"  "pal5"  "pal5@qtek.com"  "111-3555"  "1-800-111-3355"
"8""pal6"  "pal6"  "pal6@qtek.com"  "111-3366"  "1-800-111-3366"
"9""pal7"  "pal"  "pal7@qtek.com"  "111-3377"  "-800-111-3377"
```

Table 1-8 Output field description

Field	Description
ContactID	The identification of the contact.
LastName	The contact's last name.
FirstName	The contact's first name.
EmailAddress	The contact's email address.
PhoneNumber	The contact's phone number.
PagerNumber	The contact's pager number.

qip-getdebuglevel

qip-getdebuglevel retrieves the current debug level for a DHCP server, if one is set. If the debug level is set, the CLI returns one of the following values:

- levelCritical
- levelDebug
- levelError
- levelInfo
- levelWarning

Synopsis

```
qip-getdebuglevel -n dhcp_servername [-g loginserver]
   [-s dataserver] [-u username] [-p password] [-o organization]
   [-f output_file]
```

Parameters

qip-getdebuglevel recognizes the following parameter

-n ***dhcp_servername***	Specifies the DHCP server name for which you want to obtain the debug level.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***output_file***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getdebuglevel -n qipserver
```

Output example

```
levelInfo
```

qip-getdecnetaddr

qip-getdecnetaddr retrieves the next available DECNet address from a specified DECNet area of the VitalQIP database.

Synopsis

```
qip-getdecnetaddr -a decnet_area [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization]
```

Parameters

qip-getdecnetaddr recognizes the following parameters:

-a *decnet_area*	Specifies the DECNet area where you want to retrieve the DECNet address, from a DECNet node value of 1-63.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.

Command line input example

```
qip-getdecnetaddr -u qipman -p passwd -a 10
```

Output example

```
DECNetNode=6
DECNetAddress=AA00040000604
```

qip-getdhcpscopes

qip-getdhcpscopes provides a list of all configure DHCP address ranges for the specified organization.

Synopsis

qip-getdhcpscopes [**-g** ***loginserver***] [**-s** ***servername***] **-o** ***organization*** [**-u** ***username***] [**-p** ***password***] [**-f** ***output_file***]

Parameters

qip-getdhcpscopes recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. If not provided, the value of the VitalQIP LOGIN environment variable will be used.
-s ***servername***	Specifies the VitalQIP database server name. If not provided, the value of the VitalQIP QIPDATASERVER environment variable shall be used.
-u ***username***	Specifies the VitalQIP administrator login. This parameter is required but is optional if the appropriate value is defined in your policy file (*qip.pcy*).
-p ***password***	Specifies the VitalQIP administrator login password. This parameter is required but is optional if the appropriate value is defined in your policy file (*qip.pcy*).
-o ***organization***	***Required***. Specifies the VitalQIP organization name.
-f ***output_file***	A directory and file name for the CLI output. If no filename is provided, the output will be to *STDOUT*.

Command line input example

```
qip-getdhcpscopes -u qipman -p passwd -o "Active Objects
   Organization" -f extfile.txt
```

Output example

The following is an output file example.

```
<qapi-range-list>
         <qapi-range  start-addr="111.222.11.1" end-
   addr="111.222.11.1">
      </qapi-range>
         <qapi-range  start-addr="172.16.7.90" end-
   addr="172.16.7.90">
      </qapi-range>
         <qapi-range  start-addr="172.16.7.65" end-
   addr="172.16.7.75">
```

```
        </qapi-range>
           <qapi-range  start-addr="172.16.7.76" end-
addr="172.16.7.80">
        </qapi-range>
           <qapi-range  start-addr="172.16.7.81" end-
addr="172.16.7.85">
        </qapi-range>
           <qapi-range  start-addr="172.16.7.91" end-
addr="172.16.7.91">
        </qapi-range>
</qapi-range-list>
```

qip-getdhcpsubnetlst

qip-getdhcpsubnetlst makes a list of the subnets that are serviced by a specific DHCP server. The output is stored in the specified (or default) file.

Synopsis

```
qip-getdhcpsubnetlst -n dhcp_server_name [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-f filename]
```

Parameters

qip-getdhcpsubnetlst recognizes the following parameters:

-n ***dhcp_server_name***	Specifies the fully-qualified DHCP server name from which you are retrieving subnets.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getdhcpsubnetlst -u qipman -p passwd -n dhcpserver1.com -f
   dhcpsubnetlst.txt
```

Output example

```
144.144.144.0
198.200.234.0
```

qip-getdhcpsvrlst

qip-getdhcpsvrlst retrieves the entire list of existing DHCP server(s) from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

```
qip-getdhcpsvrlst [-g loginserver] [-s servername] [-u username]
   [-p password] [-n subnet_address] [-o organization]
   [-f filename] [-all] [-b show_server_type(Y/N)]
```

Parameters

qip-getdhcpsvrlst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-n *subnet_address*	Subnet address that the DHCP server(s) manage.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.
-all	Displays all primary and secondary servers. If this parameter is omitted, only primary servers are displayed.
-b *show_server_type*	Specifies if the server type should appear in the output. Allowable values are **y** - show the server type **n** - do not show the server type

Command line input example

```
qip-getdhcpsvrlst –u qipman –p password -b y -f c:\temp\output.txt
```

Output example

```
dhcp1.qtek.com,190.200.138.24,Lucent DHCP 5.4
dhcp2.quadritek.com,190.201.123.01
```

qip-getdnsrr

qip-getdnsrr retrieves the resource record(s) associated with an object, domain, or reverse zone. The output is stored in the specified (or default) file.

Synopsis

```
qip-getdnsrr -t owner_type -n owner_name |-a address/mask
   [-g loginserver] [-s servername] [-u username] [-p password]
   [-o organization] [-f filename]
```

Parameters

qip-getdnsrr recognizes the following parameters:

-t *owner_type*	Specifies the owner type of the resource records, object, domain, or reverse zone.
-n *owner_name* | -a *address/mask*	Specifies either the fully-qualified domain name for which you want to retrieve resource records *or* IP address of the object or reverse zone for which you want to retrieve resource records.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getdnsrr -t domain -u qipman -p passwd -n quadritek.com -f rr.lst
```

Output example

```
ResourceRecOwner=domain
ResourceRecType=A
ResourceClass=IN
MinimumTTL=23343
ResourceRecText=198.200.138.16
ChangeFlag=
ExternalFlag=0
Tombstoned=0
ExternalComment="Comment"
ExternalTimestamp=3/16/2005 19:59:36
```

Table 1-9 Output field definitions

Field Name/Label	Description
Note: The following fields appear once per resource record.	
ResourceRecOwner	The owner of the resource record as defined in RFC1035.
ResourceRecType	The type of resource record (for example, A, PTR) as defined in RFC 1035.
ResourceClass	The resource record class (for example, IN, CS, CH, HS) as defined in RFC 1035.
MinimumTTL	Specifies the time interval (time to live) that the resource record can be cached before the source of the information should again be consulted. A zero value is interpreted to mean that the resource record can only be used for the transaction in progress, and should not be cached. For example, SOA records are always distributed with a zero TTL to prohibit caching. Zero values can also be used for extremely volatile data. If no TTL is desired (for example, it never expires), leave this field blank. **Note:** A value of –1 indicates no TTL has been specified.
ResourceRecText	The data area of the resource record as defined in RFC 1035.
ApplyToZone	The zone where the record should be added: 1 - forward zone 0 - reverse zone **Note:** This field only appears in an output file generated with **`-t`** ***`owner_type`*** set to **`object`**.
ChangeFlag	Always returns a null value, so if a file generated by **qip-getdnsrr** is used as input to **qip-setdnsrr**, all records are skipped.
External Flag	Indicates how the record was updated 0 - Updated by VitalQIP 1 - External update
Tombstoned	0 indicates the object was not tombstoned. 1 indicates the object was tombstoned.
External Comment	A comment indicating the history of this external add.
External Timestamp	Specifies the actual date and time.

qip-getdnssvrlst

qip-getdnssvrlst retrieves the entire list of existing fully-qualified DNS server(s) under the specified domain or reverse zone from the VitalQIP database. IP addresses are also listed once the server objects have been defined. The output is stored in the specified (or default) file.

Synopsis

```
qip-getdnssvrlst [-n domain_name|-a reverse_zone_address] [-g
   loginserver] [-s servername] [-u username] [-p password]
   [-o organization] [-f filename] [-b show_server_type]
```

Parameters

qip-getdnssvrlst recognizes the following parameters:

-n ***domain_name*** | **-a** ***reverse_zone_address***	Specifies the fully-qualified domain name of the primary DNS server or the IP address of a reverse zone for the primary servers.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.
-b ***show_server_type***	Specifies if the server type should appear in the output. Allowable values are **y** - show the server type **n** - do not show the server type

Command line input examples

```
qip-getdnssvrlst -u qipman -p passwd -n quadritek.com -f c:\temp\output.txt

qip-getdnssvrlst -u qipman -p passwd -a 211.211.211.0

qip-getdnssvrlst -u qipman -p passwd
```

Output example

```
DNS1.quadritek.com,192.168.20.25
DNS2.quadritek.com
```

qip-getdomnlst

qip-getdomnlst retrieves a domain list or an associated domain(s) for a particular subnet or folder from the VitalQIP database. If the subnet address is provided, only the domain(s) associated with that subnet or folder is retrieved. If the folder name is provided, the domains and subfolders associated with that folder are retrieved. The output is stored in the specified (or default) file.

Synopsis

qip-getdomnlst [**-g** ***loginserver***] [**-s** ***servername***] [**-u** ***username***] [**-p** ***password***] [**-o** ***organization***] [**-a** ***subnet_address***] [**-n** ***folder_name***] [**-f** ***filename***]

Parameters

qip-getdomnlst recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-a ***subnet_address***	Specifies the subnet address for the domains for which you want to retrieve domain lists.
-n ***folder_name***	Specify the foldername. **Note:** In the output for this CLI, the folders are displayed with a ",f" following the name to differentiate them from domains (for example, foldername/subfolder,f).
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input examples

```
qip-getdomnlst -u qipman -p passwd -a 188.200.138.211 -f c:\temp\output.txt

qip-getdomnlst -u qipman -p passwd -f c:\output.txt
```

Output example

```
qip.quadritek.com
qa.quadritek.com
ra.qtek.com
```

qip-getfolderlst

The use of grouping domains into folders allows for easier, more efficient management of large numbers of domains in VitalQIP. Domain folders are shown in a hierarchical tree in a number of places in the GUI, if you have opted to use them (handled through a policy option for Administrators). **qip-getfolderlst** retrieves *all* existing domain folders in the specified organization, or only the top level folder in the hierarchy from the VitalQIP database. The output is stored in the specified (or default) file.

For more information on domain folders and their use, refer to Chapter 4 of the *VitalQIP User's Guide*.

Synopsis

qip-getfolderlst [**-g** ***loginserver***] [**-s** ***servername***] [**-u** ***username***] [**-p** ***password***] [**-o** ***organization***] [**-f** ***filename***]

Parameters

qip-getfolderlst recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

- To retrieve all folders in the VitalQIP Organization and store it in *folder.lst*:

```
qip-getfolderlst –u qipman –p passwd –o "VitalQIP Organization" –f folder.lst
```

Output example

```
Folder1
TopFolder/nextFolder
TopFolder/nextFolder/bottomFolder
```

qip-getfreesubnetlst

qip-getfreesubnetlst retrieves all available existing or new subnets in the specified network address with a specified subnet mask from the VitalQIP database. The default is to retrieve the existing available subnets. The output is stored in the specified (or default) file.

Synopsis

```
qip-getfreesubnetlst [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization]
   -a network_address -m subnet_mask [-t existing|new]
   [-f filename]
```

Parameters

qip-getfreesubnetlst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-a *network_address*	Specifies the IP address of the network from which you want to retrieve available subnets.
-m *subnet_mask*	Specifies the subnet mask to be applied.
-t existing|new	Specifies whether to retrieve the available existing or new subnets. Type either **e** (or **existing**) *or* **n** (or **new**) to retrieve the available existing or new subnets. The default is "existing".
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example 1

- To retrieve all NEW free subnets from a specified network with a specified subnet mask:

```
qip-getfreesubnetlst -u qipman -p passwd -t new -a 144.144.0.0 -m
   255.255.255.0
```

Output example 1

This output lists all the possible free subnets, and contains only the addresses because subnets have not been used/named.

```
144.144.234.0
144.144.235.0
144.144.236.0
```

Command line input example 2

- To retrieve all free subnets from a specified network with a specified subnet mask to a file named *subnet.txt*:

```
qip-getfreesubnetlst -u qipman -p passwd -a 144.144.0.0 -m 255.255.255.0 -f
   subnet.txt
```

Output example 2

```
"144.144.1.0" "sn144 1" "suborg 1"
"144.144.10.0" "sn144 10" "suborg 1"
```

Output field definition

In Command line input example 2, the `-f` parameter specifies *subnet.txt*. Therefore, the output is listed in the datafile *subnet.txt*. The fields are described further in Table 1-10:

Table 1-10 Subnet field definitions

Field	Description
Subnet Address	The free Subnet Address.
Subnet Name	The name of the free Subnet Address.
Subnet Organization	Subnet Organization that this free subnet belongs to.

qip-gethublst

qip-gethublst retrieves a list of all Wiring_HUBs of the specified subnet from the VitalQIP database for the subnet address specified. The output is stored in the specified (or default) file.

Synopsis

```
qip-gethublst -a subnet_address [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization] [-f filename]
```

Parameters

qip-gethublst recognizes the following parameters:

-a *subnet_address*	Specifies the subnet address from which you want to retrieve a list of Wiring_HUBs.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-gethublst -u qipman -p passwd -o "VitalQIP Organization" -a 101.200.111.0
   -f gethub.lst
```

Output example

```
Ny_hub.qtek.com
Jersey_hub.qtek.com
```

qip-gethubport

qip-gethubport retrieves available Hub ports and numbers, and marks the port number as "selected". The hub name must be a fully-qualified name. The output is stored in the specified (or default) file.

Synopsis

```
qip-gethubport -n hub_name -l slot_name [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-f filename]
```

Parameters

qip-gethubport recognizes the following parameters:

-n ***hub_name***	Specifies the fully-qualified wiring hub name for which you want to retrieve port numbers.
-l ***slot_name***	Specifies the wiring hub slot name for which you want to retrieve hub port numbers.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-gethubport -u qipman -p passwd -n hub1.qtek.com -l slotthree
```

Output example

```
HubPort=345
HubPort=113
```

qip-gethubslotlst

qip-gethubslotlst retrieves the entire slot list of the specified Wiring_HUB from the VitalQIP database. The hub name must be a fully-qualified name. The output is stored in the specified (or default) file.

Synopsis

```
qip-gethubslotlst -n hub_name [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization]
   [-f output_filename]
```

Parameters

qip-gethubslotlst recognizes the following parameters:

-n *hub_name*	Specifies the fully-qualified Wiring_HUB name from which you want to retrieve Wiring_HUB slot information.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-gethubslotlst –u qipman –p passwd -n hub1.qtek.com –f
   hubslot.lst
```

Output example

```
"Slot1" 111
"Slot3" 333
```

qip-getipaddr

qip-getipaddr retrieves the next free IP address within a subnet, and marks the address as "selected". The output is stored in the specified (or default) file.

Synopsis

```
qip-getipaddr -a subnet_address [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization] [-f filename]
```

Parameters

qip-getipaddr recognizes the following parameters:

-a *subnet_address*	Specifies the subnet address from which you want to retrieve a free address.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getipaddr –u qipman –p passwd -a 101.200.100.0
```

Output example

```
IPAddress=101.200.100.21
```

qip-getlocalsvrlst

qip-getlocalsvrlst retrieves the entire list of existing fully-qualified Local Host Servers from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

```
qip-getlocalsvrlst [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-f filename]
```

Parameters

qip-getlocalsvrlst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getlocalsvrlst -u qipman -p passwd -f localsvr.lst
```

Output example

```
Local.qtek.com
LHost.qtek.com
```

qip-getloclst

qip-getloclst retrieves a list of all existing locations from the VitalQIP database; or (if the **-t** parameter is specified), it retrieves a list of locations that match the search criteria file. The output is stored in the specified (or default) output file.

Synopsis

```
qip-getloclst [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-f filename]
   [-t search_tokens_filename]
```

Parameters

qip-getloclst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the directory and filename for the output data. The default is *STDOUT*.
-t *search_tokens_filename*	Specifies the name of the input file that contains the search criteria to match. If this file is present, the system returns a list of all locations that match the search criteria. Refer to below for sample search criteria.

Specify the information on which you want to search for locations. If you do not specify search criteria, all locations are returned. For example, your *Search_Tokens_File* may consist of the following fields, as listed in Table 1-11:

Table 1-11 Search token file fields

Search_Tokens_File fields	Description
LocationID	Unique ID representing the Location.

Search_Tokens_File fields	Description
Street1	The Street1 field of the location. Note that an implied “Begins with” search is performed. For example, if you enter **10**, all the streets that start with “10” are returned.
Street2	The Street2 field of the location. Note that an implied “Begins with” search is performed. For example, if you enter **Su**, all the streets that start with “Su” are returned.
City	The City of the location Note that an implied “Begins with” search is performed. For example, if you enter **Ma**, all the names that start with “Ma” are returned.
State	The State of the location. Note that an implied “Begins with” search is performed. For example, if you enter **P**, all the names that start with “P” are returned.
Zip	Zip of the location. Note that an implied “Begins with” search is performed. For example, if you enter **19**, all the names that start with “19” are returned.
Country	Country of the location. Note that an implied “Begins with” search is performed. For example, if you enter **US**, all the names that start with “US” are returned.

The *location.lst* file, as shown in the Input and Output examples shown below, may contain:

```
Country=usa
```

Command line input example

```
qip-getloclst -u qipman -p passwd -f location.lst
```

Output example

```
"9""20"  "Old Still Rd"  "Woodbridge"  "CONN"  "02644"  "usa"
"8""5"  "Whitney Ave"  "New Haven"  "Ct"  "02488"  "usa"
"2""400"  "Lapp Road"  "Malvern"  "PA"  "19355"  "usa"
"10""1103"  "Clover Hill Drive"  "East Bradford"  "PA"  "19382"  "usa"
"1""20"  "Lapp Road"  "Malvern"  "Penn"  "19355"  "usa"
"4""5"  "Great Valley Parkway"  "Malvern"  "Penn"  "19355"  "usa"
"3""5"  "Lapp Road"  "Malvern"  "Penn"  "19355"  "usa"
"6""5"  "Valley Road"  "Christiana"  "Penn"  "19855"  "usa"
```

qip-getmaclst

qip-getmaclst retrieves the entire list of manufacturers for a particular object class from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

```
qip-getmaclst -c object_class [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization] [-f filename]
```

Parameters

qip-getmaclst recognizes the following parameters:

-c *object_class* Specifies the object class for which you want to retrieve the list of manufacturers. The available options are:

- Workstation
- X-terminal
- PC
- Printer
- Server
- Wiring_HUB
- Router
- Bridge
- Terminal_Server
- Switch
- Legacy_System
- Gateway
- Test_Equipment
- Any user-defined object class
- Dynamic_Allocation
- Undefined
- Others

-g *loginserver* Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.

-s *servername* Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.

-u *username* Specifies the VitalQIP administrator account to be used in establishing the database connection.

-p *password* Specifies the password for the associated administrator account.

-o *organization* Specifies the VitalQIP organization (corporation) name.

-f *filename* Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getmaclst -u qipman -p passwd -c PC -f mac.lst
```

Output example

```
Gateway-2000
compac
```

qip-getmacmodellst

qip-getmacmodellst retrieves all model lists of the specified manufacturer from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

```
qip-getmacmodellst -m manufacturer -c object_class [-g
   loginserver] [-s servername] [-u username] [-p password]
   [-o organization] [-f filename]
```

Parameters

qip-getmacmodellst recognizes the following parameters:

-m *manufacturer* Specifies the manufacturer for which you want to retrieve model lists.

-c *object_class* Specifies the object class for which you want to retrieve the list of manufacturers. The available options are:

- Workstation
- X-terminal
- PC
- Printer
- Server
- Wiring_HUB
- Router
- Bridge
- Terminal_Server
- Switch
- Legacy_System
- Gateway
- Test_Equipment
- Any user-defined object class
- Dynamic_Allocation
- Undefined Others

-g *loginserver* Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.

-s *servername* Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.

-u *username* Specifies the VitalQIP administrator account to be used in establishing the database connection.

-p *password* Specifies the password for the associated administrator account.

-o *organization* Specifies the VitalQIP organization (corporation) name.

-f *filename* Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getmacmodellst -u qipman -p passwd -c PC -m compaq -f
   modellist.txt
```

Output example

```
Compaq presario
PrefixMACAddress=00008E
Compaq progsignia
PrefixMACAddress=00805F
```

qip-getmacpools

qip-getmacpools configures the global and subnet DHCP Server MAC pools, for both the inclusion and exclusion pools.

Synopsis

```
qip-getmacpools -a subnet_address | -n dhcp_server_name
   [-g loginserver] [-s servername] [-u username] [-p password]
   [-o organization] [-f filename]
```

Parameters

qip-getmacpools recognizes the following parameters:

-a *subnet_address*	Specifies the subnet address where MAC Pools are being configured.
-n *dhcp_servername*	Specifies the DHCP server name where MAC Pools are being configured.
-g *loginserver*	Specifies the login server associated with the MAC Pools.
-s *servername*	Specifies the server name where the MAC Pool is located.
-u *username*	Specifies the username of the MAC Pool administrator.
-p *password*	Specifies the password of the MAC Pool administrator
-o *organization*	Specifies the organization with which the MAC Pool is associated.
-f *filename*	Specifies the filename that contains the output from this CLI.

Output example 1

```
DHCP Server Name=testdhcp.qtek.com
I, Ethernet, 000001243fa7
E, IEEE802, 00000c1703ea
I, Token Ring, 100*
```

Output example 2

```
Subnet Address=200.100.10.1
I,Chaos,321321*
E,Pronet,aabbccddeeff
I,Ethernet,1233211233211234
E,Token Ring,222*
```

Note: In the above output examples, I indicates that the MAC address is Included, E indicates that the MAC address is Excluded.

qip-getnetlst

qip-getnetlst retrieves a list of all networks from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

```
qip-getnetlst [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-f filename]
```

Parameters

qip-getnetlst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getnetlst –u qipman –p passwd –f netlst.ext
```

Output example

The first column in the output file contains the network address, the second column contains the network name.

```
"144.143.0.0"    "Malvern"
"150.50.0.0"     "New Jersey"
"198.200.138.0"  "Atlanta"
```

qip-getnissvrlst

qip-getnissvrlst retrieves a list of existing NIS servers for the specified organization from the VitalQIP database. If no organization is specified, a list of NIS servers for the VitalQIP organization is returned. The output is stored in the specified (or default) file.

Synopsis

```
qip-getnissvrlst [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-f filename]
```

Parameters

qip-getnissvrlst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getnissvrlst –u qipman –p passwd –f nissvr.lst
```

Output example

```
nis1.qtek.com
nis2.qtek.com
```

qip-getnmdnsserver

qip-getnmdnsserver retrieves the attributes (profile) of a non-managed DNS server from the VitalQIP database. The fully-qualified QIP managed DNS server associated with this server must be specified. The output is stored in the specified (or default) file.

Note: When **-m** is used without **-a** and **-n** options, the attributes are returned for all non-managed servers that are associated with the server specified in **-m**. When **-n** and **-a** options are used, only the attributes for the non-managed server that is specified in **-n -a** are returned. If options **-n -a** are used, both options must be used together. When **-m**, **-n**, and **-a** options are used together, the same results are returned as are returned when using the **-n -a** options.

Synopsis

```
qip-getnmdnsserver -m qip_managed_svr_name|[-n nm_svr_name
   -a nm_svr_ip] [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-f filename]
```

Parameters

qip-getnmdnsserver recognizes the following parameters:

-m ***qip_managed_svr_name***	Specifies the fully-qualified VitalQIP managed DNS server that is associated with the non-managed DNS servers.
-n ***nm_svr_name*** **|** **-a** ***nm_svr_ip***	Specifies the fully-qualified non-managed DNS server name for the selected attributes when **-n** is used. Specifies the IP address of the non-managed DNS server for the selected attributes when **-a** is used.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input examples

```
qip-getnmdnsserver -u qipman -p passwd -n dns1.qtek.com -a 100.100.100.100
```

```
qip-getnmdnsserver -u qipman -p passwd -m qip.quadritek.com
```

Output example

Note: For a description of the fields in this output example and more information on non-managed DNS servers, refer to “Non-managed DNS servers” in Chapter 4 of the *VitalQIP User’s Guide*.

```
ServerName=DNS1.qtek.com
NMServerName=dns1Sec.qtek.com
NMServerAddress=150.11.22.33
DomainName=pa.qtek.com
ReverseZoneAddress=100.100.100.0/24
```

qip-getobjectlst

qip-getobjectlst retrieves a list of all objects (whether their status is “used’ or “unused”) of the specified subnet from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

```
qip-getobjectlst -a subnet_address [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-f filename]
```

Parameters

qip-getobjectlst recognizes the following parameters:

-a *subnet_address*	Specifies the subnet address from which you want to retrieve objects.
-g *loginserver*	Specifies the VitalQIP login server’s IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getobjectlst -u qipman -p passwd -a 199.201.139.0 -f
   objectlst.txt
```

Output example

For a description of the fields in the output example, refer to Table 1-12.

```
"10.98.4.10" "BHM010" "Workstation" "Static" "qtek.com"
"10.98.4.11" "BHM020" "Workstation" "Static" "qtek.com"
"10.98.4.12" "BHM030" "Workstation" "Static" "qtek.com"
"10.98.4.13" "BHM040" "Workstation" "Static" "qtek.com"
"10.98.4.14" "BHM050" "Undefined"   "Static" "qtek.com"
```

Table 1-12 Object list field definitions

Field	Description
IP Address	IP Address of Object
Host Name	Host Name of object
Object Class	Object Class of object: Workstation, X-terminal, PC, Printer, Server, Wiring_HUB, Router, Bridge, Terminal_Server, Switch, Legacy_System, Gateway, Test_Equipment, User-defined, Undefined, Others
Object Status	Static, Unused, Dynamic, M-DHCP, M-BOOTP, D-DHCP, A-DHCP, A-BOOTP
Domain	Default Domain of object, if assigned.

qip-getobjectprof

qip-getobjectprof retrieves the Object Profile of a specified object from the VitalQIP database. The **-a** parameter is required for the command to retrieve data for the object. Table 1-13 describes the possible field values. The output is stored in the specified (or default) file.

Synopsis

```
qip-getobjectprof -a object_address [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-f filename]
```

Parameters

qip-getobjectprof recognizes the following parameters:

-a *object_address*	Specifies the object address from which you want to retrieve the Object Profile.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getobjectprof -a 199.200.139.27 -u qipman -p passwd -f object prof.txt
```

Output example

For a description of the fields in the output example, refer to Table 1-13, "Object profile field definitions" (p. 1-105).

```
ObjectAddress=144.144.144.2
SubnetAddress=144.144.144.0
ObjectName=ws2
DomainName=quadritek.com
ObjectClass=PC
ExpiredDate=
ServerType=None
```

```
Application=
MACAddress=
ObjectTag=
LocationID=2
RoomID=
Street1=400
Street2=Lapp Road
City=Malvern
State=PA
Zip=19355
Country=usa
Manufacturer=
ModelType=
SerialNo=
AssetNo=
HostID=
PurchaseDate=
ObjectDescription=this is my pc
HubName=whp000002whs.quadritek.com
SlotName=hub-5
PortNum=1
ContactID=10
ContactLastName=newPerson
ContactFirstName=personF
ContactEmail=
ContactPhone=
ContactPager=
RouterGroup=
DynamicConfig=
TftpServer=
BootFileName=
HardwareType=
Aliases=Workstation-ws2 Wws2 aliasws
MailForwarders=WS1.quadritek.com(2)
MailHosts=ws3.quadritek.com(1)
HubSlots=
DNSServers=
TimeServers=
DefaultRouters=144.144.144.100
UserClasses=
Users=pal3
NameService=A,PTR
DynamicDNSUpdate=A,PTR,CNAME,MX
DHCPServer=
DHCPOptionTemplate=
```

```
DHCPPolicyTemplate=
LeaseTime=0
VendorClass=
TTLtime=100
DECNetArea=10
DECNetNode=1
DECNetAddress=AA0004000128
DualProtocol=DECNet
Tombstoned=1
ExternalComment="Comment"
ExternalTimestamp=03/09/2005 20:37:42
UserClass=MS Vendor
AllowDHCPClientsModifyDynamicObjectResourceRecords=True
```

Table 1-13 Object profile field definitions

Field name/label	Description	
ObjectAddress	The IP Address of the object.	
SubnetAddress	The subnet address of the object.	
ObjectName	The hostname of the object. Not fully-qualified.	
DomainName	The domain name of the object.	
ObjectClass	The object class of the object can be:	
	- Workstation - X-terminal - PC - Printer - Server - Wiring_Hub - Router	- Bridge - Terminal Server - Switch - Legacy System - Gateway - Test_Equipment - Any user-defined object class
ExpiredDate	The expiration date of the lease.	
ServerType	Server type. Only applicable if ObjectClass is "server": - TimeServer – if object is a time server. - TftpServer – if object is a Tftp server. - TftpServer&TimeServer – if object is a TftpServer and TimeServer	
Application	The Primary Application of the object.	
MACAddress	The MAC Address of the object.	
ObjectTag	Tag field of the object.	
LocationID	The location ID field of the object.	

Field name/label	Description
RoomID	Room ID field of the object.
Street1	Street1 field of the location of the object.
Street2	Street2 field of the location of the object.
City	City of the location of the object.
State	State of the location of the object.
Zip	Zip of the location of the object.
Country	Country of the location of the object.
Manufacturer	Manufacturer Name of the object.
ModelType	Model Name of the object.
SerialNo	Serial Number of the object.
AssetNo	Asset Number of the object.
HostID	Host ID of the object.
PurchaseDate	Purchase Date/Time of the object. Must be in mm/dd/yyyy HH:MM format. (for example, 09/22/1998 00:00)
ObjectDescription	Object Description of the object.
HubName	Hub Name(s) assigned of the object.
SlotName	Slot Name(s) assigned of the object.
PortNum	Port Numbers assigned of the object.
ContactID	The contact ID field of the object.
ContactLastName	Contact last name of the object.
ContactFirstName	Contact first name of the object.
ContactEmail	Contact email address of the object.
ContactPhone	Contact phone number of the object.
ContactPager	Contact pager number of the object.
RouterGroup	Router Group of the object. Only applicable if the object class is “router”.
DECNetArea	DECNet Area of the object. (Valid values: 1-63) Only applicable if Dual Protocol set to DECNet. The DECNet Address is returned based on the DECNet Area and DECNet Node.

Field name/label	Description
DECNetNode	DECNet Node of the object. (Valid values: 1-1024) Only applicable if Dual Protocol is set to DECNet. The DECNet Address is returned based on the DECNet Area and DECNet Node.
DECNetAddress	DECNet address of the object, which is derived from the object's DECNet Area. This field is for display only.
IPXNetworkNumber	IPX Network number of the object. Only applicable if Dual Protocol is set to IPX.
IPXNode	IPX Node of the object. Only applicable if Dual Protocol set to IPX.
DynamicConfig	Dynamic Configuration of the Object: - Blank – Static object - None – dynamic (none) - M-DHCP – Manual DHCP - A-DHCP – Automatic DHCP - D-DHCP – Dynamic DHCP - M-BOOTP – Manual Bootp - A-BOOTP – Automatic Bootp
TftpServer	TFTP Server of object. Only applicable if object type is M-BOOTP.
BootFileName	Bootfile name of object. Only applicable if object type is M-BOOTP.
HardwareType	Hardware Type of object. Only applicable for M-BOOTP objects: - Ethernet - TokenRing - AX.25 - Pronet - Chaos - IEEE802 - Arcnet
Aliases	Alias names. If multiple, separate by space (for example, www www3).
MailForwarders	Fully-qualified host name with priority, for example, mailF.qtek.com(10)
MailHosts	Fully-qualified host name with priority, for example, mailH.qtek.com(100)

Field name/label	Description
Users	User login names. To delete a user, place a (d) after the login name.
UserClass	User class, if assigned. Only applicable for QDHCP server objects.
HubSlots	List of slot names (port numbers) if the object is a Wiring_HUB.
DNSServers	DNS Servers attached to this object. If multiple, separate by space. Only applicable for M-BOOTP objects.
TimeServers	Time Servers attached to this object. If multiple, separate by space. Only applicable for M-BOOTP objects.
DefaultRouters	IP Address(es) of Default routers. If multiple, separate by space.
NameService	Name Services selection: No – Name Services checked off. Yes – Name Services checked on. A – Name Services checked on, A record checked on. PTR - Name Services checked on, PTR record checked on. A,PTR – Name Services checked on, A record checked on, PTR record checked on.
DynamicDNSUpdate	Dynamic Update selection: - A – Dynamic updates for A records. - PTR – Dynamic updates for PTR records. - CNAME – Dynamic updates for CNAME records. - MX – Dynamic updates for MX records. **Note:** Multiple selections can be accomplished by specifying each component, separated by comma. (for example, A,PTR,CNAME)
DHCPServer	The DHCP Server assigned to this object. Only applicable for dynamic objects.
DHCPTemplate	The DHCP Template assigned to this object. Only applicable for dynamic objects.
LeaseTime	The lease time (in seconds) assigned to this object. Only applicable for dynamic objects.
VendorClass	The Vendor Class assigned to this object. Only applicable for dynamic objects.
TTLtime	Time to live (TTL) value of the object, in seconds. **Note:** Since 0 is a valid value, a value of –1 indicates no TTL set.

Field name/label	Description	
NetbiosName	NetBIOS Name. Only applicable when dual protocol set to NETBIOS.	
NetbiosDomain	NetBIOS Domain. Only applicable when dual protocol set to NETBIOS.	
DualProtocol	Dual Protocol of object: - None - DECNet - IPX - NETBIOS	
Tombstoned	0 indicates the object was not tombstoned. 1 indicates the object was tombstoned.	
External Comment	A comment indicating the history of this external add.	
External Timestamp	Specifies the actual date and time.	
AllowDHCPClients ModifyDynamicObject ResourceRecords	True False Same as in Global Policies	

qip-getobjname

qip-getobjname retrieves the next free default object name of the given object class. For more information on the default object name, refer to the "Naming Policies" section in Chapter 2 of the *VitalQIP User's Guide*. The output is stored in the specified filename.

Synopsis

```
qip-getobjname -c object_class [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization] [-f filename]
```

Parameters

qip-getobjname recognizes the following parameters:

-c *object_class*	Specifies the object class for which you want to retrieve the list of manufacturers. The following options are available (note that the first letter must be a capital letter, for example enter Router not router):

- Workstation	- Terminal_Server
- X-terminal	- Switch
- PC	- Legacy_System
- Printer	- Gateway
- Server	- Test_Equipment
- Wiring_HUB	- Any user-defined object class
- Router	- Undefined
- Bridge	- Others

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getobjname -u qipman -p passwd -c Workstation
```

Output example

```
ObjectName=wsp000010wss
```

qip-getorganization

qip-getorganization retrieves the list of organizations, or specific details about one organization. The output is stored in the specified (or default) file.

Synopsis

```
qip-getorganization [-g loginserver] [-s servername]
   [-u username] [-p password] [-f filename] [-i]
   [-o organization]
```

Parameters

qip-getorganization recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the organization (corporation) name. If the **-o** parameter is used, specific information about the organization is returned. If the **-o** parameter is not specified a list of organizations is returned, as shown in the output example below.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.
-i	Displays all organization IDs. The ID is displayed in the form *<orgid>*, *<orgname>*.

Command line input example

```
qip-getorganization –u qipman –p passwd -i
```

Output example

```
New Organization
VitalQIP Organization
```

qip-getospflst

qip-getospflst retrieves a list of all OSPF areas from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

```
qip-getospflst [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-f filename]
```

Parameters

qip-getospflst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getospflst –u qipman –p passwd –f ospflst.txt
```

Output example

```
BACKBONE
rew_ospf
```

qip-getospfprof

qip-getospfprof retrieves the profile of an OSPF area from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

```
qip-getospfprof -n ospf_area_name [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-f filename]
```

Parameters

qip-getospfprof recognizes the following parameters:

-n ***ospf_area_name***	Specifies the name of the OSPF area for which you are retrieving a profile.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getospfprof –u qipman –p passwd –n new_ospf –f ospflst.txt
```

Output example

```
OspfAreaName=rew_ospf
OspfAreaID=0.1.226.64
WarningPercent=0
WarningType=0
SubnetAddress=180.200.0.1
SubnetName=subnet_name
NetworkAddress=180.200.0.0
NetworkName=network_180_200_new_name
SubnetOrgName=rew_subnet_org
OspfName=rew_ospf
SubnetAddress=180.200.0.1
SubnetName=subnet_name
```

```
NetworkAddress=180.200.0.0
NetworkName=network_180_200_new_name
SubnetOrgName=rew_subnet_org
```

qip-getpolicy

qip-getpolicy shows policy information on a policy for the enterprise server. For more information on policies, refer to Chapter 2, "Establish Policies and Profiles", in the *VitalQIP User's Guide*.

Synopsis

```
qip-getpolicy [-c class] [-n policyName] [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-f filename]
```

Parameters

qip-getpolicy recognizes the following parameters:

-c ***class***	Specifies the class of policy you want to show information on (for example, BILLING). If you do not specify a class, all classes appear.
-n ***policyName***	Specifies the policy you want to show information on (for example, PING_ATTEMPTS and ALWAYS_APPEND_ROUTER). If you do not specify a policy, you see all policies.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input examples

- To show policy information for all classes:

```
qip-getpolicy -s QIPSYBASE -U qipman -P passwd
```

- To show policy information for all policies with the Billingclass:

```
qip-getpolicy -s QIPSYBASE -u qipman -p passwd -c Billing
```

- To show specific policy information for the user class policy in the Billing class:

```
qip-getpolicy -s QIPSYBASE -u qipman -p passwd -c Billing -n Required_Data
```

Output example

```
Class=Billing
Policy=Required_Data, Value=USER GROUP
```

qip-getprimdnssvrlst

qip-getprimdnssvrlst makes a list of the primary DNS servers specified in the VitalQIP database based on domain name or the IP address of the reverse zone. The output is stored in the specified (or default) file.

Synopsis

```
qip-getprimdnssvrlst [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization] [-f filename]
   -n domain_name|-a in-addr_IPaddress
```

Parameters

qip-getprimdnssvrlst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.
-n *domain_name*| -a *in-addr_Ipaddress*	Specifies the fully-qualified domain name of the primary DNS servers or the IP address of the reverse zone for which the servers are the primary.

Command line input example 1

- To get the list of primary DNS servers for the domain *example.com*:

```
qip-getprimdnssvrlst -n example.com
```

Output example 1

```
primdnsserver.example.com
```

Command line input example 2

- To get the list of primary DNS servers for the reverse zone 10.0.0.0:

```
qip-getprimdnssvrlst -a 10.0.0.0
```

Output example 2

```
primdnsserver.example.com
```

qip-getrevzonelst

qip-getrevzonelst retrieves a list of reverse zones for an organization from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

qip-getrevzonelst [**-g** ***loginserver***] [**-s** ***servername***] [**-o** ***organization***] [**-u** ***username***] [**-p** ***password***] [**-f** ***filename***]

Parameters

qip-getrevzonelst recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getrevzonelst -u qipman -p passwd -f revzonelst.txt
```

Output example

```
10.in-addr.arpa,10.0.0.0/8
0.0.10.in-addr.arpa,10.0.0.0/24
1.0.10.in-addr.arpa,10.0.1.0/24
2.0.10.in-addr.arpa,10.0.2.0/24
3.0.10.in-addr.arpa,10.0.3.0/24
4.0.10.in-addr.arpa,10.0.4.0/24
5.0.10.in-addr.arpa,10.0.5.0/24
0.0.11.in-addr.arpa,11.0.0.0/24
123.in-addr.arpa,123.0.0.0/8
1.1.192.in-addr.arpa,192.1.1.0/24
1.1.194.in-addr.arpa,194.1.1.0/24
138.200.198.in-addr.arpa,198.200.138.0/24
0-27.138.200.198.in-addr.arpa,198.200.138.0/27
64-27.138.200.198.in-addr.arpa,198.200.138.64/27
```

qip-getrtrlst

qip-getrtrlst retrieves a list of all routers in the specific subnet from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

```
qip-getrtrlst -a subnet_address [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization] [-f filename]
```

Parameters

qip-getrtrlst recognizes the following parameters:

-a *subnet_address*	Specifies the subnet address from which you want to retrieve routers.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getrtrlst -u qipman -p passwd -a 199.200.139.0 -f rtrlst.txt
```

Output example

```
199.200.139.1
199.200.139.161
```

qip-getsecdnssvrlst

qip-getsecdnssvrlst returns a list of fully-qualified secondary DNS server names based on the domain name or the IP address of the reverse zone. The output is stored in the specified (or default) file.

Synopsis

```
qip-getsecdnssvrlst [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-f filename]
   -n domain_name|-a in-addr_IPaddress
```

Parameters

qip-getsecdnssvrlst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.
-n *domain_name* | -a *in-addr_IPaddress*	Specifies the fully-qualified domain name of the primary DNS servers or requests the IP address of the reverse zone for which the DNS servers are secondary.

Command line input example

```
qip-getsecdnssvrlst -u qipman -p passwd -a 200.200.200.0 -f sdnssvrlst.txt
```

Output example

```
dns2.qtek.com
dns3.qtek.com
```

qip-getsnaddr

qip-getsnaddr gets the subnet address, given the IP address of an object. The output is stored in the specified (or default) file.

Synopsis

```
qip-getsnaddr -a object_address [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization] [-f filename]
```

Parameters

qip-getsnaddr recognizes the following parameters:

-a *object_address*	Specifies the IP address of an object.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getsnaddr -u qipman -p passwd -a 199.200.138.27 -f snaddr.txt
```

Output example

```
SubnetAddress=199.200.138.0
```

qip-getsnorglst

qip-getsnorglst retrieves a list of all subnet organizations from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

```
qip-getsnorglst [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-f filename]
```

Parameters

qip-getsnorglst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getsnorglst –u qipman –p passwd –f subgrplst.txt
```

Output example

```
"qtek_suborg1"
"rew_subnet_org"
```

qip-getsnorgprof

qip-getsnorgprof returns the profile of a subnet organization. The output is stored in the specified (or default) file.

Synopsis

```
qip-getsnorgprof [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] -n subnet_organization_name
   [-f filename]
```

Parameters

qip-getsnorgprof recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-n ***subnet_organization_name***	Specifies the name of the subnet organization.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getsnorgprof -u qipman -p passwd -n psl_suborg1 -f out.dat
```

Output example

For a description of the fields in the output example, refer to Table 1-14.

```
SubnetOrgName=snorg1
WarningPercent=95
WarningType=1
AssignGap=Workstation,1
AssignGap=PC,2
SubnetAddress(net,mask,addr)=144.144.0.0,255.255.255.0,144.144.1.
   0
SubnetAddress(net,mask,addr)=144.144.0.0,255.255.255.0,144.144.2.
   0
```

```
W2KSiteName=site1
UseSubnetOrgName=false
AssocControllers=dc1.lucent.com
DhcpServer=dhcp1.lucent.com
DhcpOptionTemplate=general
```

Table 1-14 Subnet organization profile field definitions

Field name/label	Description
SubnetOrgName	The Subnet Organization name.
WarningPercent	The Warning Managed Addresses full %. (for example, 10 equals 10 percent)
WarningType	Warning Type: 0 – no warning 1 – email only 2 – audible only 3 – both email and audible warning
AssignGap	The object class of the GAP object defined. The relative offset. Note that a positive number (>0) indicates “from the beginning” of the subnet. A negative number indicates “from the end” of the subnet.
SubnetAddress(net,mask, addr)	Specifies the subnet address, network address, and subnet mask.
W2KSiteName	The site name of the Windows 2000 site. If blank, the subnet organization name is used. (Optional)
AssocControllers	The list of Windows 2000 controllers.
DHCPServer	The default DHCP server for this subnet.
DHCPOptionTemplate	The default DHCP Option Template for this subnet.
UseSubnetOrgName	Specifies if the subnet organization name is used.

qip-getsubnetlst

qip-getsubnetlst retrieves a list of all subnets the specified domain ([name]) manages from the VitalQIP database. If name is provided, the list includes all subnets under that domain only. Otherwise, it includes all subnets in VitalQIP. The output is stored in the specified (or default) file.

Synopsis

```
qip-getsubnetlst [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] -t owner_type
   -n name|-a address [-f filename]
```

Parameters

qip-getsubnetlst recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-t ***owner_type***	Specifies the owner type under which subnets belong. Refer to Table 1-15 following.
-n ***name*** |**-a** ***address***	Specifies the IP Address or fully-qualified domain, subnet organization, OSPF, or network name from where you want to retrieve subnets or the network address from where you want to retrieve subnets.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Table 1-15 Owner types

Owner type	Notes
Corp	Displays subnets for an Organization. Use **–n** to specify the organization name.
Subnetorg	Displays subnets for a Subnet Organization. Use **–n** to specify the subnet organization name.

Owner type	Notes
Domain	Displays subnets for a domain. Use **–n** to specify the domain name.
OSPF	Displays subnets for an OSPF Area. Use **–n** to specify the OSPF area name.
Network	Displays subnets for a network. Use **–a** to specify the network address.

Command line input example

```
qip-getsubnetlst -u qipman -p passwd -t corp -f subnetlst.txt
```

Output example

For a description of the fields in the output example, refer to Table 1-16.

```
"sn1" "10.200.60.0" "N" "Network A" "10.200.60.0" "Building 1" "os1"
   "255.255.255.128"
```

Table 1-16 Subnet list field definitions

Field name/label	Description
Subnet Name	The Subnet Name of the subnet.
Subnet Address	The Subnet Address.
Usage Flag	Usage Flag: Y–Used N–Not Used S–Scheduled Move P–Planned Move
Network Name	Network name.
Network Address	Network Address.
Subnet Organization	Subnet Organization to which this subnet belongs.
OSPF Area name	OSPF Area to which this subnet belongs.
Subnet Mask	Subnet Mask

qip-getsubnetprof

qip-getsubnetprof retrieves the Subnet Profile of a specific subnet from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

```
qip-getsubnetprof -a subnet_address [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-f filename]
```

Parameters

qip-getsubnetprof recognizes the following parameters:

-a *subnet_address*	Specifies the subnet address from where you want to retrieve a Subnet Profile.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getsubnetprof -u qipman -p passwd -a 199.200.139.0 -f subnet_prof.txt
```

Output example

For a description of the fields in the output example, refer to Table 1-17.

```
SubnetAddress=144.144.144.0
SubnetName=SN144_144
SubnetMask=255.255.255.0
NetworkAddress=144.144.0.0
LocationID=2
Street1=400
Street2=Lapp Road
City=Malvern
State=PA
Zip=19355
```

```
Country=usa
ContactID=2
ContactLastName=pal
ContactFirstName=pal
ContactEmail=pal@qtek.com
ContactPhone=111-0033
ContactPager=1-800-111-3300
Application=
Domain=qtek.com
TftpServer=
ShowUsage=N
CheckUsage=N
SubnetDescription=
SharedNetwork=
HardwareType=
WarningType=2
WarningPercentage=60
DNSServers=dns1.qtek.com
TimeServers=pal.qtek.com
DefaultRouters=144.144.144.64 144.144.144.100 144.144.144.101
DHCPServer=pal.qtek.com
DHCPOptionTemplate=
DHCPPolicyTemplate=
PrimaryInterface=No
AllowDHCPClientsModifyDynamicObjectResourceRecords=True
```

Table 1-17 Subnet profile field definitions

Field	Description
SubnetAddress	The subnet Address.
SubnetName	The subnet name.
SubnetMask	The subnet mask.
NetworkAddress	The network address.
LocationID	The Location ID of the Subnet.
Street1	Street1 field of the location.
Street2	Street2 field of the location.
City	City of the location.
State	State of the location.
Zip	Zip of the location.
Country	Country of the location.

Field	Description
ContactID	The Contact ID of the subnet.
ContactLastName	Contact last name.
ContactFirstName	Contact first name.
ContactEmail	Contact email address.
ContactPhone	Contact phone number.
ContactPager	Contact pager number.
Application	Primary Application assigned to this subnet.
Domain	Domain assigned separated by spaces.
TftpServer	Default Tftp server assigned.
ShowUsage	Show Used Only Flag: 0 = No 1 = Yes
CheckUsage	Check before assign: Y=Ping N=None
SubnetDescription	Subnet “comment” text area.
SharedNetwork	List of the shared networks for this subnet.
HardwareType	Default hardware type: Ethernet TokenRing AX.25 Pronet Chaos -IEEE802 -Arcnet
WarningType	The Warning Type is issued with one of the numeric values: 0 = No warning is set 1 = Warning is set to E-mail 2 = Warning is set to Visual 3 = Warning is set to E-mail and Visual
WarningPercentage	Indicates the percentage of objects in service for a subnet at which a warning is issued.
DNSServers	DNS Servers for this subnet, separated by spaces.

Field	Description
TimeServers	Time Servers for this subnet, separated by spaces.
DefaultRouters	Default Routers for this subnet, separated by spaces.
DHCPServer	The default DHCP server for this subnet.
DHCPOptionTemplate	The default DHCP Option Template for this subnet.
DHCPPolicyTemplate	The DHCP Policy Template for this subnet.
PrimaryInterface	Primary Interface – Yes / No
AllowDHCPClients ModifyDynamicObjectResourceRecords	True False Same as in Global Policies

qip-gettemplate

qip-gettemplate retrieves a specified DHCP template(s) and writes it to the specified (or default) output file.

Synopsis

```
qip-gettemplate -n template_name [-g loginserver]
   [-s servername][-u username] [-p password] [-o organization]
   [-f filename] -t option | policy_subnet | policy_clientclass |
   policy_scope | policy_server
```

Parameters

qip-gettemplate recognizes the following parameters:

-n ***template_name***	Specifies the name of the DHCP template you want to retrieve.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.
-t option | policy_subnet | policy_clientclass | policy_scope | policy_server	This parameter is used in conjunction with the **-n** parameter to specify the type of template for which to search. Valid options or values are: • **option** - searches for a specific option template. • **policy_subnet** - searches for a specific subnet policy template. • **policy_clientclass** - searches for a specific client class policy template. • **policy_scope** - searches for a specific scope policy template. • **policy_server** - searches for a specific server policy template.

Command line input example

qip-gettemplate -n MS_DHCP_Server_Template -s fred -u wilman -p wilma -f tmpl.dat

Output example

For information on option descriptions, refer to the tables in “DHCP/Bootp templates” section in Chapter 2 of the *VitalQIP User’s Guide*.

```
template-name=MS_DHCP_Server_Template
template-type=DHCP Option Template
002=64157
003=198.200.138.234
013=31588
015=quadritek.com
019=33
043=54321
template-name=10.200.80.0_Template
template-type=SUBNET
003=198.200.138.0
051=120
```

qip-gettemplst

qip-gettemplst retrieves a list of all DHCP templates from the VitalQIP database. The output is stored in the file name specified.

Synopsis

```
qip-gettemplst [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-f filename] [-t option |
   policy_subnet | policy_clientclass | policy_scope |
   policy_server]
```

Parameters

qip-gettemplst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.
-t option | policy_subnet | policy_clientclass | policy_scope | policy_server	This parameter is used to specify the type of template for which to search. Valid options or values are: • **option** - searches for option templates • **policy_subnet** - searches for subnet policy templates • **policy_clientclass** - searches for client class policy templates • **policy_scope** - searches for scope policy templates • **policy_server** - searches for a server policy templates

Command line input example

```
qip-gettemplst -u qipman -p passwd -f templates.txt
```

Output example

```
"general"
"microsoft_clients"
```

qip-gettimesvrlst

qip-gettimesvrlst retrieves a list of all time servers from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

```
qip-gettimesvrlst [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-f filename]
```

Parameters

qip-gettimesvrlst recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-gettimesvrlst -u qipman -p passwd -f timesvrs.txt
```

Output example

```
TimeSvr.qtek.com
TimeSvr2.qipman.com
```

qip-getudflst

qip-getudflst retrieves a list of all user-defined field names or values from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

```
qip-getudflst -t fields | values -c class_name
   [-a address|-n name]
   [-g loginserver] [-s servername] [-u username] [-p password]
   [-o organization] [-f filename]
```

Parameters

qip-getudflst recognizes the following parameters:

-t fields|values	Specifies either “fields” or “values” to retrieve a list of user-defined field names or user-defined field values.
-c *class_name*	Specifies the user-defined fields class name (type) associated with the user-defined field name. Refer to Table 1-18 for class name types and definitions.
-a *address* | -n *name*	Specifies the IP address of the class object, subnet, reverse zone, or the fully qualified object name with which the value is associated. The fully qualified object name is applicable when referring to the owner of the domain, organization, user, or object. When referring to the user, the login ID assigned to the user is applicable. You can only use the **-a** parameter if the **-t** parameter is set to **values**. You can only use the **-n** parameter if the **-t** parameter is set to **fields**.
-g *loginserver*	Specifies the VitalQIP login server’s IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Table 1-18 UDF class name types

UDF_Class_ Name	Description
User	Used for user-defined fields for a user.
Domain	Used for user-defined fields for a domain.
ReverseZone	Used for user-defined fields for a reverse zone.
Organization	Used for user-defined fields for an organization.
Object	Used for user-defined fields for an object profile.
Subnet	Used for user-defined fields for a subnet profile.

Command line input example 1

- To retrieve ALL user-defined fields of the type “User”:

```
qip-getudflst -u qipman -p passwd -t field -c User
```

Output example 1

```
"USER1"
```

Command line input example 2

- To retrieve ALL user-defined field values of the type “User”, for the user “jsmith”:

```
qip-getudflst -t value -c User -n jsmith
```

Output example 2

```
"USER1"  "jsmith"
```

Command line input example 3

- To retrieve ALL user-defined fields of the type “Organization”:

```
qip-getudflst -t field -c Organization
```

Output example 3

```
"ORG1"
"ORG2"
"ORG3"
"ORG4"
```

Command line input example 4

- To retrieve ALL user-defined field values of the type “Organization” with the name “VitalQIP Organization”:

```
qip-getudflst -t value -c Organization -n "VitalQIP Organization"
```

Output example 4

```
"ORG1"  "QA"
"ORG2"  "DEV"
"ORG3"  "MKTG"
"ORG4"  "QIP Org4"
```

Command line input example 5

- To retrieve ALL user-defined fields of the type "ReverseZone" for "Sales":

qip-getudflst –t field –c ReverseZone –o "Sales" –n reversezonels1

Output example 5

```
"REVERSEZONE1"
```

Command line input example 6

- To retrieve user-defined field values of the type "Reverse Zone" for a specific subnet, use the **-a** parameter and a network address/mask value:

qip-getudflst –t value –c ReverseZone –o "Mktg Organization" –a 10.0.0.0/8

Output example 6

```
"REVERSEZONE1"  "SALESREVERSEZONE"
```

Command line input example 7

- To retrieve user-defined field names of the type "Domain":

qip-getudflst –t field –c Domain

Output example 7

```
D1
D2
D3
```

Command line input example 8

- To retrieve user-defined field values for a *specific* domain use the **-n** option:

qip-getudflst –t field –c Domain –n seg1.qa.quadritek.com

Output example 8

```
"D1" "Seg1"
"D2" "qa"
"D3" "quadritek"
"D4" "com"
```

qip-getudfnamelst

qip-getudfnamelst retrieves a list of all user-defined field names from the VitalQIP database. The output is stored in the specified (or default) file.

Synopsis

qip-getudfnamelst [**-g** ***loginserver***] [**-s** ***servername***] [**-u** ***username***] [**-p** ***password***] [**-o** ***organization***] [**-f** ***filename***]

Parameters

qip-getudfnamelst recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getudfnamelst -u qipman -p passwd -f nameLst
```

Output example

```
PSL-UDF2,Object
SN-UDF,Subnet
DOMN-UDF,Domain
ORT-UDF,Organization
USER1,User
REVZ-UDF,ReverseZone
```

qip-getuser

qip-getuser retrieves VitalQIP User information. The output is stored in the specified (or default) file.

Synopsis

```
qip-getuser [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] -l login_name [-f filename]
```

Parameters

qip-getuser recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-l *login_name*	Specifies the login ID of the VitalQIP user whose information is being retrieved.
-f *filename*	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getuser –u qipman –p password –l pal3
```

Output example

Note: For a description of the fields in this output example and more information on users, refer to "Define VitalQIP Users" section in Chapter 6 of the *VitalQIP User's Guide*.

```
LoginID=pal3
LastName=test
FirstName=pal3
Phone=292-1312
EmailAddress=pal3@qtek.com
Password=******
Pin=
Description=another user with different id
Street1=5
Street2=Whitney Ave
```

```
City=New Haven
State=Ct
Zip=02488
Country=usa
ActivationStatus=1
GroupInformation=”isp_qtek2”,”isp_qtek”
DefaultSubnets=144.144.144.0
ManagedRange=144.144.144.3,144.144.144.2
UserDefinedFields=”USER1: user1 udf value for pal3”
```

qip-getuseraddrlst

qip-getuseraddrlst retrieves a list of IP objects associated with the user. It retrieves the IP address, MAC Address, hostname, dynamic type, lease grant, and expiration values. The output is stored in the specified (or default) file.

Synopsis

qip-getuseraddrlst [**-g** ***loginserver***] [**-s** ***servername***] [**-u** ***username***] [**-p** ***password***] [**-o** ***organization***] **-l** ***login_name*** [**-f** ***filename***]

Parameters

qip-getuseraddrlst recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-l ***login_name***	Specifies the login name of the user to retrieve the address list for.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example

```
qip-getuseraddrlst -u qipman -p password -l testuser -f
   usr_list.dat
```

Output example

The format of the file is as follows:

IP_address, hostname, MAC_address, dynamic_configuration, lease_granted, lease_expires

```
198.200.139.11, testuserpc, 223344556677, Dynamic_DHCP,,
198.200.138.12, Device1, 112233445566, Manual_DHCP, 10/15/1999
   09:00,
```

qip-getzoneext

qip-getzoneext retrieves the contents of the zone extension associated with the domain (**-n** parameter), or DNS server extensions provided in the **-d** parameter, or the reverse zone provided in the **-a** parameter. (The **-d**, **-n,** and **-a** parameters are mutually exclusive – only one can be used at a time.)

Synopsis

qip-getzoneext -n *domain_name* |-a *reverse_zone_address* / *maskLen* / -d *dns_server_fqdn* [**-g *loginserver***] [**-s *dataserver***] [**-u *username***]
[**-p *password***] [**-o *organization***] [**-f *output_filename***] [**-i *input_file***]

Parameters

qip-getzoneext recognizes the following parameters:

-n *domain_name* / -a reverse_*zone_address*/ *maskLen* / -d *dns_server_fqdn*	Specifies the fully qualified domain name (**-n**) for which you are retrieving a profile, or the IP address (**-a**) of the reverse zone and mask for which you are retrieving a profile, or the DNS Server name (**-d**) for which you want to retrieve extensions.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *filename*	Specifies the filename of the output file.
-i *input_file*	Specifies the directory and filename where the input data is to be placed.

Command line input examples

- To get the Domain Extensions, execute the following:

qip-getzoneext –u qipman –p passwd –n qtek.com –f extfile.txt

- To get the DNS Server Extensions, execute the following:

```
qip-getzoneext -u qipman -p passwd -d dnsrver1.qtek.com -f
   extfile.txt
```

- To get the Reverse Zone Extensions, execute the following:

```
qip-getzoneext -u qipman -p passwd -a 144.44.44.0/24 -f
   extfile.txt
```

Output examples

For a description of the fields in this output example and information on Domain Extensions, refer to "Define Domains" in Chapter 3 of the *VitalQIP User's Guide*.

- Output for Domain Extensions:

```
DomainName=qtek.com
OptionType=Extensions
ParameterName=Prefix of zone db file
STTL 6400
ParameterName=Postfix of zone db file
candybar IN A 19.20.21.22
snickers IN CNAME candybar.quadritek.com
candybar IN HINFO "VAX" "UNIX"
quadritek.com. IN MX 10 candybar.quadritek.com.
OptionType=BIND-8.X Options
ParameterName=zone block of named.conf
// extra zone options for BIND 8.X server
qddns {
wins-servers {1,2,3,4:1.2.3.5;}
wins-ttl 0;
};
OptionType=BIND-9.X Options
ParameterName=zone block of named.conf
// extra zone options for BIND 9.X server
forward only;
zone-statistics yes;
OptionType=LUCENT DNS 3.X Options
ParameterName=zone block of named.conf
// extra zone options for LUCENT DNS 3.X server
// …
OptionType=LUCENT DNS 4.X Options
ParameterName=zone block of named.conf
// extra zone options for LUCENT DNS 4.X Server
// …
OptionType=WINDOWS 2000 DNS Options
ParameterName=zone-options
// extra zone options for WINDOWS 2000 DNS Server
// …
```

- Output for DNS Server Extensions:

```
ServerName=dnsserver.qtek.com
ParameterName= Corporate Extension
include "allowquery.conf"
include "blackhoe.conf"
ParameterName=db.cache file extension
Include "test.conf"
```

- Output for Reverse Zone Prefix and Postfix Extensions:

```
ReverseZone=170.200.0.0/16
DomainName=qtek.com
OptionType=Extensions
ParameterName=Prefix of zone db file
STTL 4600
ParameterName=Postfix of zone db file
68 IN PTR madeup.quadritek.com.
69 IN PTR madeup2.quadritek.com.
70 IN PTR madeup3.quadritek.com.
71 IN PTR madeup4.quadritek.com.
OptionType=BIND-8.X Options
ParameterName=zone block of named.conf
qddns {
wins-servers {1,2,3,4:1.2.3.5;}
wins-ttl 0;
};
OptionType=BIND-9.X Options
ParameterName=zone block of named.conf
// extra zone options for BIND 9.X server
forward only;
zone-statistics yes;
OptionType=LUCENT DNS 3.X Options
ParameterName=zone block of named.conf
// extra zone options for LUCENT DNS 3.X server
// …
OptionType=LUCENT DNS 4.X Options
ParameterName=zone block of named.conf
// extra zone options for LUCENT DNS 4.X Server
// …
OptionType=WINDOWS 2000 DNS Options
ParameterName=zone-options
// extra zone options for WINDOWS 2000 DNS Server
// …
```

qip-getzoneprof

qip-getzoneprof retrieves the profile of a specified domain or reverse zone. The output is stored in the specified (or default) file.

Synopsis

```
qip-getzoneprof -n zone_name | -a zone_address/maskLen
   [-g loginserver] [-s servername] [-u username] [-p password]
   [-o organization] [-f filename]
```

Parameters

qip-getzoneprof recognizes the following parameters:

-n ***zone_name*** **|** **-a** ***zone_address/maskLen***	Specifies the fully-qualified domain name (**-n**) for which you are retrieving a profile, or the IP address (**-a**)of the reverse zone and mask for which you are retrieving a profile.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***filename***	Specifies the file where the information is stored. The default is *STDOUT*.

Command line input example 1

```
qip-getzoneprof -n rob.com -u qipman -p passwd -f domainprof.txt
```

Output example 1

For a description of the fields in the Output examples, refer to the section "Define Domains" in Chapter 5 of the *VitalQIP User's Guide*.

```
Zone=rob.com
dnsServers=dns1.rob.com P 0,dns2.rob1.com S 0,dns2k.rob2k.com S 0
RefreshTime=21600
ExpirationTime=604800
RetryPeriod=3600
MinimumTTL=86400
```

```
NegativeCacheTTL=86400
ZoneMail=rob@rob.com
Extensions
Prefix of zone db file= My zone prefix comments
Postfix of zone db file= My zone postfix comments
BIND-8.X Options
allow-query=Any
allow-transfer=Any
allow-update=Any
check-names=Warn
notify=No
zone block of named.conf=
BIND-9.X Options
allow-notify=Use Server Value
allow-query=Any
allow-transfer=Any
allow-update=Any
notify=No
zone block of named.conf=
LUCENT DNS 3.X Options
Import External Updates=False
allow-query=Any
allow-transfer=Any
allow-update=Any
check-names=Warn
notify=No
zone block of named.conf=
LUCENT DNS 4.X Options
Import External Updates=True
A (Host IPV4)=False
AAAA (Host IPV6)=True
CNAME (Canonical Name)=False
PTR (Pointer)=True
SRV (Server Resource Record)=False
TXT (Text)=True
allow-notify=Use List
ACL Templates=WestACLTemplate;SouthACLTemplate
other=ThisIsOther;MoreOther99;10.200.60.25
allow-query=Any
allow-transfer=Use Server Value
allow-update=localnets
notify=No
zone block of named.conf= my zone block of named.conf comments
WINDOWS 2000 DNS Options
aging=False
```

```
allow-transfer=Any
allow-update=No
no-refresh-interval=0
notify=No
refresh-interval=0
zone-options=
```

Example 2 (reversezone)

```
Zone=200.200.200.0/24
ParentAddress=
NetworkAddress=
dnsServers=revdns.rev.com P 0
RefreshTime=21600
ExpirationTime=604800
RetryPeriod=3600
MinimumTTL=86400
NegativeCacheTTL=86400
ZoneMail=rev1@rev.com
Extensions
Prefix of zone db file=My zone prefix comments
Postfix of zone db file=My zone postfix comments
BIND-8.X Options
allow-query=Any
allow-transfer=Any
allow-update=Any
check-names=Warn
notify=No
zone block of named.conf=
BIND-9.X Options
allow-notify=Use Server Value
allow-query=Any
allow-transfer=Any
allow-update=Any
notify=No
zone block of named.conf=
LUCENT DNS 3.X Options
Import External Updates=False
allow-query=Any
allow-transfer=Any
allow-update=Any
check-names=Warn
notify=No
zone block of named.conf=
LUCENT DNS 4.X Options
Import External Updates=True
```

```
A (Host IPV4)=False
AAAA (Host IPV6)=True
CNAME (Canonical Name)=False
PTR (Pointer)=True
SRV (Server Resource Record)=False
TXT (Text)=True
allow-notify=Use List
ACL Templates=WestACLTemplate;SouthACLTemplate
other=ThisIsOther;MoreOther99;10.200.60.25
allow-query=Any
allow-transfer=Use Server Value
allow-update=localnets
notify=No
zone block of named.conf=my zone block of named.conf comments
WINDOWS 2000 DNS Options
aging=False
allow-transfer=Any
allow-update=No
no-refresh-interval=0
notify=No
refresh-interval=0
zone-options=
```

qip-globalmacpool

qip-globalmacpool allows you to add or delete an individual MAC address from a DHCP server MAC pool or query the MAC pool available to a DHCP server.

Synopsis

```
qip-globalmacpool -n dhcp_server_name -a add | delete | query
   [-m mac_address] [-t hardware_type] [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-x mac_address]
```

Parameters

qip-globalmacpool recognizes the following parameters:

-n ***dhcp_server_name***	Specifies the fully-qualified server name of the MAC pool owner that you want to add or delete a MAC pool address.
-a add|delete|query	Specifies the process: to add or delete a MAC address, or to query the available MAC pool. The optional argument (**-x**) is available for **-a add** only, and indicates that specific addresses should be excluded. Using the **-a** displays the MAC address, hardware type, and excluded attribute specified with the **-x** (if applicable).
-m ***mac_address***	Specifies the MAC address to be added or deleted. Supports the trailing wildcard character (*).
-t ***hardware_type***	Specifies the hardware type of the object with the MAC address to be added. The following options are available: – Hardware_Type – Ethernet – TokenRing – AX.25 – Pronet – Chaos – IEEE802 – Arcnet
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.

-o *organization*	Specifies the VitalQIP organization (corporation) name.
-x *mac_address*	Indicates that the specified address should be excluded **Note:** This parameter is only applicable when **-a** is used.

Command line input examples

- To add a MAC address to the "dhcp1.qtek.com" server:

```
qip-globalmacpool -u qipman -p passwd -n dhcp1.qtek.com -a add -m
   00008e334455 -h ethernet
```

- To enter a query of the "dhcp1.qtek.com" server for global MAC pool addresses:

```
qip-globalmacpool -u qipman -p passwd -n dhcp1.qtek.com -a query
```

- To delete a MAC address:

```
qip-globalmacpool -a delete -m 00008e334455
```

- To add a MAC address, excluding specific addresses:

```
qip-globalmacpool -a add  -x -m 01028b* -t ethernet -n oneill-
   w2k.quadritek.com
```

Output example

```
MacAddr=111111116771, HardwareType=AX.25
MacAddr=121212121221, HardwareType=Ethernet
MacAddr=121212223333, HardwareType=Ethernet
MacAddr=333213321111, HardwareType=Pronet
MacAddr=333213775511, HardwareType=Chaos
MacAddr=333213776677, HardwareType=Arcnet
MacAddr=551212112212, HardwareType=ieee802
MacAddr=551212121111, HardwareType=Token Ring
MacAddr=553423121212, HardwareType=Ethernet
```

qip-hndbgen

qip-hndbgen is equivalent to performing the **Network Services|NIS Generation** option in the VitalQIP interface which generates NIS records into the ethers file, hosts file, and netmasks file. The output is stored in the NIS servers default directory (specified in the Server Profile) or the directory specified with the **-d** option.

Synopsis

```
qip-hndbgen -n NIS_ServerName [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization] [-d directory]
```

Parameters

qip-hndbgen recognizes the following parameters:

-n *NIS_ServerName*	Specifies the fully-qualified NIS server name.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-d *directory*	Specifies the directory name where the information is returned. If no directory is specified, the files are written to the NIS server's default directory specified in the Server Profile.

Command line input example

```
qip-hndbgen -u qipman -p passwd -n nis.qtek.com -d /opt/qip
```

qip-hostgen

qip-hostgen generates a Local Host configuration file to the Local Host Server defined. The output is stored in the specified (or default) directory.

Synopsis

```
qip-hostgen -n host_name [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization] [-d directory]
```

Parameters

qip-hostgen recognizes the following parameters:

-n ***host_name***	Specifies the fully-qualified name of the local host server.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-d ***directory***	Specifies the directory name where the information is returned. The default is the non-DNS server directory or */tmp* if not defined.

Command line input example

```
qip-hostgen –n local.qtek.com –u qipman –p passwd –d /tmp
```

Output example

```
161.251.83.58 blwp029 blwp029.qtek.com
161.251.83.59 blwp030 blwp030.qtek.com
161.251.83.60 blwp031 blwp031.qtek.com
```

qip-leasefilegen

qip-leasefilegen creates a VitalQIP DHCP Active Lease file from information stored in the VitalQIP database. The generated Active Lease file can be used to re-create active lease files in case of DHCP server failure or a corrupted active lease database. After this file is generated, you can place this file on your VitalQIP DHCP server and restart the VitalQIP DHCP server so the system reads the new configuration.

Synopsis

qip-leasefilegen [**-g** ***loginserver***] [**-s** ***servername***] [**-u** ***username***] [**-p** ***password***] [**-o** ***organization***] [**-n** ***dhcpServerName***] [**-d** ***directory***]

Parameters

qip-leasefilegen recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-n ***dhcpServerName***	Specifies the fully-qualified DHCP server name.
-d ***directory***	Specifies the directory where the *dhcp.db* file should reside.

Command line input example

```
qip-leasefilegen -u qipman -p passwd -n dhspQ.qtek.com -d /qiphome/dhcp
```

Output example for DHCP 5.5 server

```
144.144.144.013 00-60-97-40-ca-62 MDLAPTOP example.com 2 0858360960
0958362760 5206010400000123 MSFT^5.0
144.144.144.014 00-80-5f-01-84-f4 ws_support_3 example.com 2 0857758374
0857760174 ___R___ ___V___
```

Output example for DHCP 5.6 server

```
144.144.144.013 00-60-97-40-ca-62 MDLAPTOP example.com 2 0858360960
0958362760 5206010400000123 MSFT^5.0 __C__
144.144.144.014 00-80-5f-01-84-f4 ws_support_3 example.com 2 0857758374
0857760174 ___R___ ___V___ ___C___
```

where ____R___ indicates that the lease had no option 82 information, ___V___ indicates that the lease has no vendor class information, and ___C___ represents the client ID.

qip-mcancel

qip-mcancel cancels a previously scheduled move.

Synopsis

```
qip-mcancel -t object|subnet -n name|-a address [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-b begin_date] [-e end_date]
```

Parameters

qip-mcancel recognizes the following parameters:

-t object|subnet	Specifies the type of move you want to cancel, object or subnet.
-n *name*|-a *address*	Specifies the object or subnet name or the object or subnet IP address.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-b *begin_date*	Begins the cancellation from this date. Use the format mm/dd/yyyy HH:MM (time is optional). If time is used, quotes must be placed around the date and time. (**-t subnet** only)
-e *end_date*	Performs cancellation until this date. Use the format mm/dd/yyyy HH:MM (time is optional). If time is used, quotes must be placed around the date and time. (**-t subnet** only)

Command line input example

```
qip-mcancel –u qipman –p passwd –t object –n Workstation-0417
qip-mcancel –u qipman –p passwd –t object –a 198.200.138.211
```

Note: If you specify an "object" (using the **-t** option), the ***begin_date*** and ***end_date*** fields are ignored.

qip-move

qip-move either moves objects from one subnet to another on demand or pre-schedules the move for a future date. Individual objects, or entire subnets can be moved.

VitalQIP validates data to prevent CNAME conflicts in the same organization by default. If a CNAME conflict occurs during an object move, an error message displays and the object move fails. You can disable this feature by setting the Validate CNAME Records policy to False. See the Object Policies and Profiles section of the *VitalQIP User Guide* for more information on the policy. When you move an object, validation checks on the object name are made against the following:

- Object Names
- Domain Names
- Aliases
- Object Resource Records
- Domain Resource Records
- Reverse Zone Resource Records

Additionally, any resource records defined on the moved objects are validated.

Synopsis

```
qip-move -t Object|Subnet -a from_address|-n from_Name
   -l subnet|object
   -e to_address [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-d mm/dd/yyyy [-m hh:mm]]
```

Parameters

qip-move recognizes the following parameters:

-t Object|Subnet	Specifies whether you want to move the object or subnet.
-a *from_address*| -n *from _Name*	Specifies the address of the object or subnet from which you want to move from, or the name of the object or subnet from which you want to move.
-l *object|subnet*	Specifies whether you want this to be moved to an object or subnet.
-e *to_address*	Specifies the address where you want to move the object or subnet to.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.

-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-d *mm/dd/yyyy* **-m *hh:mm***	If the date is specified, move the objects on the specified date; otherwise move them immediately. Use the format mm/dd/yyyy hh:mm (time is optional).

Command line input examples

- To move an object from one subnet to another:

```
qip-move –a 101.114.21.30 –e 101.114.21.30 -d 12/09/2003
```

- To move an object to a subnet:

```
qip-move –a 101.114.21.30 –t object –l subnet –e 101.114.21.30 -d 12/09/2003
```

- To move an object to another object:

```
qip-move –u qipman –p passwd –a 199.200.13.27 –t object–e 199.200.139.64 –d
   12/31/2003
```

qip-msextract

qip-msextract extracts the configuration of a Microsoft DHCP Server running on Windows, so it can be imported into VitalQIP. This configuration is read from the Windows registry and includes scope (subnet) ranges, server- and scope-level options, lease times for each scope, excluded (static) addresses for each scope, and reserved (manual DHCP) addresses for each scope. **qip-msextract** creates data files for three VitalQIP import CLI commands: **qip-template**, **qip-scope**, and **qip-setobject**. It also creates a batch file (called *mstoqip.bat*) that can be run from a DOS window that runs the three CLI commands.

Before you begin

There are several limitations with **qip-msextract**:

- The Microsoft DHCP Server must be running in order for the import file for **qip-setobject** to be created. The reason is that the DHCP Server stores the object names of its reserved (manual DHCP) addresses in its internal database. These are retrieved using a Microsoft-supplied CLI that requires the DHCP Server to be up and running.
- **qip-msextract** must be run on the machine that has both the Microsoft DHCP Server and at least the remote portion of VitalQIP installed.
- All subnets that exist within the Microsoft DHCP Server must be defined within VitalQIP. The Microsoft DHCP Server itself must also be defined. For more information on Microsoft servers, refer to Chapters 7 and 8 in the *VitalQIP Administrator Reference Manual.*

Synopsis

qip-msextract [**-d** ***directory***]

Parameters

qip-msextract recognizes the following optional parameter:

-d ***directory*** Specifies the directory where the CLI data files and batch file is created; if no directory is specified, the files are created in the directory specified by the environment variable QIPHOME.

Command line input example

- To create a file in *c:\temp*:

```
qip-msextract -d c:\temp
```

qip-namingpolicy

qip-namingpolicy retrieves and modifies an organization's naming policies.

Synopsis

```
qip-namingpolicy -n object_class_name [-g loginserver]
   [-s dataserver] [-u username] [-p password] [-o organization]
   [-f output_file] [-e errorfile]
   [-i input_file] [-ac action] [-df data_format]
```

Parameters

qip-namingpolicy recognizes the following parameters:

-n ***object_class_name***	Specifies the object class name.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***output_file***	A directory and file name for the CLI output. Use the **-df** parameter to specify the format. If no filename is provided, the output will be to *STDOUT*.
-i ***input_file***	Specifies the file that contains the input data.

-ac ***action***	Defines what action to take. Only one action may be performed at a time. You cannot get and modify in the same call, nor can you combine retrievals and modifications in one input file. The ***action*** variable can have the following values: **get** - retrieves a specific set of naming policies for an object class or a list of all object classes and naming policies: • To retrieve a specific set of naming policies for an object class, use the **-n** parameter with the object class name. The naming policy contains the object class name, the Use Default flag, prefix, length, and suffix. • To retrieve a list of all object classes and naming policies in the database, omit the **-n** parameter. **mod** - modifies a naming policy for an object class. Use the **-i** option to specify the input file that contains the data. Use the **-df** option to specify the format of the input file.
-df ***data_format***	Determines the output format. The ***data_format*** variable has two values: **c** and **n**. Use the **c** variable to produce comma delimited output. Use the **n** variable to produce name value pairs output. If the **-df** parameter is omitted from the command line, the default is name value pairs output. The output data format is identical to the input format shown in "Comma delimited file format" or "Name value pairs file format".
-e ***error_file***	Specifies the file that contains errors generated by this CLI.

Comma delimited file format

The first line of a comma delimited list file contains a layout of the items to be entered. The columns can be in any order and any optional columns can be excluded.

The next lines encountered are the data lines. There can be multiple data lines, each separated by a line feed.

Sample comma delimited input file

```
Object Class Name, Use Default, Prefix, Length, Suffix
IBM Workstation, True,,,
Lucent Router, False,crp,6,cpp
```

Name value pairs file format

Each line of the file starts with a field name. The valid field names are identical to the column names for the comma delimited file format. The fields can be listed in any order. There can be more than one object specified in an input file. Objects should be separated by a blank line.

Sample name value pair input file

```
Object Class Name=object class 1
Use Default=False
Prefix=abc
Length=6
Suffix=xyz

Object Class Name=object class 2
Use Default=False
Prefix=def
Length=6
Suffix=uvw
```

qip-objectclass

qip-objectclass adds, modifies, retrieves, and deletes object classes.

Synopsis

```
qip-objectclass [-n object_class_name] [-g loginserver]
   [-s dataserver] [-u username] [-p password] [-o organization]
   [-f output_file] [-i input file] [-e <errorfile>] [-ac action]
   [-df data format]
```

qip-objectclass recognizes the following parameters:y

-n ***object_class_name***	Specifies the object class name.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***output_file***	A directory and file name for the CLI output. Use the **-df** parameter to specify the format. If no filename is provided, the output will be to *STDOUT*.
-i ***input_file***	Specifies the file that contains the input data.
-e ***error_file***	Specifies the file that contains errors generated by this CLI.

-ac *action*

Defines what action to take. Only one action may be performed at a time. You cannot add and modify in the same call, nor can you combine additions and modifications in one input file.

The ***action*** variable can have the following values:

add - adds an object class. Use the **-i** option to specify the input file that contains the data. Use the **-df** option to specify the format of the input file.

mod - modifies an object class. Use the **-i** option to specify the input file that contains the data. Use the **-df** option to specify the format of the input file.

del - deletes an object class. The **-n** parameter is also required. The object class specified is removed from the database.

get - retrieves a specific object class or a list of all object classes:

- To retrieve a specific object class, use the **-n** parameter with the object class name. The object class profile contains the object class name, device type, prefix, length, and suffix.
- To retrieve a list of all object classes in the database, omit the **-n** parameter.

-df *data_format*

Determines the output format. The ***data_format*** variable has two values: **c** and **n**. Use the **c** variable to produce comma delimited output. Use the **n** variable to produce name value pairs output. If the **-df** parameter is omitted from the command line, the default is name value pairs output.

The output data format is identical to the input format shown in "Comma delimited file format" or "Name value pairs file format".

Comma delimited file format

The first line of a comma delimited list file contains a layout of the items to be entered. The columns can be in any order and any optional columns can be excluded.

The next lines encountered are the data lines. There can be multiple data lines, each separated by a line feed.

Sample comma delimited input file

```
Object Class Name, Device Type, Prefix, Length, Suffix
IBM Workstation, Host, iwp, 6, iws
```

```
Lucent Router, Router,crp,6,cpp
```

Name value pairs file format

Each line of the file starts with a field name. The valid field names are identical to the column names for the comma delimited file format. The fields can be listed in any order. There can be more than one object specified in an input file. Objects should be separated by a blank line.

Sample name value pair input file

```
Object Class Name=object class 1
Device Type=host
Prefix=abc
Length=6
Suffix=xyz

Object Class Name=object class 2
Device Type=host
Prefix=def
Length=6
Suffix=uvw
```

qip-postallocation

qip-postallocation allows the user to create multiple IPv4 Addresses within the specified subnet. Multiple IP Address (object) ranges of different types can be created with one CLI invocation. The subnet must be created before invoking this CLI.

Synopsis

```
qip-postallocation [-u <username>] [-p <password>]
   [-o <organization>] -i <input_file> [-g <loginserver>]
   [-s <dataserver>] [-v] [-h]
```

Parameters

qip-postallocation recognizes the following parameter:

-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-i *input_file*	Specifies the file that contains the input data.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *dataserver*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.

Command line input example

```
qip-postallocation -u admin -p passwd
   -i"/home/qadmin/objInput.xml"
```

Sample input file

This sample file includes one range of IP Addresses (objects) for Static, Dynamic and Reserved.

```
<com.lucent.qip.service.utils.CLISubnetOptions>
  <networkAddress>10.0.0.0</networkAddress>
  <subnetAddress>10.0.5.0</subnetAddress>
  <subnetCreateOption>5</subnetCreateOption>
  <objectList>
    <com.lucent.qip.service.utils.ObjectRange>
      <addressType>1</addressType>
      <domainName>example.com</domainName>
      <objectClass>Workstation</objectClass>
      <startOffset>40</startOffset>
      <endOffset>41</endOffset>
    </com.lucent.qip.service.utils.ObjectRange>
    <com.lucent.qip.service.utils.ObjectRange>
      <addressType>2</addressType>
      <dhcpOptionTemplate>general</dhcpOptionTemplate>

   <dhcpScopePolicyTemplate>DocTestScope</dhcpScopePolicyTemplate
   >
      <dhcpServer>doctest_dhcp.example.com</dhcpServer>
      <domainName>example.com</domainName>
      <leaseTime>7776000</leaseTime>
      <objectClass>PC</objectClass>
      <startOffset>42</startOffset>
      <endOffset>45</endOffset>
    </com.lucent.qip.service.utils.ObjectRange>
    <com.lucent.qip.service.utils.ObjectRange>
      <addressType>3</addressType>
      <domainName>example.com</domainName>
      <objectClass>Printer</objectClass>
      <startOffset>46</startOffset>
      <endOffset>47</endOffset>
    </com.lucent.qip.service.utils.ObjectRange>
  </objectList>
</com.lucent.qip.service.utils.CLISubnetOptions>
```

Input file description

Field Name	Description	Required/Optional
networkaddress	Address of network to which the subnet belongs.	Required
subnetAddress	Address of subnet to which IPaddresses(objects) are being added.	Required
subnetCreateOption	Must be **5** for creating IPAddresses.	Required.
addressType	Options are: **1** - Static **2** - Dynamic **3** - Reserved	Required
domainName	Domain associated with the subnet.	Required
objectClass	Specify one of the following: • Workstation • X-Terminal • PC; Printer • Server • Wiring_HUB • Router • Bridge • Terminal_Server • Switch • Legacy_System • Gateway • Test_Equipment • Undefined • Others • Partially_Managed	Required

Field Name	Description	Required/Optional
startOffset	Offset from the subnet start address at which this managed range starts. For example: If subnet address is 10.5.0.0 and IP Addresses being created are 10.5.0.3 - 10.5.0.5, the **`startOffset`** is **`3`**. **`startOffset`** cannot be 0. **`startOffset`** should not result in an IP Address that is greater than the maximum IP Address allowed for the subnet. If multiple ranges are specified, there should be no overlap. Addresses included in any of the specified ranges should not already exist in the database.	Required
endOffset	Offset from the subnet start address at which this managed range ends. For example: If the subnet address is 10.5.0.0 and IP Addresses being created are 10.5.0.3 - 10.5.0.5, the **`endOffset`** is **`5`**. **`endOffset`** cannot be 0. **`endOffset`** should not result in an IP Address that is greater than the maximum IP Address allowed for the subnet. If multiple ranges are specified, there should be no overlap. Addresses included in any of the specified ranges should not already exist in the database. **`endOffset`** must be greater than **`startOffset`**	Required
DHCP Server	The DHCP Server to be used for the Dynamic Scopes.	Required if Address Type is "Dynamic"; Not relevant otherwise.

Field Name	Description	Required/Optional
DHCP Option Template	DHCP Option Template to be associated with this scope of addresses.	Required if Address Type is "Dynamic". Not relevant otherwise.
DHCP Scope Policy Template	DHCP Scope Policy Template to be associated with this scope of addresses.	Optional . Relevant only if Address Type is "Dynamic".
Lease Time	Lease time of Dynamic Objects. Valid values are -1 for indefinite lease or between 3600 and 2147483647.	Required if Address Type is "Dynamic". Not relevant otherwise.
User Class	The User Class to be associated with this scope of addresses.	Optional. Relevant only if Address Type is "Dynamic".
Vendor Class	The Vendor Class to be associated with this scope of addresses.	Optional. Relevant only if Address Type is "Dynamic".

qip-qdhcplease

qip-qdhcplease is used in the upgrade from the VitalQIP DHCP 3.1 and 4.0 servers to the current version of the VitalQIP DHCP server format. The command converts existing VitalQIP DHCP lease database files into the new VitalQIP DHCP lease database file format.

Synopsis

qip-qdhcplease -d [***directory***]

Parameters

qip-qdhcplease recognizes the following optional parameter:

-d *directory* Specifies the directory where the existing lease database files reside. If the **-d** option is omitted, the lease database files are assumed to be in the current working directory. The conversion utility creates the new lease file in the same directory as the existing lease files.

Command line input example

To convert the existing lease database files in the *c:\temp* directory to the format of the current version of VitalQIP DHCP:

```
qip-qdhcplease -d c:\temp
```

qip-reclaim

qip-reclaim pings all addresses under the subnet, reclaims unused addresses and produces a report. The report is stored in *QIPHOME/report* directory with the name *SR<subnet_address>.yyyymmddHHMM* (for example, *SR135.111.104.0.200505111542*).

> **Note:** *yyyymmddHHMM* represents the year (yyyy), month (mm), day (dd), hour (HH), and minute (MM).

Although this CLI can be run as a stand alone CLI, **qip-reclaim** is designed to run on scheduled intervals as specified by the administrator of the subnet. (The reclaim schedules must be setup through the GUI or by using the **qip-setreclaimschedule** CLI.) **qip-reclaim** gathers object statistics on a subnet and generates a report each time it runs. On the last run it actually does the reclaim of the objects, or lets the administrator select a set of objects based on the gathered statistics, and reclaims them through the GUI.

For more information on the Reclaim functions of the VitalQIP interface, refer to the "Reclaim Addresses" section in Chapter 4 of the *VitalQIP User's Guide*.

Before you begin

- The Collection Frequency in the report heading indicates the frequency and the total number of times (as shown above) or the times during the day and number of days (that is, 8:00, 12:00, 16:00 For 5 Days).
- Statistics are stored and displayed for all addresses in the subnet, regardless of whether the address is assigned.
- Name, Object Class and Object Status are obtained from existing objects at the time the report is executed. The name may have changed several times over the reclaim cycle - only the current name is displayed.
- If the object is deleted, the Name and Object Class are blank in the report, the Object Status is Unused, but the other statistics cannot be reset – they continue to grow.
- The length of the object name is limited by the value of the OBJECT_NAME_LEN Global Report Policy.

Synopsis

```
qip-reclaim -a subnet_address -t report|reclaim|both
   [-e email_address] [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization] [-b]
```

Parameters

qip-reclaim recognizes the following parameters:

-a *subnet_address*	Specifies the subnet whose names or IP addresses you want to reclaim.

-t report|reclaim|both	Specifies the reclaim type. **report** only generates the report and does not reclaim the addresses in subnet. **reclaim** resets the addresses to be unused if applicable. **both** generates a report as well as resets the addresses. **Note:** Reclaim and both are valid only if the global policy 'AUTOMATIC_RECLAIM' was set to True.

-e *email_address*	Specifies a valid email address to which reclaim notices are sent.
	
	Note: On Windows platforms, you must enter a valid SMTP Host IP address in the *QSendMail.ini* file located in the system directory (the default value is *localhost*). For example:
	
	`[SendMail]`
	`SMTP Host=10.100.30.1`
	`IIS header=Yes`
	`Mime Encoding=Yes`
	`Return Path=sample@sample.com`

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-b	If **-t** is set for **reclaim** or **both** on a subnet which is scheduled, an abort needs to be forced with a **-b** before a reclaim can take place.

Command line input example

```
qip-reclaim -u qipman -p passwd -a 144.144.0.0
```

Output example

```
<<< MANUAL SCHEDULED RECLAIM REPORT >>>

Report Date:  10/26/2001 08:24Start Date:  10/24/2001 16:15
Collection Frequency: Collect: Every 20 Hours Total: 2 Times
SUBNET: 144.144.0.0
                    Object     Object      #    #       #         Reclaim
  Last Time
  Address      Name    Class     Status    Try  In DNS  Reachable    Status
  Reachable
```

```
-------------------------------------------------------------------------------
   ----------
144.144.0.1          Unused     Unused     2    0        0        Not In DNS;
   Unreachable
144.144.0.2          Unused     Unused     2    0        0        Not In DNS;
   Unreachable
```

qip-report

qip-report creates an SQL report based on the format file or parameters specified on the command line. The output is stored in the specified (or default) file.

Synopsis

```
qip-report -r range_type [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization]
   [-a ip_address|-n name][-c column_name[,column_name...]]
   |[-m format_file] [-t obj_dynamic_type] [-d delimiter]
   [-f report_file]
```

Parameters

qip-report recognizes the following parameters:

-r ***range_type***	Specifies the type of report you want (for example, corporation, domain, subnet, subnet organization, network, OSPF area).
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-a ***ip_address*** **|** **-n** ***name***	Specifies the address corresponding to the specified range type (if it is a subnet or network) or the name of a specified range type (domain, subnet organization, OSPF, or Corporation) to be reported on.
-c ***column_ name [,column_ name…]***	Specifies the name(s) of the database table columns to be included in the report (to streamline your report). You can code multiple column names, separated by commas. Refer to Table 1-19 following. The default is *all* columns.
-m ***format_file***	Specifies the file that defines the report requirements. It can include information on column values to be used to filter the report. The specifiable column names are marked with an asterisk (*) in Table 1-19 following. (Refer to Note 2 for details.)
-t ***obj_dynamic_type***	Specifies the type of object (for example, dynamic, static, or reserve).
-d ***delimiter***	Specifies the delimiter character to separate fields in the output file (such as **-d |**). The default delimiter is a comma.

-f *report_file* Specifies the name of the file where the report is returned. The default is *STDOUT*.

Table 1-19 Column names

Column name	Additional notes
MacAddr	Can be used in the format file to filter the report.
ObjIpAddr	IP address of the object.
Name*	Can be used in the format file to filter the report (refer to Note 2).
SubnetName*	Can be used in the format file to filter the report (refer to Note 2).
SubnetAddress*	Can be used in the format file to filter the report (refer to Note 2).
FTPServerName*	Can be used in the format file to filter the report (refer to Note 2).
BootfileName*	Can be used in the format file to filter the report (refer to Note 2).
NameServices	
nameservicerecs	
AuthorizationName	
ObjectType	Type of object that can be passed as the **-t** parameter.
ObjectClass*	Can be used in the format file to filter the report (refer to Note 2).
ObjectDesc	
DynamicType*	Can be used in the format file to filter the report (refer to Note 2).
HardwareType	
ModelType*	Can be used in the format file to filter the report (refer to Note 2).
ClientClass	
UserClass	
ObjectTag	
DHCPServer*	Can be used in the format file to filter the report (refer to Note 2).
DHCPTemplate*	Can be used in the format file to filter the report (refer to Note 2).
LeaseTime*	Can be used in the format file to filter the report (refer to Note 2).
TTLTime*	Can be used in the format file to filter the report (refer to Note 2).
Application*	Can be used in the format file to filter the report (refer to Note 2).
Manufacturer*	Can be used in the format file to filter the report (refer to Note 2).
SerialNumber*	Can be used in the format file to filter the report (refer to Note 2).

Column name	Additional notes
AssetNumber*	Can be used in the format file to filter the report (refer to Note 2).
HostID*	Can be used in the format file to filter the report (refer to Note 2).
PurchaseDate	
ExpirationDate	
Contact	
Location	
RoomID	
BillStatus*	Can be used in the format file to filter the report (refer to Note 2).
BillLocation*	Can be used in the format file to filter the report (refer to Note 2).
BillGroup*	Can be used in the format file to filter the report (refer to Note 2).
BillObject*	Can be used in the format file to filter the report (refer to Note 2).

Note: You can use a format file (specified in the **-m** parameter) that contains the **column=*ColumnName[,ColumnName…]*** statement to define the columns of data to be output (such as **column=Name,Type,SubnetAddress**). Each output column is displayed consecutively, from left to right, even if you define multiple **column=** statements. You can specify any of the column names shown in the table above.

Note: If the column names in the table above are marked with an asterisk (*), you can also use them to filter the report by including them (in the format **ColumnName=value**) in the "format file" you specified in the **-m** parameter. For example, your format file can contain the record **SubnetAddress=150.1.0.0** to limit the report to records in which the Subnet Address is **150.1.0.0**. Example 1 shows the use of the **-m** parameter.

Command line input example 1

- The following example includes the **–m** parameter (to specify the format file that contains report format details):

qip-report -r domain -n qtek.com -m rpt.fmt

- The *rpt.fmt* format file specified in the **-m** parameter contains:

```
Column=Name,DynamicType,SubnetAddress
Column=ObjIpAddr
Application=qtek app, mci_app
SubnetAddress=150.1.0.0
```

Output example 1

The resulting output is:

```
Name,DynamicType,SubnetAddress,ObjIpAddr,Application,Billing
   Location
"hub.au.com","",150.1.0.0,150.1.0.1,qtek app,malvern
"dc.au.com","",150.1.0.0,150.1.0.2
"wpenn.au.com","",150.1.0.0,150.1.0.3,mci_app,boston
"philly.au.com","",150.1.0.0,150.1.0.4,qtek app
"geneva.au.com","",150.1.0.0,150.1.0.5,qtek app,malvern
"alps.au.com","",150.1.0.0,150.1.0.6,mci_app
"bootp1.au.com","",150.1.0.0,150.1.0.14,mci_app
```

Command line input example 2

- The following example includes the **– c** parameter (to limit the report to specific columns):

```
qip-report -c ObjIpAddr,Name -r subnet -a 150.231.35.160
```

Output example 2

The resulting output is:

```
ObjIpAddr,Name
150.231.35.169,pc1.lucent.com
150.231.35.168,pc2.lucent.com
150.231.35.164,pc3.lucent.com
150.231.35.163,pc4.lucent.com
150.231.35.161,pc5.lucent.com
```

qip-rot13

qip-rot13 encrypts a text string by using the ROT13 weak encryption algorithm. The output is stored in the specified (or default) file.

Synopsis

qip-rot13 -i *input_string* [**-f *filename***]

Parameters

qip-rot13 recognizes the following parameters:

-i *input_string* Specifies the string to be scrambled.

-f *filename* Specifies the file where the information is returned. The default is *STDOUT*.

Command line input example

```
qip-rot13 –i atest
qip-rot13 -i "test rot"
```

Output example

```
ngrfg
grfg ebg
```

qip-rptadminaudit

qip-rptadminaudit creates an Administrator Audit Report. The output is stored in the specified (or default) file.

Synopsis

```
qip-rptadminaudit [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] -n login_name [-c]
   [-d start_date] [-e end_date] [-f filename]
```

Parameters

qip-rptadminaudit recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-n *login_name*	Specifies the VitalQIP administrator login name.
-c	Specifies that the output is in CSV format.
-d *start_date*	Specifies the audit period start date (format: mm/dd/yyyy).
-e *end_date*	Specifies the audit period start and date (format: mm/dd/yyyy).
-f *filename*	Specifies the file where the report is stored. The default is *STDOUT*.

Command line input example

```
qip-rptadminaudit –u qipman –p passwd –n qipman
```

Output example

```
VitalQIP Administrator Audit Report
Date        : 1998-07-28 13:47
Organization: VitalQIP Organization
User Name   : qipman
Object Name
IP Address. : 161.251.45.79
Domain.......................
Class.......: Undefined
Name services: N
TTL time: 0
```

qip-rptadminrole

qip-rptadminrole creates an Administrative Role Report with assigned administrators to the role. The output is stored in the specified (or default) file.

Synopsis

qip-rptadminrole [**-g** ***loginserver***] [**-s** ***servername***] [**-u** ***username***] [**-p** ***password***] [**-o** ***organization***] **-n** ***role_name*** [**-c**] [**-f** ***filename***]

Parameters

qip-rptadminrole recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-n ***role_name***	Specifies the VitalQIP administrative role name.
-c	Specifies that the output is in CSV format.
-f ***filename***	Specifies the file where the report is stored. The default is *STDOUT*.

Command line input example

```
qip-rptadminrole -u qipman -p passwd -n "admin role name" -f adminrole.txt
```

Output example

```
VitalQIP Role/Administrator Report
Date        : 2000-02-09 13:42
Organization: VitalQIP Organization
User Name   : qipman
Role: testrole
    user1
```

qip-rptaudithistory

qip-rptaudithistory creates an Object History Report. This report is retrieved one object at a time. If there is more than one object with that "name", an error is sent stating that there are multiple objects with that name. The output is stored in the specified (or default) file.

Synopsis

```
qip-rptaudithistory -a ip_address|-n name [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-d start_date] [-e end_date] [-c] [-f filename]
```

Parameters

qip-rptaudithistory recognizes the following parameters:

-a ***ip_address*** **|** **-n** ***name***	Specifies the IP address of the object to be audited or the name of the object to be audited.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-d ***start_date***	Specifies the audit period start and date (format is mm/dd/yyyy).
-e ***end_date***	Specifies the audit period start and date (format is mm/dd/yyyy).
-c	Specifies that the output is in CSV format.
-f *filename*	Specifies the file where the report is stored. The default is *STDOUT*.

Command line input example

```
qip-rptaudithistory –u qipman –p passwd –a 10.200.80.10
```

Output example

```
VitalQIP Object Audit Report
Date        : 2005-05-20 14:30
Organization: VitalQIP Organization
```

```
User Name   : qipman

ADDED                02/11/2005 11:31:55 qipman

Object Name..................... qahp1104
IP Address...................... 10.200.80.10
Domain......................... seg1.qa.quadritek.com
Class.......................... Server
Subnet Address................. 10.200.80.0
Subnet Mask.................... 255.255.255.0
Subnet Name....................
Name services................. Y (A and PTR records)
Dynamic pushes................. A, PTR, CNAME, MX
TTL time....................... Unlimited
            MISC. OBJECT INFORMATION:
            OBJECT 3:
            OBJECT 4:
            OBJECT 5:
            OBJECT INFORMATION:
            OBJECTMAXCHARACTERS11111111112:
            OBJECTMAXCHARACTERS1111111111E:

   Date/Time          Administrator                 Audit Activity
```

qip-rptdhcp

qip-rptdhcp creates a DHCP Server Profile Report. This report can be generated in a compressed format. The output is stored in the specified (or default) file.

Synopsis

```
qip-rptdhcp -n dhcp_FQN [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization]
   [-t compressed|expanded] [-c] [-f filename]
```

Parameters

qip-rptdhcp recognizes the following parameters:

-n ***dhcp_FQN***	Specifies the fully-qualified name of the DHCP server on which you are seeking a Server Profile Report.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-t compressed|expanded	Specifies the report type (compressed or expanded).
-c	Specifies that the output is in CSV format.
-f ***filename***	Specifies the file where the report is stored. The default is *STDOUT*.

Command line input example

```
qip-rptdhcp –u qipman –p passwd –t expanded –n srt08.seg5.qa.quadritek.com
```

Output example

```
VitalQIP DHCP Report
Date        : 2005-05-20 14:37
Organization: VitalQIP Organization
User Name   : qipman
DHCP Server : srt08.seg5.qa.quadritek.com
Report Type : Expanded

Server Name.........................................
   srt08.seg5.qa.quadritek.com
```

```
Managed Range..................................... Corporation
Default Directory................................. /opt/qip/dhcp
DHCP Template..................................... general
Accept Client Names............................... True
Additional Policies...............................
Client Class......................................
Debug Information................................
    Debug............................................ None
    DebugFile........................................ dhcpd.log
    MaxDebugFileSize................................. -1
Failover Server Type.............................. Standalone/Primary
    CtlReqRetryMax................................... 3
    PollDelay........................................ 60
    SyncBindRetryMax................................. 3
    SyncBindingBufSize............................... 1024
    Use Failover Server.............................. False
    WaitCtlRetSecs................................... 5
    WaitSyncBindAckSecs.............................. 5
    WaitSyncBindUpdateSecs........................... 15
Remote Server Proxy...............................
    Description......................................
Scheduled Automatic Updates....................... None
Support Bootp..................................... True
Use Server Policy Template........................ False
    DHCP Server Policies.............................
        ActiveLeaseExpiration........................... Off
        Bootfile........................................ Default
        CheckTransactionID............................... False
        ClientHostNameProcessing......................... Ignore
        CompressedLog.................................... False
        DefaultDescentThreshold.......................... 0
        DefaultLease.................................... 90 Days 0 Hours
   0 Minutes
        DefaultUnavailableThreshold...................... 0
        DropZeroMacAddressPackets........................ True
        ExpireAllLeasesOnRestart......................... True
        ForceClass....................................... None
        HonorRequestedLeaseTime.......................... True
        HonorUnqualifiedBootfile......................... False
        InitRebootAddressShuffle......................... Off
        IssueDropUnknownClientTrap....................... False
        LeaseExpirationSleepTime......................... 60000
        LeaveBootpParametersInOptions.................... False
        LogLeaseGrantAndRenew............................ True
        MacWarningsToEventLog............................ False
```

```
        MaxOutgoingDhcpMessageSize........................... 1024
        MaxPendingSeconds.................................... 10
        MaxUnavailableTime................................... 86400
        NackDhcpRequestsForDuplicates........................ True
        NakUnknownClients.................................... True
        Option81Support...................................... Suppress
        PadBootpReply........................................ True
        PingAttempts......................................... 1
        PingBeforeManualBootp............................... False
        PingBeforeManualDhcp................................. True
        PingDelay............................................ 0
        PingRetention........................................ 0
        PingSendDelay........................................ 0
        RegisteredClientsOnly............................... True
        RenewAddressShuffle................................. Off
        RenewAddressShuffleMaxCount.......................... 0
        SearchDynamicFirst................................... False
        SendRequestedParamsOnly............................. False
        SendServerIdLast..................................... False
        ShareAutoBootpAndDynDhcp............................ False
        SupportAutoRelease................................... True
        SupportBootpAutoRelease............................. True
        SupportEncodingLongOptions........................... False
        SupportRelayAgentDeviceClass........................ False
        SupportRelayAgentOption............................. True
        SupportSubnetSelection.............................. False
        ThresholdMonitorSleepTime........................... 60
        Update QIP Operations...............................
            Autorelease........................................ True
            Bootp.............................................. True
            Decline............................................ True
            Delete............................................. True
            Expiration......................................... True
            Grant.............................................. True
            Release............................................ True
            Renew.............................................. True
        ZeroCiAddr........................................... False

Subnet Address: 10.58.208.0                    Subnet Name:
Subnet Mask   : 255.255.240.0                  Effective Domain:
   seg5.qa.quadritek.com
Default Router(s):

MAC Pool:
```

```
  10.58.208.4   --10.58.208.4   general
  D-DHCP  ---                  90 Days 0 Hours 0 Minutes
  seg5.qa.quadritek.com
  10.58.208.6   --10.58.208.6   general
  D-DHCP  ---                  90 Days 0 Hours 0 Minutes
  seg5.qa.quadritek.com
  10.58.208.8   --10.58.208.8   general
  D-DHCP  ---                  90 Days 0 Hours 0 Minutes
  seg5.qa.quadritek.com
  10.58.208.10  --10.58.208.12  general
  D-DHCP  ---                  90 Days 0 Hours 0 Minutes
  seg5.qa.quadritek.com
  10.58.208.14  --10.58.208.24  general
  D-DHCP  ---                  90 Days 0 Hours 0 Minutes
  seg5.qa.quadritek.com
  10.58.208.26  --10.58.214.50  general
  D-DHCP  ---                  90 Days 0 Hours 0 Minutes
  seg5.qa.quadritek.com

Option Template general:
    Subnet Mask..................................... Same as in Subnet
  Profile
    Router.......................................... Same as in Subnet
  Profile
    Domain Name Server.............................. Same as in Subnet
  Profile
    Domain Name..................................... Same as in Subnet
  Profile

MAC Pool:
```

qip-rptfreesubnet

qip-rptfreesubnet creates a report on free subnets for a given network and subnet mask. It only reports on "unallocated" subnets for a specific network using a specific mask. The output is stored in the specified (or default) file.

Synopsis

qip-rptfreesubnet -a *network_IP* -m *subnet_mask* [**-g *loginserver***] [**-s *servername***] [**-u *username***] [**-p *password***] [**-o *organization***] [**-c**] [**-f *filename***]

Parameters

qip-rptfreesubnet recognizes the following parameters:

-a *network_IP*	Specifies the network IP address where the free subnets are located.
-m *subnet_mask*	Specifies the subnet mask.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-c	Specifies that the output is in CSV format.
-f *filename*	Specifies the file where the report is stored. The default is *STDOUT*.

Command line input example

```
qip-rptfreesubnet –a 100.101.0.0 –m 255.255.255.0
```

Output example

```
VitalQIP Free Subnet Report
Date        : 1998-07-28 12:59
Organization: VitalQIP Organization
User Name   : qipman
    100.101.0.0
    100.101.1.0
    100.101.2.0
    100.101.3.0
    100.101.4.0
    100.101.5.0
```

qip-rptinquire

qip-rptinquire generates an Inquiry Report on the object specified in the command parameter. The output is stored in the specified (or default) file.

Synopsis

```
qip-rptinquire -n name|-a address [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   {-t search_type [[-b sub_searchType]
   {[-r subRange <rrType|udf_fieldname>] [-d owner|data|both]}]}
   [-c] [-f filename]
```

Parameters

qip-rptinquire recognizes the following parameters:

-n ***name*** **| -a** ***address***	Specifies the name of the object ***or*** the IP address of the object.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-t ***search_type***	The search type of the object. Refer to Table 1-20 following.
-b ***sub_searchType***	Applicable if the **–t** search_type is User_Field or Resource_Record. Refer to Table 1-21 following.
-r ***subRange***	Applicable if the **–t** search_type is User_Field or Resource_Record. Specifies the resource record type or the user-defined field names listed in Table 1-21 following.
-d owner|data|both	Applicable only if the **–t** search_type is Resource_Record.
-c	Specifies that the output is in CSV format.
-f ***filename***	Specifies the file where the report is stored. The default is *STDOUT*.

Table 1-20 Search type fields

Search_Type	Notes
All	Searches all supported fields.
Corp	Searches for an Organization, use **-n** to specify the organization name.

Search_Type	Notes
Subnetorg	Searches for a Subnet Organization, use **-n** to specify the subnet organization name.
Domain	Searches for a domain, use **-n** to specify the domain name.
Ospf	Searches for an OSPF Area name, use **-n** to specify the OSPF Area name.
Network	Searches for a network. Use **-n** to search for the network name. Use **-a** to specify the network address.
Object	Searches for an object. Use **-n** to search for the object name. Use **-a** to specify the object address.
Router_Group	Searches for router group names. Use **-n** to specify the router group name.
Resource_Record	Searches for the match of the string passed on the **Resource Record Owner** field.
User_Field	Searches for the match of the string passed on the **User Defined** field.

Table 1-21 SubSearch type fields

Search_Type	SubSearch_Type
Resource_Record	Domain, reversezone, object.
User_Field	Domain, reversezone, organization, object, subnet, user, all.

Command line input example

```
qip-rptinquire -u qipman -p passwd -t network -a 176.200.0.0
```

Output example

```
VitalQIP Inquiry Report
Date        : 1998-07-28 12:10
Organization: VitalQIP Organization
User Name   : qipman

Type                 Address         Name
--------------------------------------------------
Network              176.200.0.0     net_176
```

qip-rptmanaged

qip-rptmanaged displays the list of administrators who have been assigned the specified infrastructure item as an element of their managed list.

Synopsis

```
qip-rptmanaged -t owner_type -n name |-a address[-m access_mode]
   [-g loginserver] [-s servername][-u username] [-p password]
   [-o organization] [-f report_file]
```

Parameters

qip-rptmanaged recognizes the following parameters:

-t ***owner_type***	Specifies the type of item to search for. Options are domain, network, OSPF, addr_range, obj_range, user_group, subnet_org, subnet, application, object.
-n ***name***	Specifies the name of the item. Names are used for Domain, OSPF, Subnet Organization, Application, User Group, and Server.
-a ***ip_address***	Specifies the IP Address or IP Address Range for Network, Subnet, Object, Address Range, and Object Range ***owner_types***. Address Range and Object Range are specified as ***start_ip_address*** – ***end_ip_address***.
-m ***access_mode***	Specifies if you want the list of administrators who have **read** or **write** access to the name or address. The default ***access_mode*** is **write**.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***report_file***	Specifies the file where the report is written. If **-f** is not specified, the report is written to *STDOUT*.

qip-rptobjectlst

qip-rptobjectlst creates a report of a list of objects associated with the object owner passed in the command line. The owner of the objects can be one of the following: corporation, domain, OSPF, subnet, organization, application, administrator, location, subnet, or network. The format of the report can be a CSV file if the **-c** parameter is specified. The output is stored in the specified (or default) file.

Synopsis

```
qip-rptobjectlst -a net/subnetIP|-n name -t owner_type
   [-g loginserver] [-s servername] [-u username] [-p password]
   [-o organization] [-l alloc_type] [-d dynamic_type]
   [-j obj_class] [-c] [-f filename]
```

Parameters

qip-rptobjectlst recognizes the following parameters:

-a ***net/subnetIP*** **|** **-n** ***name***	Specifies the address of the range (for example, network or subnet) or the name of the range from which you want to retrieve the list of objects.
-t ***owner_type***	Specifies the owner type under which IP objects belong. Refer to Table 1-22 following.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-l ***alloc_type***	Specifies the Allocation type of the objects to report. Refer to Table 1-23 following for valid values.
-d ***dynamic_type***	Specifies the Dynamic type of object. Refer to Table 1-24 following for valid values. **Note:** Valid only when the Allocation_Type is "Dynamic".
-j ***obj_class***	Specifies the object class for which you want to retrieve the list of manufacturers. The following object classes are available:

	- Workstation - X-terminal - PC - Printer - Server - Wiring_HUB - Router - Bridge	- Terminal_Server - Switch - Legacy_System - Gateway - Test_Equipment - Any user-defined object class - Undefined - Others
-c	Requests output in CSV format.	
-f *filename*	Specifies the file where the report is stored. The default is *STDOUT*.	

Table 1-22 Owner type fields

Owner_Type	Notes
Corp	Displays objects for an Organization, use **-n** to specify the organization name.
Subnetorg	Displays objects for a Subnet Organization, use **-n** to specify the subnet organization name.
Domain	Displays objects for a domain, use **-n** to specify the domain name.
OSPF	Displays objects for an OSPF Area, use **-n** to specify the OSPF area name.
Network	Displays objects for a network. Use **-a** to specify the network address.
Subnet	Displays objects for a subnet. Use **-a** to specify the subnet address.
Admin	Displays objects for an administrator. Use **-n** to specify the administrator login name.
Application	Displays objects for an application. Use **-n** to specify the application name.
Location	Displays objects for the specified location ID. Use **-n** to specify the location ID. The location ID can be obtained from a call to **qip-getloclst**.

Table 1-23 Allocation types fields

Allocation_Types	Description
All	All objects regardless of assignment.
Static	An object that has been statically assigned.
Reserved	An object that has been reserved.
Dynamic	An object that has been dynamically assigned.

Allocation_Types	Description
Scheduled	An object that is scheduled to be moved.

Table 1-24 Dynamic types fields

Dynamic_Types	Description
M-BOOTP	The address was dynamically assigned using manual bootp.
M-DHCP	The address was dynamically assigned using manual DHCP.
A-BOOTP	The address was dynamically assigned using automatic bootp.
A-DHCP	The address was dynamically assigned using automatic DHCP.
D-DHCP	The address was dynamically assigned using dynamic DHCP.
All	All dynamic objects
None	Dynamic (None) objects.

Command line input example

```
qip-rptobjectlst -u qipman -p passwd -a 10.0.0.0 -t subnet
```

Output example

```
VitalQIP Object List Report
Date        : 1998-07-28 11:51
Organization: VitalQIP Organization
User Name   : qipman
Range: Domain qtek.com
    Server(s):
        DN00CORP.qtek.com
            DN00DNS1.qtek.com
            DN00DNS2.qtek.com
            DN73DNS1.qtek.com
            DN74DNS1.qtek.com
            DN75DNS1.qtek.com
Object Class: ALL                  Object Type: ALL

    Network 10.0.0.0 net_10.0.0.0
        Subnet 10.98.60.0 Name: SN10.98.60.0 Status: Y
            IP Address      Object Name           Obj Class
             -------------------------------------------------------
        10.98.60.10     BHAM3000.qtek.     Undefined
        Subnet 10.98.62.0 Name: SN10.98.62.0 Status: Y
```

qip-rptrole

qip-rptrole generates an administrative role managed list report. The output is stored in the specified (or default) file.

Synopsis

qip-rptrole [**-n** ***role_name***] [**-g** ***loginserver***] [**-s** ***servername***] [**-u** ***username***] [**-p** ***passwd***] [**-o** ***organization***] [**-c**] [**-f** ***filename***]

Parameters

qip-rptrole recognizes the following parameters:

-n ***role_name***	Specifies the VitalQIP administrative role name.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-c	Specifies that the output is in CSV format.
-f filename	Specifies the file where the report is stored. The default is *STDOUT*.

Command line input example

```
qip-rptrole –u qipman –p passwd –f output.txt
```

Output example

```
VitalQIP Role Managed List Report
Date        : 2005-05-20 14:39
Organization: VitalQIP Organization
User Name   : qipman
Role: tester1
    Domain: qa.quadritek.com
    Domain: quadritek.com
    Domain: do.main
    Subnet Organization: SubOrg
    Subnet Organization: 99
Role:  tester2
    Domain: seg1.qa.quadritek.com
    Subnet Organization: ethel
    Address Range: 10.200.100.128,10.200.100.191
```

qip-rptzonerr

qip-rptzonerr generates a report on the resource records for the specified zone. The output is stored in the specified (or default) file.

Synopsis

```
qip-rptzonerr -n zone_name|-a zone_ip/mask [-g loginserver]
   [-s servername] [-u username] [-p password]
   [-o organization] [-r rr_types] [-e] [-f filename]
```

Parameters

qip-rptzonerr recognizes the following parameters:

-n ***zone_name***	Specifies the name of the zone.
-a ***zone_ip/mask***	Specifies the reverse zone IP address and the zone mask.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-r ***rr_types***	Specifies the type of resource record to be reported (for example, CNAME, NS, WKS, and so on). The default is All. The -r option does not return SOA records for a zone. If NS is specified with the `-r` option, only the NS records that are added to the Resource Record tab of the Domain are listed, not all the NS records from within the zone file.
-e	Specifies the zone extension is to be included
-f ***filename***	Specifies the file where the report is stored. The default is *STDOUT*.

Command line input example

```
qip-rptzonerr -u qipman -p passwd -n qtek.com -r A
```

Output example

```
************************************
; Other object level resource records
;************************************
nnsa2.qtek.com.          IN      A       192.85.154.30
;************************************
; Other domain level resource records
```

```
;************************************
ahmlir3.mail.qtek.com.   IN      A       20.191.24.43
ahmlir2.mail.qtek.com.   IN      A       20.191.24.42
ahmlir1.mail.qtek.com.   IN      A       20.191.24.41
y2ktipskb.csts.qtek.com. IN      A       20.230.144.234
```

qip-scope

qip-scope defines a DHCP scope by selecting a set of addresses and assigning it an Object Profile. This can be performed by specifying individual parameters within the CLI command or creating a file with all this information in it and placing the filename (**-f** ***input_file***) in the parameter.

Synopsis

```
qip-scope [-g loginserver] [-s servername] [-o organization]
   [-u username] [-p password] -a subnet_address -t dhcp_type
   -r start,end_Address -d domn_name -m name_service(Y/N)
   [-w dynamic_update] -n dhcp_server -e dhcp_template
   [-c object_class] [-k vendor_class] [-l lease_time]
   [-x user_class] [-q error_file]
```

or (to specify a file containing the parameter information)

```
qip-scope -f input_file [-j reject_file] [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-q error_file]
```

Parameters

qip-scope recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-a ***subnet_address***	Specifies the subnet whose IP address(es) is associated with this DHCP scope.
-t ***dhcp_type***	Specifies a DHCP type. The options are A(automatic) or D(dynamic).
-r ***start,end_Address***	Specifies the beginning and ending address that defines the DHCP scope of addresses, separated by a comma.
-d ***domn_name***	Specifies the fully-qualified domain name.
-m ***name_service***	Specifies "Y" or "N" to register this object with the name service.
-w ***dynamic_update***	Specifies "Y" or "N" to turn on or off the dynamic update flags.

-n ***dhcp_server***	Specifies the fully-qualified DHCP server name to which you are assigning this scope.
-e ***dhcp_template***	Specifies the DHCP template to be assigned to this scope.
-c ***object_class***	Specifies the object class description.
-k ***vendor_class***	Specifies the vendor class to be assigned to this scope.
-l ***lease_time***	Specifies the lease time to be assigned to this scope.
-x ***user_class***	**QDHCP servers only:** Specifies user class to be assigned to scope.
-q ***error_file***	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.

or

-f ***input_file***	Specifies the scope definition format filename, as shown in Table 1-25.
-j ***reject_file***	Specifies the file where records that were not added or modified are placed.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the database server name. This value is the value of QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-q ***error_file***	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.

The **-f** ***input_file*** for **qip-scope** is in CSV format, with fields as described in Table 1-25:

Table 1-25 qip-scope file layout

Field	Description
SubnetAddress	In dotted decimal format. ***Required***.
DHCPType	A-DHCP=Automatic or D-DHCP=Dynamic. ***Required***.
StartAddress	In dotted decimal format. ***Required***.
EndAddress	In dotted decimal format. ***Required***.
FullyQualifiedDomainName	***Required***.
UseNameService	Y=Yes, N=No. Optional.

Field	Description
DynamicUpdate	Y=Yes, N=No. Optional
ObjectClass	Optional.
FullyQualifiedDHCPServerName	Required.
DHCPTemplateName	Required.
VendorClass	Optional.
LeaseTime	Optional.
UserClass	Optional.

Command line input example 1

```
qip-scope -u qipman -p passwd -a 199.200.139.0 -t A -r
   199.200.139.10,199.200.139.63
   -c PC -n dhcpserver.bedrock.com -d bedrock.com -e general -m Y -x class1 -
   l 300
```

Command line input example 2

```
qip-scope -f parmfile
```

The parameters file (specified as **-f parmfile** in this example) contains the following two records:

```
10.200.80.0,D,10.200.80.2,10.200.80.16,quadritek.com,Y,Y,PC,
   dhcpserver1.quadritek.com,10.200.80.0_Template,,120,
10.201.80.0,A,10.201.80.1,10.201.80.1,quadritek.com,Y,N,PC,
   dhcpserver1.quadritek.com,10.201.80.0_Template,,4294967295,uclass
```

qip-search

qip-search searches through various name or address fields in the database and returns the address, name, and object type of records matching the search criteria. The output is stored in the specified file (or default file), in a | delimited format.

Note: Wildcard searches are permitted when searching for a name. Use the asterisk (*) as the wildcard character, and surround the criteria with double quotes (for example, **"p*"**).

Synopsis

qip-search -n *name*|-a *address* [-g *loginserver*] [-s *servername*] [-u *username*] [-p *password*] [-o *organization*] [-t *search_type*] [-b *sub_search_type*] [-r *subRange*] [-d owner|data|both] [-f *filename*]

Parameters

qip-search recognizes the following parameters:

-n *name* **|-a *address***	Specifies either the name (***not*** fully-qualified: for example, sec1 ***not*** sec1.bedrock.com) or the IP address of the object for which you are searching.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-t *search_type*	Specifies the search type that you want to search for. Refer to Table 1-26 following. **Note:** If the ***search_type*** is **UserFields**, you must also use the **-b** parameter for the UDF type and the **-n** parameter to specify the User Defined Field value or use a wildcard combination.
-b *sub_search_type*	The search type within the ***search_type*** value (above). This option is only relevant when your ***search_type*** is "**Resource Record**" or **UserFields**. Refer to Table 1-28 following.
-r *subRange*	Specifies the "**Resource Record**" type or **UserFields** name. Refer to Table 1-27 following.

-d owner|data|both Specifies the search target (owner, data, or both). This parameter is only relevant if your ***search_type*** is "**Resource Record**".

-f *filename* Specifies the file where the information is stored. The default is *STDOUT*.

Table 1-26 Search types

Search type	Notes
All	Searches all supported fields.
Alias	Searches for an object's alias names.
Corp	Searches for an Organization, use **-n** to specify the organization name.
Subnetorg	Searches for a Subnet Organization, use **-n** to specify the subnet organization name.
Domain	Searches for a domain, use **-n** to specify the domain name.
OSPF	Searches for an OSPF Area name, use **-n** to specify the OSPF area name.
Network	Searches for a network. Use **-n** to search for the network name. Use **-a** to specify the network address.
Object	Searches for an object. Use **-n** to search for the object name. Use **-a** to specify the object address.
"Router Group"	Searches for router group names. Use **-n** to specify the router group name.
"Mac Address"	Searches for a MAC address. Use **-a** to specify the MAC address.
UserFields	Searches for values of user defined fields. Use **-n** to specify the value to search for. Wildcards are valid.
"DECNet Address"	Searches for a specific DECNet Address. Use **-n** to specify the value to search for. Enter the search criteria in AREA,NODE format.
"Resource Record"	Searches for the match of the string passed on the Resource Record Owner field.

Table 1-27 Resource Record types

Record type	Description
CNAME	Canonical Name
A	Host IPv4
HINFO	Host Information

Record type	Description
MX	Mail Exchange
NS	Name Server
PTR	Pointer
TXT	Text
WKS	Well Known Services
AAAA	Host IPv6
AFSDB	Andrew File System
MB	Mailbox Name
MG	Mail Group
MINFO	Mailbox Information
MR	Mail Rename
ISDN	Integrated Services Digital Network
SRV	Server Resource Record
X25	PSDN (Public Switched Data Network) address

Refer to Table 1-28 only if the specified ***search_type*** is **"Resource Record"** or **UserFields**.

Table 1-28 Sub search types

Sub_Search types	Notes
Object	For resource records or user fields associated with objects. The search type can be **"Resource Record"** or **UserFields**.
Domain	For resource records or user fields associated with domains. The search type can be "**Resource Record"** or **UserFields**.
ReverseZone	For resource records or user fields associated with reverse zones. The search type can be **"Resource Record"** or **UserFields**.
Organization	The user fields defined for organizations. The search type must be **UserFields**.
Subnet	The user fields associated with subnets. The search type must be **UserFields**.

Sub_Search types	Notes
User	The user fields associated with users. The search type must be **`UserFields`**.
All	All of the above sub-search types are included.

Command line input example 1

- To search for all objects that begin with the string “pc”. The search results are returned in the *search.txt* file:

```
qip-search -u qipman -p passwd -t Object -n "pc*" -f search.txt
```

Output example 1

```
10.98.4.10|pc0100.qtek.com|OBJECT
10.98.4.11|pc0200.qtek.com|OBJECT
10.98.4.12|pc0300.qtek.com|OBJECT
10.98.4.13|pc0400.qtek.com|OBJECT
10.98.4.14|pc0500.qtek.com|OBJECT
10.98.4.15|pc0600.qtek.com|OBJECT
10.98.4.16|pc0700.qtek.com|OBJECT
10.98.4.17|pc0800.qtek.com|OBJECT
10.98.4.18|pc0900.qtek.com|OBJECT
```

Command line input example 2

- To search for the subnet in the specific address and return the search results in the *search.txt* file:

```
qip-search -u qipman -p passwd -a 199.200.139.0 -t Subnet -f search.txt
```

Output example 2

```
198.200.139.0|sn_139|Subnet
```

Command line input example 3

- To search the records in the resource record with owner fields that consist of “qtek” and limit the search to the A records associated with the object. The search results are returned in the *search.txt* file:

```
qip-search -u qipman -p passwd -n qtek -t "Resource Record" -b Object
   -d owner -r A -n "*"-f search.txt
```

Output example 3

```
dns1.qtek.com A 10.100.23.1|obj1.qtek.com|OBJECT
   dhcp1.qtek.com A 10.100.23.100|obj2.qtek.com|OBJECT
```

Command line input example 4

- To search for all user-defined fields used for domains where the value starts with "West":

```
qip-search -t UserFields -b Domain -n "West*"
```

Output example 4

```
West Coast|domain.com|DOMAIN/REGION
```

Command line input example 5

- To search for the user-defined field "REGION" of all types with any value:

```
qip-search -t UserFields -b All -r "REGION" -n "*"
```

Output example 5

```
West Coast|domain.com|DOMAIN/REGION
   East Coast|example.com|DOMAIN/REGION
```

Command line input example 6

- To return entries from Resource Records tab of the Object Profile:

```
qip-search -n <object name> -t "Resource Record" -b Object
```

Command line input example 7

- ·To return only CNAME entries from Resource Records tab of Object Profile (does not include Alias names from Aliases tab of Object Profile):

```
qip-search -n <object name> -t "Resource Record" -r CNAME
```

Command line input example 8

- To return object record with <alias name> in the Aliases tab of an Object Profile:

```
qip-search -n <alias name> -t Alias
```

Command line input example 9

To return User-Defined Fields associated with allocated Pools and Blocks:

```
qip-search –t UserFields –b Prov_Pool –r <Pool Field UDF Name> -n <Pool UDF
   Data> -f PoolUDF.txt
```

where **–r** is the user-defined name of the Pool or Block fields, and **–n** is the particular data contained within the Pool or Block fields for which you are searching.

Output example 9

```
Prov Pool UDF Data|ParentPool1|/Prov Pool UserDefinedField
Prov Pool UDF Data|ParentPool2|/Prov Pool UserDefinedField
Prov Pool UDF Data|ChildPool|/Prov Pool UserDefinedField
```

qip-searchacltemplates

qip-searchacltemplates returns a list of zones followed by the servers associated with each zone that are associated with ACL Templates, if they exist.

Synopsis

```
qip-searchacltemplates -tn template_name [-g loginserver]
   [-s dataserver] [-o organization] [-u username] [-p password]
   [-f output_file] [-i input_file]
```

Parameters

qip-searchacltemplates recognizes the following parameters:

-tn ***template_name***	Specifies the name of the template being retrieved.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-f ***output_file***	Specifies the file where the report is stored. The default is *STDOUT*.
-i ***input_file***	Specifies the file that contains the input data.

Output example

```
ctest.com|server1.test.com
ctest.com|server2.test3.com
secondtest.com|serverprimary.secondtest.com
secondtest.com|serversecondary.test.com
```

Note: Servers do not have to belong to the same domain to be associated with that domain. They only need to be a primary or secondary server in that domain. Each zone is listed followed by its associated servers that contain ACL Templates. If the zone, by itself, contains a template, only the zone name is listed.

qip-setacltemplate

qip-setacltemplate sets the values for a specific template.

Synopsis

```
qip-setacltemplate [-tn old_template_name] [-a]|-i
   input_file_name [-g loginserver] [-FO] [-s servername]
   [-o organization] [-u username] [-p password] [-f output_file]
```

Parameters

qip-setacltemplate recognizes the following parameters:

-a	Specifies that a template is added. If you do not specify this parameter, the template specified by the ***-tn*** parameter is replaced by the contents of the input file.
-i *input _filename*	Specifies the file that contains the input data.
-tn *old_template_name*	Specifies the name of the template being retrieved. **Note:** The ***-tn*** option must be entered before the –i option when using this CLI.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *output_file*	Specifies the file where the report is stored. The default is *STDOUT*.
-FO	Adds line feed and carriage return characters to raw XML output to improve legibility

Input file example

Refer to "qip-getacltemplate" (p. 1-57) for an example of an input file for this CLI.

qip-setadminrole

qip-setadminrole adds or modifies an Administrative role, as specified in the input data file.

Before you begin

- The field **RoleName** is mandatory and must be the first field.
- The field **RoleDescription** is optional, and if it exists must be the second field.
- The managed list can be in any order. Available managed list types are: Domain, Network, OSPF, Subnet Organization, Subnet, Object, Application, User Group, server, Address Range, Object Range, Administrative Role, and Administrator. All managed lists have four fields; Type, Name, Address, and ReadOnly. If any field is blank, use a comma "," to separate it from the next field, as shown in the "Input file content example".
- The field **Administrator** extracts the managed list from the administrator and appends it to this creating/modifying role.

Synopsis

```
qip-setadminrole [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] -f input_filename
   [-r reject_file] [-e errmsg_file] [-t]
```

Parameters

qip-setadminrole recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used for establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *input_filename*	Specifies the directory and filename of the input data. If the name contains spaces, it must be enclosed by quotation marks.
-r *reject_file*	Specifies the file where records that are not added or modified are placed.
-e *errmsg_file*	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.
-t	Sets the flag to modify the administrative role.

Command line input example

```
qip-setadminrole –u qipman –p passwd -f c:\temp\adminprof.txt
```

Input file content example

```
RoleName=testrole
RoleDescription=This is a test role
ManagedList:
Application,"Sample Application",,True
Subnet Organization,"psl_suborg1",,True
Domain,"usa.world.com",,True
Network,"net_73",73.0.0.0,True
AddressRange,14.14.14.1-14.14.14.100,Flase
AdministrativeRole,"Subnet Admin Role"
Administrator=subnet-admin
```

qip-setclientclass

Use the **qip-setclientclass** CLI to specify a client class comprising a vendor class, a device class or one or more user classes, and then attach a DHCP option template or a client class policy template to it. This client class can then be attached to a Lucent DHCP server without being assigned to a specific address scope.

Before you begin

Although **OptionTemplate** and **ClientClassPolicyTemplate** fields are optional, at least one of them must exist in the input file.

Synopsis

```
qip-setclientclass [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-r reject_file]
   [-e errmsg_file] [-n client_class] -f input_filename
```

Parameters

qip-setclientclass recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used for establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-r *reject_file*	Specifies the file where records that are not added or modified are placed.
-e *errmsg_file*	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.
-f *input_filename*	Specifies the directory and filename of the input data. If the name contains spaces, it must be enclosed by quotation marks.
-n *client class*	Specifies the client class information.

Input file example

For a description of the fields in the input file example, refer to Table 1-30.

```
ClientClassName=new client class
ClassType=USER
AttachedClass=qa support
OptionTemplate=general
```

```
ClientClassPolicyTemplate=policyTemp
```

Table 1-29 Input file field definitions

Field name/label	Value type	Description
ClientClassName	[M] String	Specifies the client class name
ClassType	[M] VENDOR, USER or DEVICE	Specifies the class type of client class
AttachedClass	[M] String	Specifies the vendor class if the ClassType is set to VENDOR or the user class(es) if the ClassType is set to USER.
OptionTemplate	[O] String	Specifies the option template attached to the client class.
ClientClassPolicyTemplate	[O] String	Specifies the client class policy template attached to the client class.

qip-setcontact

qip-setcontact allows you to add a new contact, or to modify an existing contact within the VitalQIP database. If you want to modify existing information, you can retrieve the existing contact information by using the command **qip-getcontactlst**. Contact information is set as specified in the *input_filename* (**-f** parameter).

Synopsis

qip-setcontact -f *input_filename* [**-g *loginserver***] [**-s *servername***] [**-u *username***] [**-p *password***] [**-o *organization***] [**-r *reject_file***] [**-e *errmsg_file***]

Parameters

qip-setcontact recognizes the following parameters:

-f *input_filename*	Specifies the file that contains the input data.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-r *reject_file*	Specifies the file where records that are not added or modified are placed.
-e *errmsg_file*	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.

Command line input example

```
qip-setcontact -f contactfile
```

Input file example

For a description of the fields in the input file example, refer to Table 1-30.

```
FirstName=Bob
LastName=Smith
PhoneNum=555-5555
PagerNum=555-1234
EmailAddress=bsmith@company.com
```

Table 1-30 Input file field definitions

Field	Description
FirstName	The first name of the contact.
LastName	The last name of the contact.
PhoneNum	The phone number of the contact.
PagerNum	The pager number of the contact.
EmailAddress	The e-mail address of this contact.

qip-setdnsrr

qip-setdnsrr adds or deletes resource records specified in an input file to/from the associated object, domain, or reverse zone that is passed with the **-t** option.

VitalQIP validates data to prevent CNAME conflicts in the same organization by default. If a CNAME conflict occurs, an error message displays, and you cannot add or modify the profile. You can disable the feature by setting the Validate CNAME Records policy to False. See the Object Policies and Profiles section of the *VitalQIP User Guide* for more information on the policy. When you add or modify records other than CNAME records, validation checks are made against:

- Object aliases
- Object resource records
- Domain resource records
- Reverse zone resource records

When you add or modify CNAME records, validation checks are made against:

- Object names
- Domain names
- Any Object Profile resource records
- Any Domain Profile resource records
- Any Reverse Zone resource records
- ENUM NAPTR resource records
- IPv6 node names
- Mail servers

Synopsis

```
qip-setdnsrr -t owner type -a ip_address|-n domn_name
   -f input_filename [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization] [-r reject_file]
   [-e errmsg_file]
```

Parameters

qip-setdnsrr recognizes the following parameters:

-t ***owner_type***	Specifies the owner type of the resource records. Either object, domain, or reverse zone.
-a ***ip_address***	Specifies the IP address of the object or reverse zone with which the resource records are associated.
-n ***domn_name***	Specifies the domain name with which the resource records are associated.

-f *input_filename*	Specifies the name of the file that contains the input data.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-r *reject_file*	Specifies the file where records that are not added or modified are placed.
-e *errmsg_file*	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.

Command line input examples

```
qip-setdnsrr -u qipman -p passwd -f rr.dat -t object -a 144.144.144.4
qip-setdnsrr -t domain -n lucent.com -f domainrr.dat
qip-setdnsrr -t reverse zone -a 198.200.138.0 -f rz.dat
```

Input file example

For a description of the fields in the input file example, refer to Table 1-31 following.

```
ResourceRecOwner=host.qtek.com
ResourceRecType=TXT
ResourceClass=IN
MinimumTTL=-1
ResourceRecText=this is for XXXX
ApplyToZone=1
ResourceRecOwner=host2.qtek.com
ResourceRecType=TXT
ResourceClass=IN
MinimumTTL=3600
ResourceRecText=this is for XXXX
ApplyToZone=1
ChangeFlag=1
ExternalFlag=1
Tombstoned=1
ExternalComment="Comment"
ExternalTimestamp=03/09/2005 20:37:42
```

Table 1-31 Input file field definitions

Field Name/Label	Description
Note: The following fields appear once per resource record.	
ResourceRecOwner | Owner	Owner of the resource record.
ResourceRecType | Type	Type of resource record (see Table 1-27, "Resource Record types" (p. 1-201)).
ResourceClass | Class	Class of resource record (for example, IN, CS, CH, HS).
MinimumTTL | TTL	TTL Value of the resource record. Note that a value of –1 indicates that no TTL has been specified.
ResourceRecText | Text	Data area of the resource record. The data content varies depending on the ResourceRecType.
ApplyToZone | Zone	Zone where the record should be entered: 1 | Forward | FWD = Forward zone 0 | Reverse | REV = Reverse zone Only applies to owner_type of object.
ChangeFlag | Change	1 | Add = Add new resource record 2 | Delete | Del = Delete existing resource record If the input file contains ChangeFlag= (with no value following the equal sign) then the record is skipped. If the input record does not mention ChangeFlag at all, the record is added.
External Flag	Indicates if the record is external. 1 = External; 0=Not external.
Tombstoned	Indicates if the record is tombstoned. 1=Tombstoned; 0=Not tombstoned.
ExternalComment	Traces how the record entered VitalQIP. This field is for informational purposes only and should not be used.
ExternalTimeStamp	The actual date and time at which the record was last updated by external updates. This field is for informational purposes only and should not be used.

qip-setdomainfolder

qip-setdomainfolder assigns a domain or a list of domains to a specific folder.

Synopsis

```
qip-setdomainfolder -n folderName -d domain_name -f input_filename
   [-g loginserver] [-s servername] [-u username] [-p password]
   [-o organization] [-r reject_file] [-e errmsg_file]
```

Parameters

qip-setdomainfolder recognizes the following parameters:

-n *folderName*	Specifies the name of an existing folder the domain(s) is to be assigned to.
-d *domain_name*	Specifies the fully-qualified domain name.
-f *input_filename*	This file consists of the list of domain names to be assigned to the folder.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-r *reject_file*	Specifies the file where records that are not added or modified are placed.
-e *errmsg_file*	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.

Command line input example

The file specified in the **-f** parameter contains a list of domains to assign to a folder.

```
qip-setdomainfolder -n folder1 -f domain.txt
```

Input file Content Example

```
home.com
east.com
west.com
```

qip-setlocation

qip-setlocation allows you to add a new location, or to modify an existing location within the VitalQIP database. If you want to modify existing information, you can retrieve the existing location information by using the **qip-getloctlst** command.

Synopsis

```
qip-setlocation -f input_filename [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-r reject_file] [-e errmsg_file]
```

Parameters

qip-setlocation recognizes the following parameters:

-f ***input_filename***	Specifies the file that contains the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-r ***reject_file***	Specifies the file where records that are not added or modified are placed.
-e ***errmsg_file***	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.

Command line input example

```
qip-setlocation -u qipman -p passwd -f location.lst
```

Input file content example

For a description of the fields in the input file example, refer to Table 1-32.

```
Street1=10
Street2=Valley Parkway
City=Malvern
State=PA
Zip=19355
Country=USA
```

Table 1-32 Input file field definitions

Field	Description
Street1	The Street1 field of the location.
Street2	The Street2 field of the location.
City	The City of the location.
State	The State of the location.
Zip	Zip of the location.
Country	Country of the location.

qip-setmacpools

qip-setmacpools configures the global and subnet DHCP Server MAC pools, for both the inclusion and exclusion pools. **qip-setmacpools** replaces all the existing macpools with the ones specified in the input file.

Synopsis

qip-setmacpools [**-g** ***loginserver***] [**-s** ***servername***] [**-u** ***username***] [**-p** ***password***] [**-o** ***organization***] [**-f** ***filename***] [**-r** ***reject_file***] [**-e** ***errmsg_file***]

Parameters

qip-setmacpools recognizes the following parameters:

-g ***loginserver***	Specifies the login server associated with the MAC Pools.
-s ***servername***	Specifies the server name where the MAC Pool is located.
-u ***username***	Specifies the username of the MAC Pool administrator.
-p ***password***	Specifies the password of the MAC Pool administrator.
-o ***organization***	Specifies the organization with which the MAC Pool is associated.
-f ***filename***	Specifies the filename that contains the output from this CLI.
-r ***reject_file***	Specifies the file where records that are not added or modified are placed.
-e ***errmsg_file***	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.

Input file content example

```
DHCP Server Name=testdhcp.qtek.com
E, AX25, 112233445566
E, IEEE802, 00000c1703ea
I, Token Ring, 1000023abcde
I, Arcnet, 1234123412341234

Subnet Address=200.100.10.1
I,Chaos,321321*
E,Pronet,aabbccddeeff
I,Ethernet,1233211233211234
E,Token Ring,222*
```

qip-setnmdnsserver

qip-setnmdnsserver sets the attributes of a non-managed DNS server in the specified filename (**-f** parameter).

VitalQIP validates data to prevent CNAME conflicts in the same organization by default. If a CNAME conflict occurs when adding or modifying a non-managed DNS server using the qip-setnmdnsserver CLI, an error message displays and the action on the server fails. You can disable this feature by setting the Validate CNAME Records policy to False. You can disable this feature by setting the Validate CNAME Records policy to False. See the Object Policies and Profiles section of the *VitalQIP User Guide* for more information on the policy. When you add or modify a non-managed DNS server, validation checks on the server name are made against the following:

- Object Names
- Domain Names
- Aliases
- Object Resource Records
- Domain Resource Records
- Reverse Zone Resource Records

Synopsis

```
qip-setnmdnsserver -f input_filename [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-r reject_file] [-e errmsg_file]
```

Parameters

qip-setnmdnsserver recognizes the following parameters:

-f ***input_filename***	Specifies the file that contains the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-r ***reject_file***	Specifies the file where records that are not added or modified are placed.
-e ***errmsg_file***	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.

Command line input example

```
qip-setnmdnsserver -f myinputfile
```

Input file content example

The first three fields are required in every set of input fields.

```
ServerName=qipdnsserver.lucent.com
NMServerName=nmdnsserver.lucent.com
NMServerAddress=1.2.3.4
```

Note: You must specify a full set of fields for ***either*** Domain Information or Reverse Zone Address information. You cannot specify both **DomainName** and **ReverseZoneAddress** in the same set of fields, even though the remaining fields in the set are identical.

You can specify a full set of fields for Domain Information and immediately follow it with a full set of fields for Reverse Zone Address information.

The *first* set of fields from **DomainName** to **AlsoNotify** are for Domain Information.

```
DomainName=lucent.com
RefreshTime=21600
ExpirationTime=604800
RetryPeriod=3600
MinimumTTL=86400
MaxTransferTimeIn=0
ZoneMail=email@lucent.com
CheckNames=Warning
AllowUpdate=any
AllowQuery=any
AllowTranfer=any
Notify=No
MSAllowUpdate=0
Aging=0
AgeRefreshHours=0
AgeNoRefreshHours=0
AlsoNotify=
```

The *second* set of fields from **ReverseZoneAddress** to **AlsoNotify** are for Reverse Zone Information.

```
ReverseZoneAddress= 135.114.105.0/24
RefreshTime=21600
ExpirationTime=604800
RetryPeriod=3600
MinimumTTL=86400
MaxTransferTimeIn=0
ZoneMail=email@lucent.com
CheckNames=Warning
```

```
AllowUpdate=any
AllowQuery=any
AllowTranfer=any
Notify=No
MSAllowUpdate=0
Aging=0
AgeRefreshHours=0
AgeNoRefreshHours=0
AlsoNotify=
```

qip-setobject

qip-setobject adds or modifies an object or objects. To use this CLI command effectively, first use **qip-getobjectprof** to obtain the Object Profile data; then modify it for the target object.

> **Note:** VitalQIP offers a **-t** parameter, which must be used if you only want to *modify a few attributes of the object*, rather than overwriting the entire object. Using the **-t** parameter allows you to update a portion of the record, leaving the rest of the record intact. If you omit the **-t** parameter, the fields you supply replace the *entire* record, and the resulting record contains *only* those fields. Therefore, if you use the **-t** option with this CLI command (to modify Contact or Location information), the individual detail fields for the specified Contact or Location are updated with the supplied values.

VitalQIP validates data to prevent CNAME conflicts in the same organization by default. If a CNAME conflict occurs when adding or modifying an object via the qip-setobject CLI, an error message displays and the object add or modify fails. You can disable this feature by setting the Validate CNAME Records policy to False. You can disable this feature by setting the Validate CNAME Records policy to False. See the Object Policies and Profiles section of the *VitalQIP User Guide* for more information on the policy. When you add or modify an object, validation checks on the object name are made against the following:

- Object Names
- Domain Names
- Aliases
- Object Resource Records
- Domain Resource Records
- Reverse Zone Resource Records

With this CLI command, you can modify Contact or Location information by using one of the following three methods:

- Specify the LocationID or ContactID – a number that points to a specific set of information regarding the Location or Contact to be applied to the object. For Location information, the fields are Street1, Street2, City, State, Zip, and Country. For Contact information, the fields are ContactLastName, ContactFirstName, ContactEMail, ContactPhone, and ContactPager. In the following paragraphs, these fields are referred to as the “detail fields”.

- In this method, specify the detail fields instead of specifying the LocationID or ContactID. VitalQIP finds an exact match and supplies the corresponding LocationID or ContactID. If VitalQIP does not find an exact match, it creates a new LocationID or ContactID and associates it with the object.
- In this method, you specify the ID AND any of the detail fields. If they differ from the match that VitalQIP finds, VitalQIP uses the ID field and UPDATES the detail fields with the information you provided, replacing the fields that were originally associated with the ID.

The format file (**-f** format_file) and the rejection file (**-r** *reject_file*) are also available for this CLI. Any records that are not successfully added or modified are placed in the rejection file (**-r *reject_file***). If your data file is in CSV format, specify the format file (**-f** format_file).

qip-setobject converts a data file into the correct data file format, according to the *format_file*. This structure allows users to import data from any format, as long as the example is followed.

A few things to keep in mind

– The fields can be in any order – with the exception of **ObjectAddress** – ***which must be the first field***. Any field not in the data file is represented as a NULL value.

Synopsis

```
qip-setobject [-t] [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] -d data_file [-f format_file]
   [-r reject_file] [-e errmsg_file] [-x] [-w]
```

Parameters

qip-setobject recognizes the following parameters:

-t	Sets the flag to modify the object.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-d *data_file*	Specifies the file where the information is stored. This parameter is mandatory.

-f ***format_file***	Specifies the format filename. Refer to Table 1-33 for a list of valid field names. "ObjectAddress" must exist, and must be first. Separate field names with a comma.
-r ***reject_file***	Specifies the file where records that are not added or modified are placed.
-e ***errmsg_file***	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.
-x	Allows an IP address with a status of "Selected" or "Planned Use" to be overwritten. An address may be marked "Selected" as a result of the qip-getipaddr CLI being run (for example, this CLI is called with the Quick Add function in the Web Client).
-w	Overwrites the object when that object has a duplicate name.

Example 1

```
qip-setobject -u qipman -p passwd -d setobj.txt -f setobj.fmt -r setobj.rej
```

Command line input example 1

Contents of the **-d** *setobj.txt* (data) file:

```
144.144.144.5, host1
123.123.123.123, ws2
```

Contents of the **-f** *setobj.fmt* (format) file:

`ObjectAddress, ObjectName` (other field names, as described in Table 1-33)

Example 2

```
qip-setobject -u qipman -p passwd -d objprof.txt -r setobj.rej
```

Command line input example 2

Contents of the **-d** *objprof.txt* (data) file:

```
ObjectAddress=199.200.139.28
SubnetAddress=199.200.138.0
ObjectName=hostxyz
DomainName=qtek.com
ObjectClass=PC
Application=PAYROLL
```

(other field names, as described in Table 1-33)

Input file content example

For a description of the fields in the input file example, refer to Table 1-33 following.

Note: For more information on objects, refer to "Manage Objects in a Subnet" in Chapter 4 of the *VitalQIP User's Guide*.

```
ObjectAddress=144.144.144.5
SubnetAddress=144.144.144.0
ObjectName=ws2
DomainName=quadritek.com
ObjectClass=PC
ExpiredDate=
ServerType=None
Application=
MACAddress=000a0c112233
ObjectTag=abc
LocationID=22
RoomID=
Street1=
Street2=
City=
State=
Zip=
Country=
Manufacturer=abc corp
ModelType=T1
SerialNo=008899
AssetNo=
HostID=
PurchaseDate=02/01/1998 10:00
ObjectDescription=test station
HubName=Hob21.qtek.com
SlotName=slot-21
PortNum=1
ContactID=10
ContactLastName=
ContactFirstName=
ContactEmail=
ContactPhone=
ContactPager=
RouterGroup=
DynamicConfig=A-BootP
TftpServer=
BootFileName=
HardwareType=Ethernet
Aliases=Workstatin-ws2
MailForwarders=
MailHosts=mailHost.qtek.com
HubSlots=
DNSServers=
TimeServers=
```

```
DefaultRouters=144.144.144.1
NameService=A,PTR
DynamicDNSUpdate=A,PTR,CNAME,MX
DHCPServer=dhcp1.qtek.com
DHCPOptionTemplate=general
DHCPPolicyTemplate=
LeaseTime=-1
VendorClass=
TTLtime=100
DualProtocol=
Users=
UserClasses=MS Vendor
Tombstoned=1
ExternalComment="Comment"
ExternalTimestamp=03/09/2005 20:37:42
AllowDHCPClientsModifyDynamicObjectResourceRecords=True
```

Table 1-33 Input file field definitions

Field name/label	Description	
ObjectAddress	***Required***. The IP Address of the object.	
SubnetAddress	*This field is currently ignored and may be utilized in future releases. Subnets are retrieved from the Object's Address.*	
ObjectName	***Required***. The hostname of the object. Not fully-qualified.	
DomainName	***Required***. The domain of the object. If no value is provided, the Domain Name is identified as the default domain of the subnet the Object is on. If you choose not to identify a Domain Name, enter **None** (for example, DomainName=None).	
ObjectClass	The object class of the object:	
	- Workstation - X-terminal - PC - Printer - Server - Wiring_HUB - Router - Bridge	- Terminal_Server - Switch - Legacy_System - Gateway - Test_Equipment - Undefined - Others -Any user-defined object class name
ExpiredDate	The date when a reserved object expires and is no longer reserved.	

Field name/label	Description
ServerType	Server type. Only applicable if ObjectClass is “server”: - TimeServer – If object is a time server. - TftpServer – If object is a Tftp server. - TftpServer&TimeServer – If object is a TftpServer and TimeServer
Application	The 'Primary Application' of the object.
MACAddress	The Mac Address of Object. **Note:** Exclude the colons (:) when entering a MAC address.
ObjectTag	Tag field
RoomID	Room ID field
LocationID	The Location ID. If no Location ID is specified, an attempt is made to match the record EXACTLY with the Street1, Street2, City, State, Zip and/or Country information that is supplied. If a match is not found, a new record is added without an ID. If the Location ID is specified, the information provided updates the current record using the **`-t`** option (refer to the note at the beginning of this CLI command).
Street1	Street1 field of the location.
Street2	Street2 field of the location.
City	City of the location.
State	State of the location.
Zip	Zip of the location.
Country	Country of the location.
Manufacturer	Manufacturer Name
ModelType	Model Name
SerialNo	Serial Number of Object
AssetNo	Asset Number of object
HostID	Host ID of object
PurchaseDate	Purchase Date/Time of object. Must be in “mm/dd/yyyy HH:MM” format (for example, 09/22/1998 00:00).
ObjectDescription	Object Description
HubName	Hub Name(s) assigned to object.
SlotName	Slot Name(s) assigned to object.

Field name/label	Description
PortNum	Port Numbers assigned to object.
ContactID	The Contact ID. If no Contact ID is specified, an attempt is made to match the record EXACTLY with the ContactLastName and ContactFirstName information that is supplied. If no match is found, a new record is added without an ID. If the Contact ID is specified, the information provided updates the current record using the **`-t`** option (refer to the note at the beginning of this CLI command). **Note:** If you have multiple users with the same name (first and last names) and you want to enter that same name for another individual, this field value should be 0. This tells the system to add the exact same name.
ContactLastName	Contact last name.
ContactFirstName	Contact first name.
ContactEmail	Contact email address.
ContactPhone	Contact phone number.
ContactPager	Contact pager number.
RouterGroup	Router Group of object. Only applicable if the object class is "router".
DECNetArea	DECNet Area. (Valid values are 1-63) Only applicable if Dual Protocol is set to DECNet. The DECNet Address is calculated on the input in the DECNetArea and the DECNet Node.
DECNetNode	DECNet Address. (Valid values are 1-1024) Only applicable if Dual Protocol set to DECNet. The DECNet Address is calculated on the input in the DECNetArea and the DECNet Node.
IPXNetworkNumber	IPX Network number. Only applicable if Dual Protocol is set to IPX.
IPXNode	IPX Node. Only applicable if Dual Protocol is set to IPX.

Field name/label	Description
DynamicConfig	Dynamic Configuration of Object: - Blank – Static object - None – Dynamic (none) - M-DHCP – Manual DHCP - A-DHCP – Automatic DHCP - D-DHCP – Dynamic DHCP - M-BOOTP – Manual Bootp - A-BOOTP – Automatic Bootp - Reserved - To reserve an object (the IP must be unused).
TftpServer	TFTP Server of object. Only applicable if object type is M-BOOTP.
BootFileName	Bootfile name of object. Only applicable if object type is M-BOOTP.
HardwareType	Hardware Type of object. Only applicable for M-BOOTP objects. - Ethernet - TokenRing - AX.25 - Pronet - Chaos - IEEE802 - Arcnet
Aliases*	Alias names. If multiple, separate by space (for example, www www3).
MailForwarders*	Fully-qualified host name with priority (for example, mailF.qtek.com(10)).
MailHosts*	Fully-qualified host name with priority (for example, mailH.qtek.com(100)).
Users*	User login names. To delete a user place a ‘(d)’ after the login name (for example, admin1(d).
UserClasses	Specific categories of user options. To enter one or more user classes for this object, enclose a list of users in double quotes (for example, “user1,user2,user3”).
HubSlots*	List of slot names (port numbers) if the object is a Wiring_HUB.
DNSServers*	DNS Servers attached to this object. If multiple, separate by space. Only applicable for M-BOOTP objects.

Field name/label	Description
TimeServers*	Time Servers attached to this object. If multiple, separate by space. Only applicable for M-BOOTP objects.
DefaultRouters	IP Address(es) of Default routers, separated by space if multiples.
NameService	Name Services selection: - No – Name Services checked off. - Yes – Name Services checked on. - A – Name Services checked on, A record checked on. - PTR – Name Services checked on, PTR record checked on. - A PTR – Name Services checked on, A record checked on, PTR record checked on.
DynamicDNSUpdate	Dynamic Update selection: - A – Dynamic updates for A records. - PTR – Dynamic updates for PTR records. - CNAME – Dynamic updates for CNAME records. - MX – Dynamic updates for MX records. **Note:** Multiple selections can be accomplished by specifying each component, separated by a space as a delimiter (for example, A PTR CNAME)
DHCPServer	The DHCP Server assigned to this object. Only applicable for dynamic objects.
DHCPOptionTemplate	The DHCP Option Template assigned to this object. Only applicable for dynamic objects.
DHCPPolicyTemplate	The DHCP Policy template assigned to this object. Only applicable for dynamic objects.
LeaseTime	The lease time (in seconds assigned) to this object. Only applicable for dynamic objects.
VendorClass	The Vendor Class assigned to this object. Only applicable for dynamic objects.
TTLtime	Time to live (TTL) value, in seconds. **Note:** Since 0 is a valid value, a value of –1 indicates no TTL set.
NetbiosName	NetBIOS Name. Only applicable when dual protocol is set to NETBIOS.
NetbiosDomain	NetBIOS Domain. Only applicable when dual protocol set to NETBIOS.

Field name/label	Description
DualProtocol	Dual Protocol of object: - None - DECNet
Note: The following fields appear once per resource record.	
Tombstoned	[O] 1 indicates that this is a tombstoned external record. 0 indicates that it is not a tombstoned external record.
ExternalComment	[O] text (up to 255 characters), a comment indicating the history of this external add.
ExternalTimestamp	[O] numeric. Specifies the actual date and time.
AllowDHCPClients ModifyDynamicObject ResourceRecords	True False Same as in Global Policies

Note: Fields marked with an asterisk indicate that you can add multiples on one line, separated by a space, as well as on separate lines in normal Field Name=Value format. For example, if there are multiple aliases, they can be listed as:

```
Alias=john sam lynn brian
```

or

```
Alias=john
Alias=sam
Alias=lynn
Alias=brian
```

qip-setospfprof

qip-setospfprof sets the profile of an OSPF area, based on the data provided in the input file specified in the **-f** parameter.

Synopsis

```
qip-setospfprof -f input_filename [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-r reject_file]
   [-e errmsg_file]
```

Parameters

qip-setospfprof recognizes the following parameters:

-f ***input_filename***	Specifies the file that contains the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-r ***reject_file***	Specifies the file where records that are not added or modified are placed.
-e ***errmsg_file***	Specifies filename for error messages. The default is *STDERR*.

Command line input example

```
qip-setospfprof –u qipman –p passwd –f ospfprof.txt
```

Input file content example

Note: The Subnet Address must appear before the information associated with that managed subnet. For a description of the fields in this input example and more information on objects, refer to "Define an OSPF Area" in Chapter 3 of the *VitalQIP User's Guide*.

```
OspfAreaName=OSPF1
OspfAreaID=0.0.48.58
WarningPercent=90
WarningType=1
OspfAddressRange=198.200.138.000 198.200.138.031 255.255.255.224;
```

qip-setpolicy

qip-setpolicy adds or modifies a policy entry for the enterprise server. For more information on policies, refer to Chapter 3, "Object Policies and Profiles" in the *VitalQIP User's Guide*.

Synopsis

```
qip-setpolicy -c class -n policy -d value [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-r reject_file] [-e errmsg_file] [-a new_data_type]
```

Parameters

qip-setpolicy recognizes the following parameters:

-c ***class***	Specifies the class of policy you want to modify or add (for example, General).
-n ***policy***	Specifies the policy you want to add or modify (for example, PING_ATTEMPTS and ALWAYS_APPEND_ROUTER).
-d ***value***	Specifies the value you want to assign to the policy. Alphabetic values must be in uppercase letters.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-r ***reject_file***	Specifies the file where records that were not added or modified are placed.
-e ***errmsg_file***	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.
-a ***new_data_type***	Specifies the new data type. Values are as follows:

boolean True/False	time_of_day_list
ip_address	numeric_list
ip_address_list	Delete
ip_address_mask_list	numeric_signed
ip_address_pair_list	password
ip_mask	text
multi-lined_text	text_list
numeric	time_interval

Command line input example

- To set up the enterprise server to send DNS updates directly to the DNS server:

```
qip-setpolicy –s QIPSYBASE –u qipman –p passwd -c Dynamic DNS –n
   Use_DNS_Update_Service
   -d FALSE
```

- To add a new policy to the Dynamic DNS class:

```
qip-setpolicy -c “Dynamic DNS” -n NewPolicy -a Text -d “this is my
   new policy”
```

qip-setreclaimschedule

qip-setreclaimschedule sets or resets the subnet reclaim schedule using the parameters given in the input file.

Synopsis

```
qip-setreclaminschedule -f input_filename [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-b]
```

Parameters

qip-setreclaimschedule recognizes the following parameters:

-f ***input_filename***	Specifies the input file name, which contains the schedule information
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection
-o ***organization***	Specifies the VitalQIP organization (corporation) name
-p ***password***	Specifies the password for the associated administrator account
-b	If a reclaim schedule for a given subnet exists, the user needs to use "**-b**" option to overwrite the existing schedule. When **-b** is used, the existing scheduled reclaim is aborted, a report generated and emailed to the administrator who setup the original schedule.

Input example 1

```
SubnetAddress=144.144.1.0
Reclaim Type=Report Only
Reclaim Email Address=root@test.com
Data Collection(Ping) Schedule=By Day
Time of Day=36000
Time of Day=54000
Total Number Of Days=5
```

Command line input example 2

```
SubnetAddress=144.144.1.0
Reclaim Type=Report Only
Reclaim Email Address=root@test.com
Data Collection(Ping) Schedule=By Interval
Time Interval=1 day 20 hours
```

```
Total Number Of Times=5
SubnetAddress=144.144.0.0
Reclaim Type=Report Only
Reclaim Email Address=root@test.com
Data Collection(Ping) Schedule=By Day
Time of Day=10:00
Total Number Of Days=5
```

Output example

The "Automatic Reclaim Report" is generated only for an existing scheduled reclaim when **-b** option is used. It is generated using the objects statistics accumulated thus far, with a note stating the reason for the abort, and sent to the email address of the administrator who scheduled the reclaim. The report, which has the same format as the report generated by **qip-reclaim** CLI, is also placed in *$QIPHOME/report* named with the date/time stamp and subnet address for uniqueness.

```
  <<<<< AUTOMATIC RECLAIM REPORT >>>>>
  <<<<< Please Note: >>>>>
Scheduled Reclaim Aborted due to Schedule Reset by : qipman ( )

Report Date:  11/19/2003 14:45Start Date:  11/19/2003 14:18
Collection Frequency: Collect: Every Day At (09:03 ) Total: 1 Days

                                          Object   Object   #    #
   #    Reclaim  Last Time
Address              Name                 Class     Status  Try  In
   DNS  Reachable Status Reachable
-------------------------------------------------------------------------------
   ---------------------

10.200.90.1   rtp000006rts.seg4.qa.quadritek.com     Router       Static
   0      0        0
10.200.90.2                                                     Unused
   0      0        0
10.200.90.3                                                     Unused
   0      0        0
10.200.90.4                                                     Unused
   0      0        0
10.200.90.5   wsp000074WSS.seg4.qa.quadritek.com    Workstation   Static
   0      0        0
10.200.90.6                                                     Unused
   0      0        0
10.200.90.7                                                     Unused
   0      0        0
```

qip-setsnorgprof

qip-setsnorgprof establishes the profile of a subnet organization based on the data provided in the input file specified in the **-f** parameter.

Synopsis

```
qip-setsnorgprof -f input_filename [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-r reject_file] [-e errmsg_file]
```

Parameters

qip-setsnorgprof recognizes the following parameters:

-f ***input_filename***	Specifies the file that contains the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-r ***reject_file***	Specifies the file where records that are not added or modified are placed.
-e ***errmsg_file***	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.

Command line input example

```
qip-setsnorgprof -u qipman -p passwd -f subnetorg.txt
```

Input file content example

For a description of the fields in the input file example, refer to Table 1-34.

```
SubnetOrgName=snorg1
WarningPercent=95
WarningType=1
AssignGap=Workstation,1
AssignGap=PC,2
SubnetAddress(net,mask,addr)=144.144.0.0,255.255.255.0,144.144.1.
   0
SubnetAddress(net,mask,addr)=144.144.0.0,255.255.255.0,144.144.2.
   0
```

```
W2KSiteName=site1
UseSubnetOrgName=false
AssocControllers=dc1.lucent.com
DhcpServer=dhcp1.lucent.com
DhcpOptionTemplate=general
```

Table 1-34 Input file field definition

Field name/label	Description
SubnetOrgName	The Subnet Organization name.
WarningPercent	The Warning for the percentage of Managed Addresses, (for example, 10 equals 10 percent).
WarningType	Warning Type: 0 – No warning 1 – Email only 2 – Visual only 3 – Both email and visual warning
AssignGap	The object class of the GAP object defined. The relative offset. Note that positive number (>0) indicates "from the beginning" of the subnet. A negative number indicates "from the end" of the subnet.
SubnetAddress(net,mask,addr)	The subnet address, network address, and subnet mask of the subnet.
W2KSiteName	The site name of the Windows 2000 site. If blank, the subnet organization name is used. (Optional)
AssocControllers	List of Windows 2000 controllers.
DHCPServer	The default DHCP server for this subnet.
DHCPOptionTemplate	The default DHCP Option Template for this subnet.
UseSubnetOrgName	Specifies if the subnet organization name is used.

qip-setsubnet

qip-setsubnet adds or modifies Subnet Profiles, based on the data provided in the input file specified in the **-f** parameter. To effectively use this CLI command, first use **qip-getsubnetprof** to obtain the Subnet Profile data; then modify it as input. The fields can be in any order except for the first field, which must be the subnet address.

> **Note:** VitalQIP offers a **-t** parameter, which must be used if you only want to *modify a few attributes of the subnet*, rather than overwriting the entire profile. Using the **-t** parameter allows you to update a portion of the record, leaving the rest of the record intact. If you omit the **-t** parameter, the fields you supply replaces the *entire* record, and the resulting record contains *only* those fields. Therefore, if you use the **-t** option with this CLI command (to modify Contact or Location information), the individual detail fields for the specified Contact or Location are updated with the supplied values.

With this CLI command, you can modify Contact and Location information by using one of the following three methods:

- Specify the **LocationID** or **ContactID** – a number that points to a specific set of information regarding the Location or Contact to be applied to the subnet. For Location information, the fields are **Street1**, **Street2**, **City**, **State**, **Zip**, and **Country**. For Contact information, the fields are **ContactLastName**, **ContactFirstName**, **ContactEMail**, **ContactPhone**, and **ContactPager**. In the following paragraphs, these fields are referred to as the “detail fields”.
- In this method, *instead of* specifying the **LocationID** or **ContactID**, you specify the detail fields. VitalQIP finds an exact match and supply the corresponding **LocationID** or **ContactID**.
- In this method, you specify the ID *AND* the detail fields. If they differ from the match that VitalQIP finds, VitalQIP uses only the ID field; it ignores your specified detail fields.

Before you begin

- The fields can be in any order with the exception of **SubnetAddress** - ***which must be the first line*.** Any field not in the data file is represented as a NULL value.
- To add a new subnet, the fields **NetworkAddress=** and **SubnetMask=** are mandatory. The network must already have been defined in the system

Synopsis

```
qip-setsubnet -f input_filename [-t] [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-r reject_file] [-e errmsg_file] [-t]
```

Parameters

qip-setsubnet recognizes the following parameters:

-f ***input_filename***	Specifies the file that contains the input data.
-t	Sets the flag to modify the subnet profile. The fields listed in the input file are changed to the specified values, and all other fields will be left unchanged.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-r ***reject_file***	Specifies the file where records that are not added or modified are placed.
-e ***errmsg_file***	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.

Command line input example

```
qip-setsubnet -u qipman -p passwd -f subnet.txt
```

Input File Content Example

For a description of the fields in the input file example, refer to Table 1-35.

```
SubnetAddress=144.144.144.0
SubnetName=SN144_144
SubnetMask=255.255.255.0
NetworkAddress=144.144.0.0
LocationID=2
ContactID=2
ContactLastName=pal
ContactFirstName=pal
ContactEmail=pal@qtek.com
ContactPhone=111-0033
ContactPager=1-800-111-3300
Application=
Domain=quadritek.com
TftpServer=
ShowUsage=N
CheckUsage=N
SubnetDescription=
```

```
SharedNetwork=
HardwareType=ethernet
WarningType=Visual
WarningPercent=80
DNSServers=pallas.quadritek.com
TimeServers=
DefaultRouters=
DHCPServer=dhcp.quadritek.com
DHCPOptionTemplate=general
DHCPPolicyTemplate=
PrimaryInterface=No
AllowDHCPClientsModifyDynamicObjectResourceRecords=True
```

Table 1-35 Input file field definition

Field	Description
SubnetAddress	***Required***. The subnet address. Must be the first line.
SubnetName	The subnet name.
SubnetMask	***Required when adding a subnet***. Subnet mask of the subnet.
NetworkAddress	The network address of this subnet. The network must already be defined within VItalQIP. Required when adding a subnet.
LocationID	The Location ID can be found by running **qip-getloclst**. If no Location ID is specified, the record is attempted to be matched EXACTLY with the Street1, Street2, City, State, Zip and/or Country information that is supplied. If it is not matched, a new record is added without an ID. If the Location ID is specified, the information provided updates the current record using the **`-t`** option (refer to the note at the beginning of this CLI command).
Street1	Street1 field of the location.
Street2	Street2 field of the location.
City	City of the location.
State	State of the location.
Zip	Zip of the location.
Country	Country of the location.
ContactID	The Contact ID can be found by running **qip-getcontactlst**. If no Contact ID is specified, the record is attempted to be matched EXACTLY with the ContactLastName and ContactFirstName information that is supplied. If it is not matched, a new record is added without an ID. If the Contact ID is specified, the information provided updates the current record using the **`-t`** option (refer to the note at the beginning of this CLI command).
ContactLastName	Contact last name.
ContactFirstName	Contact first name.
ContactEmail	Contact email address.
ContactPhone	Contact phone number.
ContactPager	Contact pager number.
Application	Primary Application assigned to this subnet.

Field	Description
Domain	Domain(s) assigned separated by spaces. If multiple domains are listed, the first one will be the default. All valid domains for this subnet should be in the list. Objects in this subnet will not be modified even if they are in different domains.
TftpServer	Default Tftp server assigned.
ShowUsage	Show Used Only Flag: 0 – No 1 – Yes
CheckUsage	Check before assign: Y=Ping N=None
SubnetDescription	Subnet "comment" text area.
SharedNetwork	The Shared Network name.
HardwareType	Default hardware type: - Ethernet - TokenRing - AX.25 - Pronet - Chaos - IEEE802 - Arcnet
WarningType	Warning Type: 0 – No warning 1 – Email only 2 – Visual only 3 – Both email and visual warning
WarningPercent	Threshold percentage before alarm is issued. Percent range is 0 to 99.
DNSServers	DNS Servers for this subnet, separated by spaces.
TimeServers	Time Servers for this subnet, separated by spaces.
DefaultRouters	Default Routers for this subnet, separated by spaces.
DHCPServer	The default DHCP server associated with this subnet.
DHCPOptionTemplate	The default DHCP Option Template associated with this subnet.
DHCPPolicyTemplate	The default DHCP Policy Template associated with this subnet.
PrimaryInterface	Primary Interface – Yes / No

Field	Description
AllowDHCPClients ModifyDynamicObjectResourceRecords	True False Same as in Global Policies

qip-setudf

qip-setudf sets the value of a pre-existing user-defined field associated with the specified class (User, Domain, Reverse_Zone, Organization, Object or Subnet).

Synopsis

```
qip-setudf [-g loginserver] [-s servername] [-o organization]
   [-u username] [-p password] -c class_name -a ip_address
   [-n owner_name] -f field_name [-d field_value][-r reject_file]
   [-e errmsg_file]
```

Parameters

qip-setudf recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-c *class_name*	Specifies the class with which the user field is associated. Refer to Table 1-36 following.
-a *ip_address*	Specifies the IP address of the user-defined field class; reverse_zone, object, or subnet. If the user-defined field class is Reverse_Zone, the address is xxx.xxx.xxx.xxx/mask.
-n *owner_name*	Specifies the IP address of the class User, Domain or Organization. If the user-defined field class is User, then it requires the user's Login ID, as defined in the User Profile. If this parameter and the **-d** parameter are not specified, then a new user-defined field is created.
-f *field_name*	Specifies the field name of the value to be updated or added.
-d *field_value*	Specifies the field value of the field to be updated or added. If this parameter and the **-n** parameter are not specified, then a new user-defined field is created.
-r *reject_file*	Specifies the file where records that are not added or modified are placed.
-e *errmsg_file*	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.

Table 1-36 User field classes

User field classes	Notes
User	Used for User Defined Fields for a User Profile.
Domain	Used for User Defined Fields for a Domain.
Reverse_Zone	Used for User Defined Fields for a Reverse Zone.
Organization	Used for User Defined Fields for an Organization.
Object	Used for User Defined Fields for an Object Profile.
Subnet	Used for User Defined Fields for a Subnet Profile.

Command line input examples

```
qip-setudf -u qipman -p passwd -c object -a 100.34.12.12 -f "field desc" -d
   "this is bob's pc"
qip-setudf -o "Test Org" -c Reverse_Zone -a 10.0.0.0/24 -f R21 -d Test
qip-setudf -n "USA Org" -c Organization -f "Managed by" -d "John Smith"
```

qip-setuser

qip-setuser adds or modifies a VitalQIP User Profile. The information is taken from the input file specified on the command line.

Before you begin

- If the Location ID for the user location is known, the Street1, Street2, City, State, Zip, and Country can be replaced with **loc_id=*<location_ID>*** (for example, loc_id=10).
- If multiple user-defined fields are to be entered, UserDefinedFields=***<value>*** must be specified for each user-defined field.
- Multiple Group Information, Default Subnets, and Managed Range can be entered on single line, which is separated by a comma for each entry. It can also be deleted by using the following format example:

```
ManagedRange=144.144.144.2(D),144.144.144.1
```

If the (D) value is not specified, the value is assumed to be an addition to the User Profile.

Synopsis

qip-setuser [**-g** ***loginserver***] [**-s** ***servername***] [**-u** ***username***] [**-p** ***password***] [**-o** ***organization***] **-f** ***input_filename*** [**-r** ***reject_file***] [**-e** ***errmsg_file***] [**-t**]

Parameters

qip-setuser recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-r ***reject_file***	Specifies the file where records that are not added or modified are placed.
-e ***errmsg_file***	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.
-f ***input_filename***	Specifies the filename that contains the user profile data.
-t	If specified, updates the entry with the supplied data. Otherwise, it replaces the entry.

Command line input example

```
qip-setuser -u qipman -p password -f usr_prof.dat
```

Input file content example

Note: For a description of the fields shown below, refer to the "Define VitalQIP users" section in Chapter 4 of the *VitalQIP User's Guide*.

```
LoginID=pal3
LastName=test
FirstName=pal3
Phone=610-292-1312
E-mailAddress=pal3@qtek.com
Password=333333
PIN=
Description=another user with different id
Street1=5
Street2=Whitney Ave
City=New Haven
State=Ct
Zip=02488
Country=usa
ActivationStatus=1
GroupInformation="isp_qtek2","isp_qtek"
DefaultSubnets=144.144.144.0
ManagedRange=144.144.144.3,144.144.144.2
UserDefinedFields=USER1, user1 udf value for pal3
```

qip-setzonednsoptions

qip-setzonednsoptions supports multiple zone selection (domain and reverse zones) and specifies a set of DNS options to apply to all those zones. **Qip-setzonednsoptions** requires two input files: One that contains a list of Forward/Reverse (**-z**) zones, and one that contains a list of data to be changed (**-f**).

Before you begin

- If the user-defined field name in the input field can be found in both the domain and reverse zone, it is implemented in both.
- The input fields can be in any order.
- Any field not in the input file is not changed.
- The value format for those fields that allow <Use List> is as follows.
  - <ACL Templates=TemplateName> means the TemplateName is added to the ACL Template list if possible (for example, **AllowQueryBIND8X=ACL Templates=ACLTwo**)
  - <other=ValueName> means the ValueName is added to the "other" list if possible (for example, **AllowQueryBIND8X=other=extraVal**)
  - <ACL Templates=TemplateNameD> means the TemplateName is deleted from the original list if possible (for example, **AllowQueryBIND8X=ACL Templates=ACLTwoD**)
  - * means ALL templates are deleted from the list if possible (for example, **AllowQueryBIND9X=ACL Templates=*** or **AllowQueryBIND9X=other=***)
- The value format for those fields that allow <Use List> for address lists is as follows.
  - <address> means the address is added into the original list if possible (for example, **AllowTransferW2K=198.200.138.123**)
  - <addressD> means the address is deleted from the original list if possible (for example, **NotifyW2K=170.200.12.162D**)
  - * means ALL addresses are deleted from the list if possible (for example, **NotifyW2K=***)
- For those notify fields (that is, NotifyBIND8X, NotifyBIND9X, NotifyLUCENT3,NotifyLUCENT4) that allow "Yes" and/or "Explicit" to be specified and, if you do so, you are also allowed to specify the sub-field "also-notify", the value format is as follows:
  - Example 1: For Lucent DNS 4.X Options, to set the "notify" to "Yes" and add "1.2.3.4, 5.6.7.8" to "also-notify" address list:

NotifyLUCENT4=Yes, 1.2.3.4, 5.6.7.8

– Example 2: For Lucent DNS 4.X Options, to set the “notify” to “Explicit” and “also-notify” address list to empty:

```
NotifyLUCENT4=Explicit, *
```

– Example 3: For Lucent DNS 4.X Options, to set the “notify” to “Yes” and delete “1.2.3.4” from the address list:

```
NotifyLUCENT4=Yes, 1.2.3.4D
```

Synopsis

qip-setzonednsoptions [**-g** *loginserver*] [**-s** *servername*] [**-u** *username*] [**-p** *password*] [**-o** *organization*] [**-r** *reject_file*] [**-e** *errmsg_file*] **-f** *input_filename* **-z** ***zone_filename***

Parameters

qip-setzonednsoptions recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server’s IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-r ***reject_file***	Specifies the file where records that are not added or modified are placed.
-e ***errmsg_file***	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.
-f ***input_filename***	Specifies the filename that contains the DNS options.
-z ***zone_filename***	Specifies the file containing the list of domain names and/or reverse zone addresses.

Zone file content example (-z)

```
a.com
b.com
c.com
170.20.0.0/16/*the reverse zone format: address/masklength*/
180.20.0.0/17
180.20.0.128/17
```

Input file content example (-f)

```
ZoneMail=admin@company.com
RefreshTime=21600
ExpireTime=604800
MinimumTime=86400
RetryTime=3600
NegativeCacheTTL=86400
AllowUpdateW2K=Yes
AgingW2K=True
RefreshIntervalW2K=1
noRefreshIntervalW2K=1
AllowTransferW2K=200.0.0.100D
NotifyW2K=200.0.0.101,198.0.0.198
ZoneOptionW2K=/temp/filename
CheckNamesBIND8X=Fail
AllowUpdateBIND8X=other=extraVal
AllowQueryBIND8X=ACL Templates=ACLTwo,ACLone
AllowTransferB
IND8X=ACL Templates=ACLTwo NotifyBIND8X=
   Yes,10.20.30.40,198.200.138.4
ZoneBlockBIND8X=/tmp/filename
AllowNotifyBIND9X=other=extravalue,ACL Templates=ACLTwo
AllowQueryBIND9X=ACL Templates=*,other=*
AllowTransferBIND9X=None
AllowUpdateBIND9X=None
NotifyBIND9X= Use Server Value
ZoneBlockBIND9X=/temp/filename
ZoneEdupLUCENT3=False
AllowQueryLUCENT3=None
AllowTransferLUCENT3=None
AllowUpdateLUCENT3=None
CheckNamesLUCENT3=Fail
NotifyLUCENT3=Use Server Value
ZoneBlockLUCENT3=/temp/filename
ZoneEdupLUCENT4=False
AllowNotifyLUCENT4=None
AllowQueryLUCENT4=None
AllowTransferLUCENT4=None
AllowUpdateLUC
ENT4=None NotifyLUCENT4=Explicit,1.2.3.4, 198.200.138.10
ZoneBlockLUCENT4=/temp/filenameI
```

Table 1-37 Input file fields

Field	Description
ZoneMail	Any errors posted by DNS are sent to this email address.
ExpireTime	In seconds. When the expire time is reached, the secondary server stops handling information about the data because the data is too old to be useful.
RefreshTime	In seconds. Dictates how often the secondary server should verify its data.
MinimumTime	In seconds. Defines the time interval for other servers to cache all resource records in the database file.
RetryTime	In seconds. Dictates the interval for attempting to refresh in the event that the primary server is unavailable.
AllowUpdateW2K	Values are yes, no, or Use Server Value.
AgingW2K	Values are true, false, or Use Server Value.
RefreshIntervalW2k	In hours.
NoRefreshIntervalW2K	In hours.
AllowTransferW2K	Values are any, none, NameServersOnly, <Use List>, Use Server Value.
NotifyW2K	Values are yes, no, <Use List>, Use Server Value.
ZoneOptionW2K	The filename of the Windows 2000 zone option. The contents of the file is appended to the existing options.
ZoneBlockBIND8x	The contents of the file is appended to the existing options.
CheckNamesBIND8X	Values are warn, fail, ignore, Use Server Value.
AllowUpdateBIND8X	Values are any, none, localhost, localnets, <Use List>, Use Server Value.
AllowQueryBIND8X	Values are any, none, localhost, localnets, <Use List>, Use Server Value.
AllowTransferBIND8X	Values are any, none, localhost, localnets, <Use List>, Use Server Value.
NotifyBIND8X	Values are yes, no, Use Server Value.
ZoneBlockBIND8X	The filename of the zone block of *named.conf*. The contents of the file are appended to the existing options.

Field	Description
<User-Defined Field Names>	Specify the valid user-defined field name(s). The real field name(s) here are those defined in the Global Policies|User-Defined Fields with the type "Domain" and/or "Reverse Zone".
NegativeCacheTTL	The amount of time to cache negative responses (entries that do not exist); a typical value is 600.
AllowNotifyBIND9X	Values are Any, None, localhosts, localnets, <Use List>, Use Server Value
AllowQueryBIND9X	Values are Any, None, localhosts, localnets, <Use List>, Use Server Value
AllowTransferBIND9X	Values are Any, None, localhosts, localnets, <Use List>, Use Server Value
AllowUpdateBIND9X	Values are Any, None, localhosts, localnets, <Use List>, Use Server Value
NotifyBIND9X	Values are No, Yes, Explicit, Use Server Value
ZoneBlockBIND9X	The filename of the zone block of the *named.conf*. The contents of the file are appended to the existing options.
ZoneEdupLUCENT3	Values are True, False
AllowQueryLUCENT3	Values are Any, None, localhosts, localnets, <Use List>, Use Server Value
AllowTransferLUCENT3	Values are Any, None, localhosts, localnets, <Use List>, Use Server Value
AllowUpdateLUCENT3	Values are Any, None, localhosts, localnets, <Use List>, Use Server Value
CheckNamesLUCENT3	Values are Warn, Fail, Ignore, Use Server Value
NotifyLUCENT3	Values are No, Yes, Use Server Value
ZoneBlockLUCENT3	The filename of the zone block of the *named.conf*. The contents of the file are appended to the existing options.
ZoneEdupLUCENT4	Values are True, False
AllowNotifyLUCENT4	Values are Any, None, localhosts, localnets, <Use List>, Use Server Value
AllowQueryLUCENT4	Values are Any, None, localhosts, localnets, <Use List>, Use Server Value

Field	Description
AllowTransferLUCENT4	Values are Any, None, localhosts, localnets, <Use List>, Use Server Value
AllowUpdateLUCENT4	Values are Any, None, localhosts, localnets, <Use List>, Use Server Value
NotifyLUCENT4	Values are No, Yes, Explicit, Use Server Value
ZoneBlockLUCENT4	The filename of the zone block of the *named.conf*. The contents of the file are appended to the existing options.

qip-setzoneext

qip-setzoneext resets the content of the zone extension associated with a domain or domains, a DNS server or DNS servers, a reverse zone or reverse zones. To use this CLI command effectively, first use **qip-getzoneext** to obtain the zone extension data, then modify it for the target object.

Synopsis

```
qip-setzoneext -a domain|reverse|server [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
   [-r reject_file] [-e errmsg_file] -f filename [-t]
```

Parameters

qip-setzoneext recognizes the following parameters:

-a domain|reverse|server	Specifies the zone extensions target (domain, reverse zone, or server).
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-r *reject_file*	Specifies the file where records that are not added or modified are placed.
-e *errmsg_file*	Specifies the filename where error messages are written, if they occur. The default is STDERR.
-f *filename*	Specifies the filename of the input data.
-t	Specifies that the zone extensions in the input file are appended to the existing zone extensions. Omitting this flag causes existing zone extensions to be removed and replaced with the zone extensions in the input file.

Command line input examples

Note: For a description of the fields in this input example and more information on Domain Extensions, refer to "Define Domains" in Chapter 3 of the *VitalQIP User's Guide*.

To set Domain Extensions, execute the following:

qip-setzoneext -u qipman -p passwd -a domain -f extfile.txt

Contents of extfile.txt

```
DomainName=qtek.com
OptionType=Extensions
ParameterName=Prefix of zone db file
STTL 6400
ParameterName=Postfix of zone db file
candybar IN A 19.20.21.22
snickers IN CNAME candybar.quadritek.com
candybar IN HINFO "VAX" "UNIX"
quadritek.com. IN MX 10 candybar.quadritek.com.
OptionType=BIND-8.X Options
ParameterName=zone block of named.conf
// extra zone options for BIND 8.X server
qddns {
wins-servers {1,2,3,4:1.2.3.5;}
wins-ttl 0;
};
```

To set the DNS server extensions, execute the following:

qip-setzoneext -u qipman -p passwd -a server -f extfile.txt

Contents of extfile.txt

```
ServerName=dnsserver.qtek.com
ParameterName= Corporate Extension
include "allowquery.conf"
include "blackhoe.conf"
ParameterName=db.cache file extension
Include "test.conf"
```

To set the Reverse Zone Extensions, execute the following:

qip-setzoneext -u qipman -p passwd -a reverse -f extfile.txt

Contents of extfile.txt

```
ReverseZone=170.200.0.0/16
DomainName=qtek.com
OptionType=Extensions
```

```
ParameterName=Prefix of zone db file
STTL 4600
ParameterName=Postfix of zone db file
68 IN PTR madeup.quadritek.com.
69 IN PTR madeup2.quadritek.com.
70 IN PTR madeup3.quadritek.com.
71 IN PTR madeup4.quadritek.com.
OptionType=BIND-8.X Options
ParameterName=zone block of named.conf
qddns {
wins-servers {1,2,3,4:1.2.3.5;}
wins-ttl 0;
};
OptionType=BIND-9.X Options
ParameterName=zone block of named.conf
// extra zone options for BIND 9.X server
forward only;
zone-statistics yes;
OptionType=LUCENT DNS 3.X Options
ParameterName=zone block of named.conf
// extra zone options for LUCENT DNS 3.X server
// …
OptionType=LUCENT DNS 4.X Options
ParameterName=zone block of named.conf
// extra zone options for LUCENT DNS 4.X Server
// …
OptionType=WINDOWS 2000 DNS Options
ParameterName=zone-options
// extra zone options for WINDOWS 2000 DNS Server
// …
```

qip-setzoneprof

qip-setzoneprof assigns the Reverse Zone Profile or Domain Profile information. The file specified in the **-f** parameter contains the input information.

VitalQIP validates data to prevent CNAME conflicts in the same organization by default. If a CNAME conflict occurs, an error message displays, and you cannot add or modify the profile. You can disable the feature by setting the Validate CNAME Records policy to False. See the Object Policies and Profiles section of the *VitalQIP User Guide* for more information on the policy. When you add or modify records other than CNAME records, validation checks are made against:

- Object aliases
- Object resource records
- Domain resource records
- Reverse zone resource records

When you add or modify CNAME records, validation checks are made against:

- Object names
- Domain names
- Any Object Profile resource records
- Any Domain Profile resource records
- Any Reverse Zone resource records
- ENUM NAPTR resource records
- IPv6 node names
- Mail servers

Synopsis

```
qip-setzoneprof [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] [-r reject_file]
   [-e errmsg_file] -f filename [-t] [-T Domain|Zone}
```

Parameters

qip-setzoneprof recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.

-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-r *reject_file*	Specifies the file where records that are not added or modified are placed.
-e *errmsg_file*	Specifies the filename where error messages are written, if they occur. The default is *STDERR*.
-f *filename*	Specifies the file containing the input data.
-t	Associates the given forward zone with all assigned subnets in a specific organization.
-T domain|zone	Specifies whether this server is a domain server or a zone server. Domain specifies a forward zone, Zone specifies a reverse zone. If not specified, the zone type is determined by the zone name. Any name in the format of IPV4/length (xxx.xxx.xxx.xxx/xx) is interpreted as a reverse zone.

Command line input example

```
qip-setzoneprof –u qipman –p passwd -f domain.txt
```

Input File Content Example

Note: For description of the fields in this input example and details on Domains, refer to “Define Domains” in Chapter 3, of the *VitalQIP User’s Guide*.

Domain input example

```
Zone=qtek.com
dnsServers=dns1.qtek.com P 0,dns2.qtek1.com S 0,dns2k.qtek2k.com S
   0
RefreshTime=21600
ExpirationTime=604800
RetryPeriod=3600
MinimumTTL=86400
ZoneMail=qtek@qtek.com
Extensions
Prefix of zone db file= My zone prefix comments
Postfix of zone db file= My zone postfix comments
BIND-8.X Options
allow-query=Any
allow-transfer=Any
allow-update=Any
check-names=Warn
notify=No
zone block of named.conf=
BIND-9.X Options
```

```
allow-notify=Use Server Value
allow-query=Any
allow-transfer=Any
allow-update=Any
notify=No
zone block of named.conf=
LUCENT DNS 3.X Options
Import External Updates=False
allow-query=Any
allow-transfer=Any
allow-update=Any
check-names=Warn
notify=No
zone block of named.conf=
LUCENT DNS 4.X Options
Import External Updates=True
A (Host IPV4)=False
AAAA (Host IPV6)=True
CNAME (Canonical Name)=False
PTR (Pointer)=True
SRV (Server Resource Record)=False
TXT (Text)=True
allow-notify=Use List
ACL Templates=WestACLTemplate;SouthACLTemplate
other=ThisIsOther;MoreOther99;10.200.60.25
allow-query=Any
allow-transfer=Use Server Value
allow-update=localnets
notify=No
zone block of named.conf= my zone block of named.conf comments
WINDOWS 2000 DNS Options
aging=False
allow-transfer=Any
allow-update=No
no-refresh-interval=0
notify=No
refresh-interval=0
zone-options=
```

Reverse Zone input example

```
Zone=200.200.200.0/24
ParentAddress=
NetworkAddress=
dnsServers=,revdns.rev.com P 0
RefreshTime=21600
```

```
ExpirationTime=604800
RetryPeriod=3600
MinimumTTL=86400
ZoneMail=rev1@rev.com
Extensions
Prefix of zone db file= My zone prefix comments
Postfix of zone db file= My zone postfix comments
BIND-8.X Options
allow-query=Any
allow-transfer=Any
allow-update=Any
check-names=Warn
notify=No
zone block of named.conf=
BIND-9.X Options
allow-notify=Use Server Value
allow-query=Any
allow-transfer=Any
allow-update=Any
notify=No
zone block of named.conf=
LUCENT DNS 3.X Options
Import External Updates=False
allow-query=Any
allow-transfer=Any
allow-update=Any
check-names=Warn
notify=No
zone block of named.conf=
LUCENT DNS 4.X Options
Import External Updates=True
A (Host IPV4)=False
AAAA (Host IPV6)=True
CNAME (Canonical Name)=False
PTR (Pointer)=True
SRV (Server Resource Record)=False
TXT (Text)=True
allow-notify=Use List
ACL Templates=WestACLTemplate;SouthACLTemplate
other=ThisIsOther;MoreOther99;10.200.60.25
allow-query=Any
allow-transfer=Use Server Value
allow-update=localnets
notify=No
zone block of named.conf= my zone block of named.conf comments
```

```
WINDOWS 2000 DNS Options
aging=False
allow-transfer=Any
allow-update=No
no-refresh-interval=0
notify=No
refresh-interval=
```

Note: DNS servers should be listed as FQDN followed by a “P” or “S” for Primary or Secondary, with each server name/type separated by a comma. Additionally, you must specify a 0 (no) or 1 (yes) to indicate sending secure updates. The server must be a secure server in order to send secure updates.

qip-setzoneserveroptions

This CLI allows you to set the zone server options for an existing DNS zone server.

Synopsis

```
qip-setzoneserveroptions [-g loginserver] -s servername
   [-u username] [-p password] [-o organization] -f <input_file>
   [-t domain|zome] [-r <rejectfile>] [-e errmsgfile]
```

Parameters

qip-setzoneserveroptions recognizes the following parameters:

-s *servername*	Specifies the DNS zone server you are modifying.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *input file*	Specifies the file that contains input for this CLI.
-t domain|zone	Specifies whether this server is a domain server or a zone server. Domain specifies a forward zone, Zone specifies a reverse zone. If not specified, the zone type is determined by the zone name. Any name in the format of IPV4/length (xxx.xxx.xxx.xxx/xx) is interpreted as a reverse zone.
-r *rejectfile*	Specifies the file where rejected data from this CLI are written. If not specified, errors are written to standard output.
-e *errmsgfile*	Specifies the file where errors from this CLI are written. If not specified, output is written to standard output.

Command line input example

```
qip-setzoneserveroptions -g 10.25.230.12 -s eastcoast1
   -f serveropts -t domain
```

Input file content example

The format of an input file is as follows:

```
Zone_name
Primary=primary_server_name
Secondary=server_name_if_the_server_is_a_secondary_server
Option_name=option_values
```

The following is an example of a input file for the server "dns1.abc.com" on zone "abc.com"

```
abc.com
Primary=dns1.abc.com
Secondary=
LUCENT DNS 4.X Options=Use Zone Value
Stealth Server=True
Override Server Name in SOA=True
Override Server Name=mag.abc.com
```

qip-sitegen

"Sites" are a method of further defining subnet groups, and are equivalent to subnet organizations in Windows 2000. **qip-sitegen** is used to push new and changed configurations to Windows 2000 Domain Controller.

Synopsis

```
qip-sitegen -n dc_svr [-g loginserver] [-s servername] [-u
   username] [-p password] [-o organization] [-d directory] [-l]
   [-z] [-f] [-c config_action]
```

Parameters

qip-sitegen recognizes the following parameters:

-n ***dc_svr***	Specifies the fully-qualified domain controller name.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-d ***directory***	Specifies the directory name where the file is to be placed.
-l	Generates a local file that describes the changes without entering them in active directory.
-z	Updates the changed sites and subnets only. If the **-z** parameter is not specified, all sites and subnets are updated.
-f	Forces a push to the domain controller if the domain controller server was previously locked. If it is still locked, an error message is returned indicating it is "push locked".

Command line input example

```
qip-sitegen -n dcsvr.test.com
```

qip-siteimport

qip-siteimport takes site files and subnet files and attempts to create subnets and a subnet organization in VitalQIP. If the network does not exist, the subnet is rejected. If there are conflicts creating the subnet organization (such as when a subnet organization with the same name already exists) the subnet organization creation fails. If there are difficulties creating the subnet (for example, the network does not yet exist for that subnet), the creation of that subnet fails. Only the items that have problems are rejected. If some subnets for a site are valid but others are not, the valid ones are imported into VitalQIP. The invalid ones are written out to a reject file.

Note: **qip-siteimport** does not stop processing if an error is encountered and a site or subnet is rejected; it logs the error to the reject file and continues processing. The error message file is populated with non-recoverable type errors, such as "unable to connect to the database".

Synopsis

```
qip-siteimport [-g loginserver] [-s servername] [-u username]
   [-p password][-o organization] -i sitefile -b subnetfile
   [-r reject_file] [-e errormsg_file]
```

Parameters

qip-siteimport recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the VitalQIP administrator password to be used in establishing the database connection.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-i *sitefile*	Specifies the name of the file that the exported site information is located in. (Generated by Windows 2000 Domain Controller "ldifde" utility.)
-b *subnetfile*	Specifies the name of the file in which the exported subnet information is located. (Generated by Windows 2000 domain controller ldifde utility.)
-r *reject_file*	Specifies the name of the file where rejected sites and subnets are logged.
-e *errormsg_file*	Specifies the name of the file where error messages are logged.

Command line input example

Use the “ldifde” utility preceding this CLI in order to generate the input file. For example,

```
ldifde -f sitefile -d cn=sites,cn=configuration,dc=alexis,dc=cube,dc=com -r
   Objectclass=site
ldifde -f subnetfile-d cn=sites,cn=configuration,dc=alexis,dc=cube,dc=com -r
   Objectclass=subnet
```

where **dc=alexis,dc=cube,dc=com** would be replaced with the user’s active directory domain branch. The file name specified after the **-f** option is the name of the output files; one for sites, one for subnets.

qip-splitmergeenum

qip-splitmergeenum allows you to split NAPTR (Naming Authority Pointer Record) resource records from a parent zone into one or more child zones, and to merge one or more child zones into a parent zone. Regardless of whether you are splitting or merging zones, you only specify the child zones with this command. The parent zone is always detected by the system.

Note: You can only use this command if ENUM is installed.

VitalQIP validates data to prevent CNAME conflicts in the same organization by default. If a CNAME conflict occurs when adding or modifying an NAPTR resource record via the qip-splitmergeenum CLI, an error message displays and the resource record modify fails. You can disable this feature by setting the Validate CNAME Records policy to False. See the Object Policies and Profiles section of the *VitalQIP User Guide* for more information on the policy. When you split the NAPTR resource records, validation checks on the resource record owner data are made against the following:

- Object Names
- Domain Names
- Aliases
- Object Resource Records
- Domain Resource Records
- Reverse Zone Resource Records

Synopsis

```
qip-splitmergeenum -fr domains|-to domains -g Login_Server
   -s servername -u UserName -p Password [-o Organization]
   [-f Output_File] [-df Data_Format]
```

Parameters

qip-splitmergeenum recognizes the following parameters:

-fr|-to *domains*

Specifies if you are merging one or more child domains to a parent domain (**-fr**) or splitting a parent domain into one or more child domains (**-to**).

If you are merging child domains to a parent domain, the domains you specify must already exist. The merged child domains are deleted from VitalQIP.

If you are splitting a parent domain into one or more child domains, if the domains do not exist, the CLI creates them. The domains must be child domains of the parent domain.

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the VitalQIP administrator password to be used in establishing the database connection.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *output_file*	Specifies the directory and filename where the output data is to be placed.
-df *data_format*	Determines the output format. The ***data_format*** variable must be set to **x** for XML output.

Command line input examples

The following commands show a zone split and a zone merge.

Zone Split

The following example splits a parent zone into two child zones.

```
qip-splitmergeenum -to 0.1.6.1.ea64.arpa,5.1.2.1.e164.arpa
-g myloginserver -s mydbsrvr -u qipman -p password
```

The output from the above line is as follows:

```
2 NAPTRs moved from: e164.arpa to: 0.1.6.1.e164.arpa
4 NAPTRs moved from: e164.arpa to: 5.1.2.1.e164.arpa
Please perform ASAP: the pushes of zones changed by the
   split/merge operation.
```

Zone Merge

The following example merges two child zones into their parent zone.

```
qip-splitmergeenum -fr 0.1.6.1.ea64.arpa,5.1.2.1.e164.arpa
-g myloginserver -s mydbsrvr -u qipman -p password
```

The output from the above command is as follows:

```
4 NAPTRs moved from: 5.1.2.1.e164.arpa to: e164.arpa
2 NAPTRs moved from: 0.1.6.1.e164.arpa to: e164.arpa
Please perform ASAP: the pushes of zones changed by the
   split/merge operation.
```

qip-splitrevzone

qip-splitrevzone provides the capability to split reverse zones.

Synopsis

```
qip-splitrevzone [-g loginserver] [-s servername] [-u username]
   [-p password] [-o organization] -f fromZone/mask -t
   toZones/mask
```

Parameters

qip-splitrevzone recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-f *fromZone/mask*	Specifies the existing address and mask of the reverse zone. Use a comma to separate addresses, and precede the mask with a slash (/).
-t *toZones/mask*	Specifies the resultant subnet address or addresses and the new mask that you want. Use a comma to separate addresses, and precede the mask with a slash (/).

Command line input example

```
qip-splitrevzone –u qipman –p passwd –f 144.144.0.0/16 –t
   144.144.0.0,144.144.64.0/19
```

qip-subnetmacpool

qip-subnetmacpool provides the capability to add, delete and query MAC addresses in a subnet's MAC address pool.

Synopsis

```
qip-subnetmacpool [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization]
   -n subnet_addr -a query|add|delete -m mac_address [-x]
   [-t hardware_type]
```

Parameters

qip-subnetmacpool recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-n *subnet_addr*	Specifies the address of the subnet.
-a query|add|delete	Specifies the process: to add or delete a MAC address, or to query the available MAC pool. The optional argument (**-x**) is available for **-a add** only, and indicates that the specific address should be excluded.
-m *mac_address*	Specifies the MAC address to be added or deleted. Supports the trailing wildcard character (*).
-x	Excludes the MAC address from the processing.
-t *hardware_type*	Specifies the hardware type of the object with the MAC address to be added. The following options are available: -Ethernet -Token Ring -AX.25 -Pronet -Chaos -IEEE802 -Arcnet

Command line input example

The following example adds the excluded MAC address “123456* type ethernet” to the MAC pool on subnet 10.100.30.0.

```
qip-subnetmacpool –u qipman –p passwd –n 10.100.30.0 -a add -m
   123456* -x -t ethernet
```

qip-syncexternal

qip-syncexternal retrieves A, AAAA, PTR, SRV, CNAME or TXT resource records from a DNS server. It tries to create External objects with A and PTR resource records. If an object cannot be created, it adds a resource record managed by external updates to a domain or reverse zone.

qip-syncexternal also retrieves A records that have the same name as a domain when SRV and CNAME records are specified in the command line.

When invoked with a **-z** parameter, **qip-syncexternal** retrieves resource records from a particular zone. Otherwise it retrieves resource records for every primary zone on the DNS server.

When invoked with the **-a** parameter (**Append Only mode**), new resource records and objects are added to VitalQIP – no objects or resource records will be tombstoned in VitalQIP.

Resource records that are tombstoned by the qip-syncexternal CLI, can be deleted from VitalQIP with the **qip-tombstonepurge** CLI.

> **Note:** The **qip-syncexternal** CLI command is restricted to capturing CNAME and SRV resource records that are added to DNS by a Windows operating system.

The qip-syncexternal CLI replaces the **qip-miniddma** CLI.

Other than for A and PTR resource records which can appear anywhere in a zone, qip-syncexternal only records or tombstones resource records that exist in the *_msdcs hive* directory.

VitalQIP validates data to prevent CNAME conflicts in the same organization by default. If a CNAME conflict occurs, an error message displays, and you cannot add or modify the profile. You can disable the feature by setting the Validate CNAME Records policy to False. See the Object Policies and Profiles section of the *VitalQIP User Guide* for more information on the policy. When you add or modify records other than CNAME records, validation checks are made against:

- Object aliases
- Object resource records
- Domain resource records
- Reverse zone resource records

When you add or modify CNAME records, validation checks are made against:

- Object names
- Domain names
- Any Object Profile resource records

- Any Domain Profile resource records
- Any Reverse Zone resource records
- ENUM NAPTR resource records
- IPv6 node names
- Mail servers

Synopsis

```
qip-syncexternal [-g loginserver] -d dns_server_name
   [-s servername] [-o organization] [-u username] [-p password]
   [-z zone_name] [-a] [-w] CNAME|SRV|PTR|A|AAAA|TXT|All
```

Parameters

qip-syncexternal recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-d *dns_server_name*	Specifies the fully-qualified name of a DNS server to retrieve records from.
-s servername	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the administrator account.
-z *zone_name*	Specifies the name of a zone on the DNS server to retrieve records from. If this parameter is omitted, **qip-syncexternal** transfers records for all primary forward zones on this DNS server.
-a	Specifies **Append Only** mode, in which new resource records and objects are added to VitalQIP – no objects or resource records are deleted from VitalQIP.
-w	Sets the CLI in verbose mode.

CNAME|SRVIA|AAAA|PTR|TXT|All	Specifies the records to retrieve from the DNS server. Code one or more of these parameters, separated by a space: **CNAME** Retrieves Canonical Name (CNAME) records. **SRV** Retrieves Server Resource (SRV) records. **A** Retrieves Host Ipv4 (A) records and creates object records with A records that are published in DNS. **AAAA** Retrieves Host Ipv6(AAAA) records. **PTR** Retrieves pointer (PTR) records and creates object records with PTR records that are published in DNS **TXT** Retrieves Text (TXT) records. **All** Retrieves CNAME, SRV, A, and PTR records.

Command line input example

- To sync A, CNAME, SRV and PTR records for all primary zones on the DNS server *dns.example.com*:

```
qip-syncexternal -d dns.example.com All
```

- To sync CNAME and SRV records for all primary zones on the DNS server *dns.example.com*:

```
qip-syncexternal -d dns.example.com CNAME SRV
```

- To sync A, CNAME, SRV and PTR records for the zone zone.example.com on the DNS server *dns.example.com*:

```
qip-syncexternal -d dns.example.com -z zone.example.com All
```

- To sync A, CNAME, SRV and PTR records for all primary zones on the DNS server *dns.example.com* without tombstoning any records or objects in VitalQIP:

```
qip-syncexternal -d dns.example.com -z zone.example.com -a All
```

- To sync A, CNAME, SRV and PTR records for all primary zones on the DNS server *dns.example.com* while printing progress messages to standard out:

```
qip-syncexternal -d dns.example.com -w All
```

qip-template

qip-template creates DHCP templates based on the data provided in the input file passed as command parameter **-f**.

Before you begin

- Multiple templates can be added/modified in one input file.
- Use field “template-type” to specify the type of template and a value. The available values are: OPTION, SUBNET_POLICY, CLIENTCLASS_POLICY, SCOPE_POLICY.
- To assign options to policy templates (the template-type is set to SUBNET_POLICY, CLIENTCLASS_POLICY, or SCOPE_POLICY), specify the option name explicitly (for example, renew-address-shuffle=on).
- For options that can have the value “SAME AS IN SUBNET PROFILE” or “SAME AS IN OBJECT PROFILE”, a string value can be set.

Synopsis

qip-template -t *template_type* -f *input_filename* [**-g *loginserver***] [**-s *servername***] [**-u *username***] [**-p *password***] [**-o *organization***] [**-r *reject_file***] [**-e *errmsg_file***]

Parameters

qip-template recognizes the following parameters:

-t *template_type*	Specifies the type of template you are creating. Allowable values are: **OPTION** **SUBNET_POLICY** **CLIENTCLASS_POLICY** **SCOPE_POLICY**
-f *input_filename*	Specifies the name of the file that contains the input data.
-g *loginserver*	Specifies the VitalQIP login server’s IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.

-r *reject_file*	Specifies the directory and filename for the rejected records.
-e *errmsg_file*	Specifies the filename where error messages are written, if they occur.

Command line input example

```
qip-template -s qipdataserver -u qipman -p passwd -f tmpl.dat
```

Input file content examples

Option Template

```
template-name=Testtemp1
template-type=DHCP Option Template
001=User Defined
001>User Defined=198.200.138.0
002=1300
003=User Defined
003>User Defined=198.200.138.156
004=User Defined
004>User Defined=198.200.138.65
005=198.200.138.60
006=User Defined
006>User Defined=198.200.138.45
007=198.200.138.5
008=198.200.138.8
009=198.200.138.46
010=198.200.138.54
011=198.200.138.59
012=User Defined
012>User Defined=nametest
013=43243
014=32534
015=User Defined
015>User Defined=Domain2.qa.quadritek.com
016=198.200.138.4
017=/opt/qip50/qip/rootpath
018=/opt/qip50/path
019=False
020=True
021=198.200.138.0 255.255.255.0
022=65432
023=10000
024=9098
025=5222
026=0
```

```
027=True
028=User Defined
028>User Defined=10.200.70.10
029=False
030=False
031=True
032=198.200.138.179
033=198.200.138.179 198.200.138.165
034=True
035=9999
036=False
037=34938
038=30600
039=False
040=User Defined
040>User Defined=NISDomain
041=User Defined
041>User Defined=198.200.138.53 10.200.60.99
042=User Defined
042>User Defined=198.200.138.5 10.200.60.4
043=whatever1
044=10.200.70.99
045=10.200.60.100
046=B-node
047=testfield1
048=198.200.138.4
049=198.200.138.179
051=Limited
051>Limited Lease Time=8046183
052=60000
058=34502
059=45454
060=101010
061=ci4
064=25345
065=198.200.138.52 10.200.60.4
066=Name Server
067=test bootfile
068=10.200.70.100 198.200.138.77
069=10.200.60.55 10.200.100.6
070=198.200.138.52 10.200.160.2
071=10.200.160.5 198.200.138.74
072=10.200.160.211 198.200.138.211
073=198.200.138.52 198.200.138.56 198.200.138.59
074=10.200.160.53 198.200.138.46
```

```
075=198.200.138.47 10.200.160.5
076=10.200.100.66 10.200.100.166
126=/opt/qip50/qip/bootfile
127=7
```

Subnet Policy Template

```
template-name=testsubnet1
template-type=DHCP Policy Template - Subnet
init-reboot-address-shuffle=on
```

Client Class Policy Template

```
template-name=testcc1
template-type=DHCP Policy Template - Client Class
init-reboot-address-shuffle=off
renew-address-shuffle=on
renew-address-shuffle-max-renews=4
```

qip-tombstonepurge

qip-tombstonepurge deletes tombstoned external objects and resource records from the VitalQIP database. It looks at the record's EDUP timestamp and the database policy DYNDNS|Tombstone Max Life, and deletes any records in the current organization that are older than their natural lifespan. The **qip-tombstonepurge** CLI is installed on the enterprise server and is called periodically by the Schedule Service.

Synopsis

qip-tombstonepurge [**-g** ***loginserver***] [**-s** ***servername***] [**-u** ***username***] [**-p** ***password***] [**-o** ***organization***]

Parameters

qip-tombstonepurge recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.

Command line input example

To delete tombstoned objects and resource records in organization "VitalQIP Organization":

```
qip-tombstonepurge -o "VitalQIP Organization"
```

Output example

```
Successfully purged 0 Objects and 0 Resource Records in Org VitalQIP
   Organization.
```

qip-udf2uda

qip-udf2uda converts a user-defined field to a user-defined attribute and sets the user-defined attribute to the same value as the user-defined field that was converted.

This CLI is intended to be run after a successful migration to VitalQIP 7.2 and above. Only database administrators can run this CLI.

UDFs can be converted to UDAs either one at a time, or in batches. Batches can consist of all UDFs in the database, or just a subset, for example all UDFs associated with IPv4 subnets, or a certain organization.

Synopsis

```
qip-udf2uda [-a UDA_name] [-n_UDF_Name] [-t infrastructure_type]
   [-g UDG_name] [-r] [-g dataserver_name] [-w] [-d] [-u username]
   [-p password] [-o organization]
```

- If you enter only user, password, and organization parameters, all UDFs in the organization are converted to UDAs.
- If you enter a list of UDFs, you must enter a list of UDAs that contains the same number of elements. You can optionally specify an infrastructure type to limit the change to just UDFs in that group.
- If you specify a User-Defined Group (UDG) as a target, the group must exist before you run this command.

Parameters

qip-udf2uda recognizes the following parameters:

-a *uda_name*	Specifies the name of the UDA after the conversion. Separate multiple names with commas, with no spaces before or after the commas. The number of UDA names must match the number of UDF names if these options are specified.
-d	Delete existing UDF after successful conversion to UDA.
-t *infrastructure_type*	Specifies that all UDAs belonging to this infrastructure are to be converted.
-r	List all supported infrastructure types. Use this option similarly to the -h option.
-y *UDG_name*	Specifies the name of User-Defined Group (UDG). The UDG must exist before this command is run.
-n *UDF_Name*	Specifies the name of the UDF to be converted. Separate multiple names with commas, with no spaces before or after the commas. The number of UDA names must match the number of UDF names if these options are specified.
-g	Login server name.

-w	Overwrite existing UDAs with new data from this command.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.

Command line input example

To convert all UDFs to UDAs in the Accounting organization, use the following command:

```
qip-udf2uda –u qipman –p passwd –t Organization -y Accounting
```

qip-ungetdecnetaddr

qip-ungetdecnetaddr releases a DECNet node in a specified DECNet area from the VitalQIP database. Once you run **qip-getdecnetaddr** to get a DECnet node and you decide not to use that address, run **qip-ungetdecnetaddr**. That address is marked as "selected". However, the **qipd** daemon cleans up all selected entries every day.

Synopsis

```
qip-ungetdecnetaddr -a DECnet_area -d DECnet_node
   [-g loginserver] [-s servername] [-u username] [-p password]
   [-o organization]
```

Parameters

qip-ungetdecnetaddr recognizes the following parameters:

-a *DECnet_area*	Specifies the DECNet area.
-d *DECnet_node*	Specifies the DECNet node.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.

Command line input example

To release the DECNet Node 123 on DECNet Area 10:

```
qip-ungetdecnetaddr –u qipman –p passwd –a 10 –d 123
```

qip-ungethubport

qip-ungethubport is used in conjunction with **qip-gethubport**. If you run **qip-gethubport** to get an available hub port and you do not want to use that port, run **qip-ungethubport**. Running **qip-gethubport** marks the port number as "selected". However, the **qipd** daemon cleans up all selected entries every day.

Synopsis

```
qip-ungethubport -b hub_name -l slot_name -n hub_port_num
   [-g loginserver] [-s servername] [-u username] [-p password]
   [-o organization]
```

Parameters

qip-ungethubport recognizes the following parameters:

-b *hub_name*	Specifies the hub name.
-l *slot_name*	Specifies the slot name.
-n *hub_port_num*	Specifies the port number.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.

Command line input example

```
qip-ungethubport -u fred -p passwd -b hub1 -l slot3 -n 55
```

qip-ungetipaddr

qip-ungetipaddr is used in conjunction with **qip-getipaddr**. If you run **qip-getipaddr** to get an available IP address, but you decide not to use that address, run **qip-ungetipaddr** if possible. Running **qip-getipaddr** marks the address as "selected". However, the **qipd** daemon cleans up all selected entries every day.

Synopsis

```
qip-ungetipaddr -a object_address [-g loginserver]
   [-s servername] [-u username] [-p password] [-o organization]
```

Parameters

qip-ungetipaddr recognizes the following parameters:

-a *object_address*	Specifies the object address.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.

Command line input example

```
qip-ungetipaddr –u fred –p passwd –a 199.200.139.27
```

qip-unlock

qip-unlock clears all addresses in a “Selected” state.

Note: Objects should not ordinarily be in a Selected state. If objects are in this state, there may be an issue with the database. Please contact Technical Support (page xxi) before using this CLI.

Synopsis

qip-unlock -n *name*/-a *address*|-c [**-g *loginserver***] [**-s *servername***] [**-u *username***] [**-p *password***] [**-o *organization***]

Parameters

qip-unlock recognizes the following parameters:

-n *name* **|-a *address*** **|-c**	Specifies the hostname of the object (**-n**) or the address (**-a**) that is in a Selected state and needs to be cleared. If you pass **-c**, all addresses in a Selected state are cleared.
-g *loginserver*	Specifies the VitalQIP login server’s IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-o *organization*	Specifies the VitalQIP organization (corporation) name.

Command line input examples

```
qip-unlock -u qipman -p passwd -a 199.200.139.27
qip-unlock -u qipman -p passwd -n myhostname
qip-unlock -u qipman -p passwd -c // unlocks all
```

qip-util

qip-util drops database tables, indexes, and so on, from either the LAM (Audit Manager) or VitalQIP database and the Sybase or Oracle databases. All tables in the Audit Manager database *or* VitalQIP database are changed.

Synopsis

```
qip-util [-t QIPDBASE] [-s QIPDATASERVER] [-u user] [-p password]
   [-l log_file] [-q output_file] [-i script_path] [-b db_name]
   [-a] FUNCTION VALUES… [-z]
```

Parameters

qip-util recognizes the following parameters:

-t *QIPDBASE*	Specifies the database server type: **Oracle** or **Sybase**.
-s *QIPDATASERVER*	Specifies the name of the database server. Note the following: • The database server name must match the Sybase server name or the Oracle database alias name. • This parameter is optional if the $QIPDATASERVER environment variable is set. • The command line argument overrides the environment variable. • For Audit Manager, the Audit Manager database server name must be specified.
-u *user*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-l *log_file*	Specifies the name of the log file.
-q *output_file*	Specifies Quiet Mode. If this parameter is omitted, the output is sent to *STDOUT*.
-i *script_path*	Specifies the directory where the SQL scripts are located.
-b *db_name*	Specifies the database name; VitalQIP or LAM (Audit Manager). You cannot specify both databases at the same time. The default is VitalQIP.
-a	Appends function values discussed in Table 1-38 to the log file. The default is to overwrite the log file.
-z	Allows commands to be executed with the encrypted password.

Table 1-38 Function values

"Function Values" name	Description	Database	Product
CalculateQIPSize	Estimates the size of the VitalQIP database based on the following: *<number_of_objects|number_of_subnets>*	Sybase/ Oracle	VitalQIP
CalculateLAMSize	Estimates the size of the LAM (Audit Manager) database, based on the following: *<number_of_dhcp_clients> <number_of_static_objects> <number_of_nt_objects>*	Sybase/ Oracle	Audit Manager
CheckDatabaseLogin	Checks to see if the connection to Sybase/Oracle is OK.	Sybase/ Oracle	VitalQIP/ Audit Manager
CheckDBProcesses	Shows the number of processes that are currently connected to the Sybase/Oracle database by using: *<database_name>*. The default is **QIP**.	Sybase/ Oracle	VitalQIP/ Audit Manager
CheckVersionFromDatabase	Checks the version information from the qip_version table in the Sybase/Oracle database by using: *<database_name>*. The default is **QIP**.	Sybase/ Oracle	VitalQIP/ Audit Manager
CheckVersionFromData	Checks the version information from the qip_version (qef) file by using: *<export_path>*.	N/A	VitalQIP/ Audit Manager
CheckVersionFromScript	Checks the version information from the **table.sql** script.	Sybase/ Oracle	VitalQIP/ Audit Manager
CheckSybaseDevice	Checks to see if a Sybase device exists by using: *<logical_device_name>*.	Sybase	VitalQIP/ Audit Manager
CheckSybaseDatabase	Checks to see if a Sybase database exists by using: *<database_name>*. The default is **QIP**.	Sybase	VitalQIP/ Audit Manager

"Function Values" name	Description	Database	Product
CheckUserExists	Checks to see if the database login user exists by using: <*login_name*>	Oracle	VitalQIP
ClearAdmin	Removes all logins and users assigned to the database with the related roles or groups by using: <*database_name*>. The default is **QIP**.	Sybase/ Oracle	VitalQIP/ Audit Manager
ClearData	Truncates all data from all user tables by using: <*database_name*>. The default is **QIP**. For Sybase, drops all tables with type equal to "U" (user tables -- not system tables). For Oracle, drops all tables owned by the user running **qip-util**.	Sybase/ Oracle	VitalQIP/ Audit Manager
CreateAccess	Calls **create_access.sql**. This function should only be called by the installation.	Sybase/ Oracle	VitalQIP/ Audit Manager
CreateSybaseDatabase	Creates a Sybase database by using: <*data_device_name*><*data_size*> <*log_device_name*><*log_size*> <*database_name*>	Sybase	VitalQIP/ Audit Manager
CreateSybaseDevice	Creates a Sybase device by using: <*logical_device_name*| *physical_name size*|*device_size*>	Sybase	VitalQIP/ Audit Manager
DropIndex	Drops all indexes on all user indexes in the database by using: <*database_name*>. The default is **QIP**. For Sybase, drops all indexes on all user tables. For Oracle, drops all indexes owned by the user running **qip-util**.	Sybase/ Oracle	VitalQIP/ Audit Manager
DropSP	Drops all stored procedures in the database by using: <*database_name*>. The default is **QIP**. For Sybase, drops all stored procedures in the database. For Oracle, drops all stored procedures owned by the user running **qip-util**.	Sybase/ Oracle	VitalQIP/ Audit Manager

"Function Values" name	Description	Database	Product
DropTable	Drops all user tables in the database by using: *<database_name>*. The default is **QIP**. For Sybase, drops all user tables in the database. For Oracle, drops all tables owned by the user running **qip-util**.	Sybase/ Oracle	VitalQIP/ Audit Manager
DropTrigger	Drops all triggers on all user tables in the database by using: *<database_name>*. The default is QIP. For Sybase, drops all triggers on all user tables in the database. For Oracle, drops all triggers on all user tables owned by the user running **qip-util**.	Sybase/ Oracle	VitalQIP/ Audit Manager
EditTextFile	Finds the text file name and the first line containing the text you want to delete. You can then establish what you would like to add in its place with the “add_line” (optional). Use: *<text_file_name><delete_line> <add_line>*.	N/A	N/A
EstimateRequiredSpace	Estimates the minimum disk space used during qip-import using: *<database_name>*.	Sybase/ Oracle	VitalQIP/ Audit Manager
GetAndSetSybaseDBOption	Gets and sets the value of the Sybase dboption trunc.Log on chkpt by using *<option_value><database_name>*. The default is **QIP**.	Sybase	
GetDatabaseSize	Gets and sets the size of the database by using: *<database_name>*. The default is **QIP**.	Sybase	VitalQIP/ Audit Manager
OracleReCompile	Recompiles all stored procedures and triggers owned by the user running **qip-util** by using: *<database_name>*. The default is **QIP**.	Oracle	VitalQIP/ Audit Manager

"Function Values" name	Description	Database	Product
RebuildIndexes	Drops the database constraints, rebuilds all QIP indexes, and then adds the database constraints back to the database. These steps are necessary in order to rebuild indexes, starting with VitalQIP 7.x.	Sybase/ Oracle	VitalQIP
RemoveInvalidChars	Removes all invalid (unprintable) characters from all files in the specified directory by using: <*export_path|file_name*>	N/A	VitalQIP
RemoveSpaceFields	Goes through all the string fields on all user tables and removes all space-only fields by using: <*table_name*>	Sybase	VitalQIP
SearchReplace	Searches some special characters, and replaces them with proper characters by using: <*search_char*> <*replace_char*>. By default, changes "\n" to a single space; a double quote to a single quote; and "^" to a single space.	Sybase	VitalQIP
SetSybaseConfigure	Sets Sybase configuration values based on the file <*config_file*>. VitalQIP and LAM (Audit Manager)databases require the following settings: - Procedure cache percent=22 - Total memory=21577 - Number of locks=100000	Sybase	VitalQIP/ Audit Manager
SybaseUpdateStatistics	Runs "update statistics" on all tables by using: <*database_name*>	Sybase	VitalQIP/ Audit Manager
OracleUpdateStatistics	Runs an analysis on all objects within the database.	Oracle	VitalQIP/ Audit Manager

vercheck

Note: To run this utility, you should have the Java Runtime Environment installed, and located in the execution path of your environment. If you do not, you will not be able to display the version of any Java-based Lucent product components such as VitalQIP.

The **vercheck** utility lists the VitalQIP version number of every VitalQIP program in a specific directory (and its subdirectories). The resulting information can be displayed on the screen or output to a file and includes the filename, file size, date/time stamp, file type, version number, and the file's "checksum". Using this utility helps you to maintain consistency between upgrades.

Synopsis

vercheck [**-d** ***directory***] [**-m** ***field_mask***] [**-j** ***java_FQN***] [**-c**] [**-e**] [**-z**] [-**5**] [***filename***]

Parameters

vercheck recognizes the following parameters:

-d ***directory***
: Specifies the directory for which version information is to be obtained. The default is the current working directory.

-m ***field_mask***
: Identifies the fields that are to be displayed by **vercheck**. The fields are identified using a field mask consisting of "**0**"s and "**1**"s. A "**1**" indicates that the field should be displayed, and a "**0**" indicates that the field should not be displayed. The fields, in order of specification, are:

- File Name
- File Size
- File Owner
- File Permissions
- File Creation Date
- File Modification Date
- File Type
- File Version
- File Checksum

If this parameter is omitted, all fields are displayed.

-j *java_FQN*

This option allows you to specify the location (the fully-qualified file name) of the JAVA runtime executable (java on UNIX and *java.exe* on Windows) to be used to determine the version of the Java archive (JAR) file. If this parameter is not specified, the utility assumes that the directory containing the JAVA runtime executable is defined in the PATH environment variable. For example, on a UNIX system, you can determine the version of the *acmengine.jar* file using either of the following methods (this assumes Bourne Shell or K-Shell in this example):

```
# vercheck -z -j /usr/local/jdk1.3/bin/java
   acmengine.jar
   acmengine.jar: 1.1.5
```

or

```
# PATH=$PATH:/usr/local/jdk1.3/bin; export
PATH
# vercheck -z acmengine.jar
acmengine.jar: 1.1.5
```

-c

Outputs the information in CSV format.

-e

Provides VitalQIP environment information.

-z

Provides only the filenames and product version numbers.

-5

Searches only for VitalQIP 5.x versions. If this parameter is omitted, **vercheck** searches for all VitalQIP versions.

filename

Provides information about the specified file. If this parameter is omitted, all files in the current directory or the directory specified by the **-d** option is processed. The subdirectories are also be processed.

2 Export and import whole databases

Overview

VitalQIP has various types of import and export capabilities. You can import and export whole VitalQIP databases with the **qip-import** and **qip-export** CLI commands. These CLI commands are mainly used for upgrades or backup/recovery purposes.

Note: Make sure you have shut down your VitalQIP services before running either of these CLI commands.

Note: If the VitalQIP Oracle database server is configured to use a language other than American English, ensure that the **`NLS_LANG`** environment variable is set to the correct language. If the environment variable is not set to the correct language, VitalQIP data will not be imported and exported in the correct language to the database. For more information on changing the environment variable, refer to Chapter 4, “Install VitalQIP enterprise server”, in the *VitalQIP Installation Guide*.

qip-export

qip-export exports all data from either the VitalQIP database *or* Audit Manager database for use with Sybase and Oracle. This can be performed on either a UNIX or Windows platform. It does not change the data in the database. Error data is output to the *QIPHOME/log/qip-export.log* file.

Synopsis

```
qip-export [-t QIPDBASE] [-s QIPDATASERVER] [-u user]
   [-p password] [-l log_file] [-q output_file] [-i script_path]
   [-b database_name] [-d export_path] [-a] [-k] [-z
   encrypted_password]
```

Parameters

qip-export recognizes the following parameters:

-t ***QIPDBASE***	Specifies the type of database – either **SYBASE** or **ORACLE**. This parameter is optional if the $QIPDBASE environment variable is set. The command line argument overrides the environment variable.
-s ***QIPDATASERVER***	Specifies the name of the database server. Note the following: • The database server name must match the Sybase server name or the Oracle database alias name. • This parameter is optional if the $QIPDATASERVER environment variable is set. • The command line argument overrides the environment variable. • For Audit Manager, the Audit Manager database server name must be specified.
-u ***user***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-l ***log_file***	Specifies the file name of the log file.
-q ***output_file***	Quiet Mode. If this parameter is omitted, the output is sent to *STDOUT*.
-i ***script_path***	Specifies the directory where the SQL scripts reside.
-b ***db_name***	Specifies the database name; QIP *or* LAM (Audit Manager). You cannot specify both databases at the same time. The default is QIP.
-d ***export_path***	Specifies the path and directory of the export data file.
-a	Appends to the log file. The default is to overwrite the log file.
-k	Skips the prompt.

-z encrypted password	Allows commands to be executed with the encrypted password.

Note: The **qip-export** utility performs a "net start" command to determine what processes are currently running on Windows. It will cycle through all of these processes and kill the VitalQIP processes. There is a technical limitation on Windows with respect to the number of processes the “net start” buffer can hold and therefore the **qip-export** utility might encounter one, few, or all VitalQIP processes running on the machine. It depends on how many overall processes are running on the Windows box. You might be prompted more than once to kill all the VitalQIP processes until all of them are shutdown. If the number of Windows processes running on the machine is low, all the VitalQIP processes will be shutdown via one prompt. When you run the **qip-export** utility on Windows, you are prompted to kill all VitalQIP processes, prior to running the **qip-export**. If you specify the **-k** parameter when running the CLI, you will not be prompted to kill the VitalQIP processes and the export process will move ahead. Alternatively, you can limit the number of overall processes running on the machine, to avoid multiple prompting.

Command line input example

```
qip-export -u qipadmin -p <password> -d /tmp/qip
```

qip-import

qip-import imports either all VitalQIP *or* all Audit Manager data from an earlier database export into the database for use with Sybase and Oracle. It overwrites the entire database. Error data is output to the file *$QIPHOME/log/qip-import.log*. The **qip-import** command is only installed with the enterprise server component and can only be run on the VitalQIP enterprise server. The **-d** parameter is required for the **qip-import** command.

Note: To ensure optimal database performance, Oracle database administrators can have the **qip-import** utility automatically analyze the VitalQIP tables. Remove the comment character (#) from the following line in the Regular Import section of the *$QIPHOME/script/qipinst.config* file:

```
#qip-util(oracle)::OracleUpdateStatistics
```

Synopsis

qip-import [**-t** ***QIPDBASE***] [**-s** ***QIPDATASERVER***] [**-u** ***user***]
[-**p** ***password***] [**-l** ***log_file***] [**-q** ***output_file***] [**-i** ***script_path***]
-d ***input_path*** [**-a**] **[-k]** [**-z** ***encrypted_password***]

Parameters

qip-import recognizes the following parameters:

-t ***QIPDBASE***
Specifies the type of database - either SYBASE or ORACLE. This parameter is optional if the $QIPDBASE environment variable is set. The command line argument overrides the environment variable.

-s ***QIPDATASERVER***
Specifies the name of the database server. Note the following:

- The database server name must match the Sybase server name or the Oracle database alias name.
- This parameter is optional if the $QIPDATASERVER environment variable is set.
- The command line argument overrides the environment variable.
- For Audit Manager, the Audit Manager database server name must be specified.

-u ***user***
Specifies the VitalQIP administrator account to be used in establishing the database connection. If the administrator is not specified, it will use the information in the *qip.pcy* file.

-p ***password***
Specifies the password for the associated administrator account. If the administrator is not specified, it will use the information in the *qip.pcy* file.

-l ***log_file***
Specifies the file name of the log file.

-q ***output_file***	Specifies Quiet Mode. If this parameter is omitted, the output is sent to *STDOUT*.
-i ***script_path***	Specifies the directory where the SQL scripts reside.
-d ***input_path***	***Required***. Specifies the path (directory) of the input data.
-a	Appends to the log file. The default is to overwrite the log file.
-k	Skips the prompt.
-z ***encrypted password***	Allows commands to be executed with the encrypted password.

Command line input example

```
qip-import -u qipadmin -p <password> -d /tmp/qip
```

3 Exporting and importing DNS and Bootptab files

Overview

VitalQIP offers DNS and Bootptab file import and export utilities, which allow you to extract current BIND 9.x or bootp data and import it into your VitalQIP system.

These import/export utilities are available on both UNIX and Windows platforms and can be used for either upgrades or new installations.

Exporting DNS files

The **qip-dnscsv** CLI command searches all DNS database files and extracts data to create CSV (comma separated variable length) files capable of being imported into VitalQIP. (Alternately, you can use the CLI, in interactive mode, to transfer DNS data directly from an operational DNS server.) These files are created in the format specified in the sections describing each CLI command.

The extracted data can be imported by using the CLI commands listed in Table 3-1 (and explained later in this chapter).

Table 3-1 CLI import commands

CLI Command	Description
enterserver	Use to import DNS servers.
enterdomain	Use to import Domain information.
enternetwork	Use to import Network information.
entersubnet	Use to import Subnet information.
entersimpleobj ***or*** qipbulkload	Use to import Object information.
enterdnsobj	Use to import Resource Record information for objects.
enterdnsrr	Use to import Resource Record information for zones.
enteralias	Use to import additional Object Alias information.

Specific information must be provided to the CLI, such as DNS BIND version information, the location of the *named.conf* (BIND 9.x) file, the location of the output files, the subnet mask to derive the subnet address, and the location of the “rules” file. The “rules” file is explained later.

Figure 3-1 explains the process.

Figure 3-1 The DNS export process

qip-dnscsv does the following:

1. Read the DNS startup file (*named.boot* or *named.conf*) to find the DNS zones and database files. If the **-z** parameter is used to specify the zone in the interactive mode, the DNS startup file is not read.
2. Optionally, read the “rules” file to obtain user preferences as to how to extract information from the database files. The “rules” file enables the user to force **qip-dnscsv** to obtain information from the Resource Records, specifically the network and subnet mask, to extract domain names from SOA only or from all objects.
3. Format the information into specific CSV formats so data can be imported into VitalQIP.
4. Retrieve the DNS information directly from the server in interactive mode. Otherwise, it will retrieve the information from reading the *named.boot/named.conf* file, locating the directory of the database files, and going through all database files to extract the required information.
5. Additionally, it reads resource records and retrieves the corresponding information, as shown in Table 3-2 following.

6. From the reverse zone file, **qip-dnscsv** obtains the PTR record to determine which name to use as object name when multiple A records of the same IP are present.

   Note: **qip-dnscsv** only imports data from Master servers (not slaves).

Table 3-2 Resource Records

Resource Records	Retrieved information
SOA	Domain Name, Primary Server Name, Mail address of contact, Refresh Interval, Retry Time, TTL, and minimum TTL
NS	Secondary Servers
A	Host Name and IP Address*
CNAME	Host Alias
HINFO	CPU Type, OS Type
WKS	Comment
TXT	Object Description (free form information)
$ORIGIN	The base Domain for the subsequent records
$INCLUDE	The filename and, if available, the new ORIGIN

A few things to keep in mind*

- If the $ORIGIN statement exists, there may also be an $INCLUDE record with it. If the $ORIGIN domain name and the $INCLUDE domain name differ, the domain with the $INCLUDE record overrides the domain in the SOA and the $ORIGIN statement. However, the $ORIGIN record may not exist. If so, the $INCLUDE record has the domain name following the file name.
- If the Owner of an A resource record is a domain, “NULL” is assigned as the object name. Only one object is created. If more than one object shares an IP address, the other objects assigned to that IP address are created as resource records.
- $GENERATE statements are supported. For example:

```
$GENERATE 10-20 QIPDNS$ 10.20.30.$
```

  Please refer to “Output File 5 - qip-dnsobject.csv” (p. 3-16) for details of resource records generated by the $GENERATE statement.

qip-dnscsv

Purpose

This section describes how to use the **qip-dnscsv** CLI. First, you need to gather certain information and do some preparation.

Procedure

Before using the **qip-dnscsv** CLI, do the following:

1. Formulate the optional "rules" file, based on the description in the section on "Using the "Rules" file with qip-dnscsv" (p. 3-8)", and give it an appropriate name, for example: */home/you/named/dnsrule.dat*.

2. Find out the DNS BIND version in your facility (for example, BIND 9.x).

3. Find out the directory name of the DNS startup files. Normally, it is in */etc/named/named.conf* (for BIND 9.x).

4. For interactive mode, obtain the IP address of the DNS server to be used as the source of the information.

5. Decide what directory you want the output to go to.

6. Find out where **qip-dnscsv** resides and verify that it is in your path.

7. Initiate the process by typing the command in the command line. For example:

```
qip-dnscsv -b 9 -n /etc/named/named.boot -o /home/you/named
   -r /home/you/named/dnsrule.dat
```

In this example, you have BIND 9.x in your facility, and you want to retrieve specific DNS information from the Resource Records by using the "rules" file specified in the **-r** parameter, specifically:

/home/you/named/dnsrule.dat

In another example, the process is initiated in interactive mode:

```
qip-dnscsv -b 9 -n /etc/named/named.conf -o /home/you/named
   -r /home/you/named/dnsrule.dat -i -s 10.200.20.10
```

In this example, you have BIND 9.x running on the server at IP address 10.200.20.10. The zones to be processed are read from the *named.conf* files. The same rule file is used as in the previous example.

8 In both examples in Step # 7, the output files are placed in the */home/you/named* directory by using the **-o** parameter. They are labeled, as follows:

- *qip-dnssvr.csv* (the DNS server information)
- *qip-dnsdomain.csv* (the domain information)
- *qip-dnsnet.csv* (the network information)
- *qip-dnssn.csv* (the subnet information)
- *qip-dnsobject.csv* (the object information)
- *qip-dnsobjrr.csv* (the resource record information associated with objects)
- *qip-dnszonerr.csv* (the resource record information associated with zones)
- *general.ext* (additional object alias or subnet domain information)

There is also a log file, called *qip-dnscsv.log* (the process log), in the */home/you/named* directory.

9 Use your favorite editor to review and modify, as required, before you import.

10 Import, using the CLI commands discussed in Chapter 4, "Exporting and importing with CLIs".

END OF STEPS

Synopsis

```
qip-dnscsv -b 9 -n [named.boot|named.conf]
   -r rules_file_name|-m subnet_mask [-d SOA|AllObj]
   [-o output_directory] [-c default_object_class] [-v] [-h]
```

or

```
qip-dnscsv -b 9 -i -s dns_svr_IPaddr -z zone_name|
   -n named.boot|named.conf [-d SOA|AllObj] [-o output_directory]
   [-c default_object_class] [-v] [-h]
```

Parameters

qip-dnscsv recognizes the following parameters:

-b 9	Specifies the DNS BIND version number. The Value is **9** for BIND 9.x. **Note:** If you use this CLI to import Lucent DNS data, the server is represented as a BIND server type.
-n named.boot|named.conf	Specifies the location of the DNS startup file. This must be the full file name (for example, *n/etc/named/named.boot*). **Note:** This is optional if running in interactive mode and specifying the zone to be retrieved using the **-z** parameter.
-o *output_directory*	Specifies the location of the output files. This must be the full directory name (for example, */home/youroutputfile/named*).
-r *rules_file_name*	Specifies the location of the "rules" file. You must specify the full file name (for example, */home/you/named/dnsrule.dat*).
-m *subnet_mask*	Specifies the mask to use to derive network mask length and subnet address.
-d SOA|AllObjects	To create domain records only or from all objects. This can be specified in the rules file also. The default is **AllObjects**.
-c *default_object_class*	If no rule can be found pertaining to the object class, the value supplied here is used as the default object class. If no value is supplied, the class "Undefined" is assigned.
-i	Specifies interactive mode. This specification allows the CLI to retrieve information directly from the server specified by the **-s** parameter.
-s *dns_svr_IPaddr*	Specifies the IP address of the DNS server from which to retrieve the information. This is mandatory if running in interactive mode.
-z *zone_name*	Specifies the zone name to be retrieved from the DNS server. This is only valid in interactive mode if the **-n** parameter is not specified.

Using the "Rules" file with qip-dnscsv

Purpose

You must create the "rules" file manually if you want to have the **qip-dnscsv** CLI command retrieve specific information from the DNS Resource Records and place them in the appropriate *.csv* files.

The "rules" file is called whenever **qip-dnscsv** is run. Below are two examples of the "rules" file. The second example uses the Token List so you can "map" the specific data to other data in *qip-dnsobject.csv*.

Note: Multiple rules can be defined in one "rules" file.

The "rules" file allows you to:

1. Specify how the network and subnet masks are to be used, since that information is not available in most BIND database files.

2. Specify how to derive an object class description by mapping information from the A, CNAME or TXT Resource Record. For example, the alias for the **host101** is **gwto101** as it appears on a CNAME record. You can specify a "rule" to search for "gw" at the beginning of a name, and if found, attach the object class description for that object as "Gateway".

   The "rules" file setup must be entered manually in sets of **`name=value`** pairs. The directory path and file name of the "rules" file is passed to the program as:

```
qip-dnscsv -b 9 -n /etc/named.boot -o /home/usr/qip/csvdir
   -r /usr/qip/rulesfile –m 255.255.255.0
```

The "name" portion of the pair can be any or all of the names in Table 3-3 following. The "value" choices for each "name" are discussed in the Value Description column.

Table 3-3 "Rules" file definitions

Name	Value description	Value example
CSVFile	This indicates to which CSV file the rule is applicable. Currently, the only choice is **`SimpleObject`**.	**`CSVFile=SimpleObject`** This rule is applicable during the creation of the object CSV file (*qip-dnsobject.csv*).

Name	Value description	Value example
Target	This indicates to which data item on the output record the rule is applicable. The choices are **Comment** or **ObjectClass**. Reference the import CLI"entersimpleobj" (p. 4-47) .	**Target=Comment** This target places the found information of the Resource Record (for example, TXT record) in the "Comment" field of the *qip-dnsobject.csv* file. **Target=ObjectClass** This target places the found information (based on the value of the Source below) of the Resource Record (for example, TXT record) in the Object Class field of the *qip-dnsobject.csv* file.
Source	The Source is directly related to the Target. The Source value indicates which input record type and field is used to derive the output information. That output information is placed in the **Target** (either Comment or Object Class). The valid values are **TXT<txt-strings>** or **A<owner>**.	**Source=TXT<txt-strings>** If the Source is TXT<txt-strings>, the information in this portion of the TXT record is placed in the specified Target (either Object Class or Comment). **Source=A<owner>** If the Source is A<owner>, the information in this portion of the A record is placed in the specified Target (either Object Class or Comment). <owner> is the hostname read to the first ".". In other words, if the information in the **<owner>** field is *quad1.quadritek.com*, the data used will only be **quad1**.
Method	This value is directly related to the **Source** and the **Target**. This indicates the source information is to be placed in the target field as a whole or is parsed and mapped to some other value based on the provided listing. The choices are **Insert** or **Map**.	**Method=Insert** This indicates the data from the **Source** is placed in the **Target** field as a "whole". **Method=Map** This indicates that the data from the **Source** is mapped to a specific value before placing it in the **Target** field. Refer to Example #2 below for a more complete explanation.

Name	Value description	Value example
TokenList	The Token List lists all the targets (Object Class or Comment) you want to map to other data in the object record. Refer to Example #2 following for more information.	
Network Masks	The network mask length can be specified in the rules file; otherwise, **qip-dnscsv** will take default action, assuming a standard class A, B, C boundary.	**`NetworkMask=begin`** **`20.0.0=255.255.0.0`** **`NetworkMask=end`**
Subnet Masks	The subnet mask to be used for a network can be specified in the rules, if it is not just a common mask for subnets. With this rule defined, **qip-dnscsv** loops through all the Object IPs that have A records, creates subnet IP addresses, and places them in the **qip-dnssn.csv** file. Refer to Example #2.	**`SubnetMask=begin`** **`20.100.23.0=255.255.255.0`** **`20.100.100.0=255.255.255.0`** **`SubnetMask=end`**
Domain Record	The Domain Record specification indicates whether to extract domains from SOA only or from all objects. The choices are **`SOAonly`** or **`FromAllObject`**. Example #1 illustrates the use of this rule.	**`DomainRecord=SOAonly`** Indicates extraction from SOA records only. If this is specified and the A record contains an object name with a dotted name, it will be written to the *domain.ext* file as a resource record instead of having an object record created. **`DomainRecord=FromAllObject`** Indicates extraction from all objects.

Example #1

```
DomainRecord=FromAllObject
   CSVFile=SimpleObject
   Source=TXT<txt-strings>
   Target=Comment
   Method=Insert
```

The above example in the "rules" file defines that the data in the **`<txt-strings>`** portion of the TXT resource record should be placed in the "Comment" field of *qip-dnsobject.csv*. It also specifies that the data should be placed there as a "whole" (Insert).

If the **<txt-strings>** field of the TXT resource record specifies "This is a wiring hub", the Comment field in the *qip-dnsobject.csv* record would read exactly that. The *qip-dnsobject.csv* would look like this:

```
usa.world.com,198.200.234.2,hub-ny-1,,,Wiring_HUB,This is a
   wiring hub
CSVFile=SimpleObject
   Source=A<owner>
   Target=Object Class
   Method=Insert
```

The above example in the "rules" file defines that the data in the **<owner>** portion of the A resource record should be placed in the Object Class field of *qip-dnsobject.csv.* It also specifies that the data should be placed there as a "whole".

If the **<owner>** field specifies "This is a wiring hub", the Object Class field in the *qip-dnsobject.csv* record would read exactly that. The *qip-dnsobject.csv* would look like this:

```
usa.world.com,198.200.234.2,hub-ny-1,,,This is a wiring hub
```

Example #2

```
;rule1
CSVFile=SimpleObject
   Target=Comment
   Source=TXT<txt-strings>
   Method=Insert
   ;rule2
   CSVFile=SimpleObject
   Source=A<owner>
   Target=ObjectClass
   Method=Map
   TokenList=Begin
   Search end=hub=Wiring_HUB
   Search begin=gateway=Gateway
   Search Anywhere=router=Router
   Search Begin=rtr=Router
   TokenList=End
   ;rule3
   mask=begin
   10=255.255.255.0
10.230.255=255.255.255.240
100=255.255.0.0
74=255.255.0.0
75=255.255.0.0
205=255.255.255.0
mask=end
NetworkMask=begin
209.249.64=255.255.252.0
216.200.28=255.255.252.0
216.200.232=255.255.252.0
```

```
209.185.209=255.255.252.0
NetworkMask=end
SubnetMask=begin
209.249.64=255.255.254.0
216.200.28=255.255.252.0
216.200.232=255.255.255.128
209.185.209=255.255.252.0
SubnetMask=end
```

Note: Comments must be preceded by **#** (a pound sign) or **;** (a semi-colon).

Example #2, rule 1, indicates that the data in the **`<txt-strings>`** portion of the **TXT** Resource Record should be placed in the Comment field of *qip-dnsobject.csv*. It also specifies that the data should be placed there as a "whole" (Insert).

However, the second rule (rule2) states that the <owner> portion of the **A** record should be placed in the Object Class field of *qip-dnsobject.csv*. The information should be replaced with other information (indicated by **`Method=Map`**) as per the Token List. All values used to designate Object Class are placed between **`TokenList=Begin`** and **`TokenList=End`**.

Looking more closely at the Token List example from the above rule, you can see that the program can determine that if any data in the **`<owner>`** portion of the **A** record ends with "hub", it will be replaced in the "Object Class" field of *qip-dnsobject.csv* with the word "Wiring_HUB".

```
TokenList=Begin
   Search end=hub=Wiring_HUB
   Search begin=gateway=Gateway
   Search Anywhere=router=Router
   Search Begin=rtr=Router
   TokenList=End
```

- If the **`<owner>`** portion of the A record contains the word "router" ***anywhere*** in the data it is replaced with the word "Router" in the "Object Class".
- If any data in the **`<owner>`** portion of the A records ***ends*** with "hub", it is replaced with "Wiring_Hub" in the "Object Class" description.

Note: The information in the Token List is case-insensitive.

This example (rule 3) also creates subnet for the associated object as follows, as indicated by the **`mask=begin`** and **`mask=end`** code.

- Subnet 10.114.60.0 for object 10.114.60.10
- Subnet 10.230.4.0 for object 10.230.4.20
- Subnet 10.100.0.0 for object 10.100.18.64
- Subnet 205.200.10.0 for object 205.200.10.10

These masks are placed in the *qip-dnssn.csv* file.

qip-dnscsv output files

This section uses examples to help explain the output file layouts for the eight files created by **qip-dnscsv**. As shown in the examples, each field is separated by a comma. For a full explanation of the file formats, refer to Chapter 4, "Exporting and importing with CLIs".

Output File 1 - qip-dnssvr.csv

This file contains the DNS servers found in the DNS database files (obtained mostly from the NS records). To import this information, use the **enterserver** CLI command, specifying *qip-dnssvr.csv* as the value for the **-f** parameter (for example, **-f qip-dnssvr.csv**).

```
QIPDNS2, qip.com, c:\\qip\\named\, xyz@lucnet.com, 12345
```

For a description of the contents of the following file, refer to Table 3-4.

Table 3-4 Output File Definitions

Record Value	Field Name	Comments
QIPDNS2	Host Name	
qip.com	Domain Name	
c:\\qip\\named	Default Directory	
xyz@lucent.com	Email address for local and reverse zones	
12345	RNDC key	Plain text representation of RNDC-key definition in named.conf. Applicable only for BIND 9.

Output File 2 - qip-dnsdomain.csv

This file contains the DNS Domain information. To import this information, use the **enterdomain** CLI command, specifying *qip-dnsdomain.csv* as the value for the **-i** parameter (for example, **-i qip-dnsdomain.csv**).

For a description of the contents of the following file, refer to Table 3-5.

```
usa.world.com,21600,604800,3600,86400,root@dns1.world.com,dns1.world.com,
   P,dns2.usa.world.com,S,dns3.usa.world.com,S,dns4.uk.world.com,S
```

Table 3-5 Output file definitions

Record value	Field name	Derived from	Comments
usa.world.com	Domain Name	SOA Record	Required
21600	Refresh Time	SOA Record	Required

Record value	Field name	Derived from	Comments
604800	Expire Time	SOA Record	Required
3600	Retry Time	SOA Record	Required
86400	Minimum Time	SOA Record	Required
root@dns1.world.com	Zone e-mail Address	SOA Record	Required
dns1.world.com	Primary Server	SOA Record	Optional
P	Primary Server Type	From the **-b** parameter of the **qip-dnscsv** process.	Required P=primary
dns2.usa.world.com	Additional Server 1	NS Record	Optional
S	Server Type		Required if a Server 2 exists. P=primary S=secondary
dns3.usa.world.com	Additional Server 2	NS Record	Optional
S	Server 2 Type		Required if a Server 2 exists. P=primary S=secondary
dns4.uk.world.com	Additional Server 3	NS Record	Optional
S	Server 3 Type	NS Record	Required if a Server 3 exists. P=primary S=secondary

Output File 3 - qip-dnsnet.csv

This file contains the DNS network information. Use the **enternetwork** CLI command to import this file, using *qip-dnsnet.csv* as the value for the **`-f`** parameter (for example, **`-f qip-dnsnet.csv`**).

The following example is used to explain the output file contents. Note that the example (and Table 3-6 following) only explains the first row in the file. More "Optional" fields (that can be imported) are available.

```
150.1.6.0,24,root@unilever.com,net_150.1.6.0,N
```

Table 3-6 Output file definitions

Record value	Field name	Derived from	Comments
150.1.6.0	Network Address	Xxxin-addr.arpa	Required
24	Network Mask length	System-generated	Required
root@unilever.com	e-mail address	System-generated	Required
Net_150.1.6.0	Network Name	System-generated	Optional
N	CIDR indicator	System-derived	Optional

Output File 4 - qip-dnssn.csv

This file contains the DNS subnet information. Use the **entersubnet** CLI command to import this file, using *qip-dnssn.csv* as the value for the **-f** parameter (for example, **-f qip-dnssn.csv**).

The following example is used to explain the output file contents. Note that the example (and Table 3-7 following) only explains the first row in the file. More "Optional" fields (that can be imported) are available.

```
150.1.6.0,255.255.255.192,150.1.0.0,SN_150.1.6.0,usa.world.com
```

Table 3-7 Output file definitions

Record value	Field name	Derived from	Comments
150.1.6.0	Subnet Address	Zone (startup file)	Required
255.255.255.0	Subnet Mask	System-generated*	Required
150.1.0.0	Network Address	Reverse Zone	Required
SN_150.1.60.0	Subnet Group Name	System-generated*	Required
usa.world.com	Domain Name	SOA Record of the DNS startup file	Optional

Note: Derivation of the Network Address* = The Network address is obtained from the numeric part of the zone name when it is in the form of X.Y.Z.IN-ADDR.ARPA, where X,Y, and Z are numerical. If a mask is not provided in the rules file or passed as a command parameter, and if there are two numeric values in front of IN-ADDR.ARPA, a 255.255.0.0 mask is assumed. If there are three numeric values in

front of IN-ADDR.ARPA, a 255.255.255.0 mask is assumed for the network IP of Z.Y.X.0.

Derivation of the Subnet Address* = The Subnet Address is derived from subnet mask and the object IP address.

Output File 5 - qip-dnsobject.csv

The file contains the DNS object information. Use the **entersimpleobj** CLI command to import this file, using *qip-dnsobject.csv* as the value for the **-f** parameter (for example, **-f qip-dnsobject.csv**).

The following example is used to explain the output file contents. Note that the example (and Table 3-8 following) only explains the first row in the file.

```
1980.200.234.2,alps,,,world.com,Router,,,,,,,,,,router-
   alps,Router-1
```

Use of $GENERATE statements in zone files is supported. For example, the following statement:

$GENERATE 10-20 QIPDNS$ 10.20.30$ statement for qip.com zone will result in the following records being created in *qip-dnsobject.csv*:

```
10.20.30.10, QIPDNS10, , , qip.com, Undefined
10.20.30.11, QIPDNS11, , , qip.com, Undefined
10.20.30.12, QIPDNS12, , , qip.com, Undefined
10.20.30.13, QIPDNS13, , , qip.com, Undefined
10.20.30.14, QIPDNS14, , , qip.com, Undefined
10.20.30.15, QIPDNS15, , , qip.com, Undefined
10.20.30.16, QIPDNS16, , , qip.com, Undefined
10.20.30.17, QIPDNS17, , , qip.com, Undefined
10.20.30.18, QIPDNS18, , , qip.com, Undefined
10.20.30.19, QIPDNS19, , , qip.com, Undefined
10.20.30.20, QIPDNS20, , , qip.com, Undefined
```

Table 3-8 Output file definitions

Record Value	Field Name	Derived From	Comments
198.200.234.2	Object IP Address	A Record	Required
Alps	Object (Device) Name	A Record	Required
world.com	Domain Name	SOA Record or the DNS startup file	Optional
Router	Object Class Description	TXT Record or A Record	Optional

Record Value	Field Name	Derived From	Comments
router-alps	Alias Name 1	CNAME Record	Optional.
Router-1	Alias Name 2	CNAME Record	Optional.

Output File 6 - qip-dnsobjrr.csv

This file contains the DNS resource record information. Use the **enterdnsrr** CLI command to import this file, using *qip-dnsobjrr.csv* as the value for the **-f** parameter and **object** as the owner type (in the **-t** parameter); for example, **-f qip-dnsobjrr.csv -t object**.

The following example is used to explain the output file contents. Note that the example (and Table 3-9 following) only explains the first row in the file. More "Optional" fields (that can be imported) are available.

```
150.1.6.10,pco10.qtek.com,,TXT,,"This is a test field"
```

Table 3-9 Output file definitions

Record value	Field name	Derived from	Comments
150.1.6.10	Owner IP	A record	The owner of the resource record.
pco10.qtek.com	Owner Name	A record	
Txt	Resource Record type	Resource Record	
This is a text field	Resource Record text	Resource Record	

Output File 7 - qip-dnszonerr.csv

This file consists of resource records that are associated with zones. Use the **enterdnsrr** CLI command to import this file, using *qip-dnszonerr.csv* as the value for the **-f** option and **-t domain** as the owner type (for example, **-f qip-dnszonerr -t domain**).

```
Qtek.com.,,pco10.qtek.com,,MX,,"This is a test field"
```

Output File 8 - general.ext

This file consists of object alias records that the **qip-dnscsv** utility associated to an IP address, but there were too many aliases to be written to the *qip-dnsobject.csv* file. The records contain that object IP address with a list of aliases. Refer to **enteralias** for further details.

```
198.200.138.10,mail.city.com, cckk.net
```

Importing DNS files

The DNS CSV files must be imported in the following sequence:

1. *qip-dnssvr.csv* is imported into VitalQIP by using **enterserver**.
2. *qip-dnsdomain.csv* is imported into VitalQIP by using **enterdomain**. (If there are additional DNS servers specified in the record, those DNS servers must exist.)
3. *qip-dnsnet.csv* is imported into VitalQIP by using **enternetwork**.
4. *qip-dnssn.csv* is imported into VitalQIP by using **entersubnet**.
5. *qip-dnsobject.csv* is imported into VitalQIP by using **entersimpleobj**.
6. *qip-dnsobjrr.csv* is imported into VitalQIP by using **enterdnsrr**.
7. *qip-dnszonerr.csv* is imported into VitalQIP by using **enterdnsrr**.
8. *general.ext* is imported into VitalQIP by using **enteralias**.

For detailed information on these import CLI commands, refer to the previous section.

Exporting Bootptab files

The Bootptab File Export utility CLI command, **qip-bootptabcsv**, searches the Bootptab file and captures Bootp object and subnet information in CSV formats. These files are created in the same format as the import utilities currently available in VitalQIP.

Note: Mac addresses in the bootptab file are required to have a ‘0x’ prefix, as is the case with the VitalQIP-generated bootptab files. When using the **qip-bootptabcsv** CLI with bootptab files from other servers this is not always the case and can result in the MAC address being truncated. Alcatel-Lucent recommends you check the bootptab file and add a ‘0x’ prefix where necessary.

The import of the Manual Bootp object information is performed using the **qip-setobject** import module. The Network information is imported using the **enternetwork** module. The Subnet information is imported using the **entersubnet** module.

qip-bootptabcsv

qip-bootptabcsv reads the Bootptab file and creates the following output CSV files, if possible:

- *qip-bootpobject.fmt* - This file contains the Object information *field names* from the Bootptab file for the **qip-setobject** import file.
- *qip-bootpobject.csv* **-** This file holds the *values* of the Object information field names as listed in the *qip-bootpobject.csv* file.
- *qip-bootpnet.csv* - This file contains the network information in the format required by the **enternetwork** module.
- *qip-bootpsubnet.csv* - This file contains the subnet information in the format required by the **entersubnet** module.

Synopsis

```
qip-bootptabcsv [-i bootptab_file_name]
   [-o output_file_directory] [-d default_domain] [-v] [-h]
```

Parameters

qip-bootptabcsv recognizes the following parameters:

-i ***bootptab_file_name***	***Required***. Specifies the location and file name of the Bootptab file. This must be the full file name (for example:*/etc/boot/bootptabfile*).
-o ***output_file_directory***	***Required***. Specifies the location (Directory name) of the output files. This must be the full file name (for example, */home/your/bootp*).
-d ***default_domain***	***Optional***. Specifies the default domain name. This must be the full file name (for example, *qtk.com*).

Using qip-bootptabcsv

Purpose

This section describes how to use the **qip-bootptabcsv** CLI. First, you need to gather certain information and do some preparation.

Procedure

Before using the **qip-bootptabcsv** CLI, do the following:

1. Find out the location and the file name of the bootptab file (for example, */etc/bootp/bootptab*).

2. Decide where the output of the process should go (for example, */home/your/bootp*).

3. You must provide a default domain name (for example, *qtek.com*) in case records within the bootptab file are missing a domain. Decide what you want the default domain name to be.

4. Make sure that **qip-bootptabcsv** is in your path.

5. Initiate the process by typing the command in the command line. For example:

```
qip-bootptabcsv -i /etc/bootp/bootptabfile -o /home/your/bootp -d qtek.com
```

6. The output files are stored in the output directory you specified above with the following names:

   qip-bootpnet.csv, *qip-bootpsn.csv*, *qip-bootpobject.fmt*, *qip-bootpobject.csv* and *qip-bootptabcsv.log*

7. Use **qip-setobject** to import the data. Refer to **qip-setobject** for details.

END OF STEPS

qip-bootptabcsv output files

Here are some sample file layouts for the output files created by **qip-bootptabcsv**. Each field is separated by a comma. For a full description of the file formats, refer to "The Import Utilities" section.

Output File 1 - qip-bootpobject.fmt

The *qip-bootpobject.fmt* file is the file that contains the object field names (not the values), which are then imported using **qip-setobject**. Some of the values for this file are derived using the Bootptab file tags. More "Optional" fields, which can be imported, are available. See Chapter 4 for more information.

Example

```
ObjectAddress,ObjectName,DomainName,MACAddress,ObjectClass,TftpServer,
   DefaultRouters,DynamicConfig,HardwareType,BootFileName
```

Table 3-10 Output file definitions

Value in the record	Field name	Derived from	Bootptab file tag	Comments
ObjectAddress	ObjectAddress	Bootptab file	ip	Required
ObjectName	Object Name	Bootptab file		Required
DomainName	DomainName	Bootptab file	dn or the **-d** parameter of the **qip-bootpcsv** process	Required
MACAddress	MACAddress	Bootptab file	ha	Required
ObjectClass	ObjectClass	Bootptab file	ht	Optional
TftpServer	TftpServer	Bootptab file	ts	Optional
DefaultRouters	DefaultRouters	Bootptab file	gw	Optional
DynamicConfig	DynamicConfig			Optional. This is assumed to be Manual Bootp.
HardwareType	HardwareType	Bootptab file	ht	Optional
BootFileName	BootFileName	The CLI qip-bootptabcsv command	hd and bf	Required

Output File 2 - qip-bootpobject.csv

The *qip-bootpobject.csv* file is the file that contains the *values* of the object field names, which are then imported using **qip-setobject**. Table 3-11 only explains the first row in the file. Some of the values for this file are derived using the Bootptab file tags. More "Optional" fields, which can be imported, are available. Refer to Chapter 4, "Exporting and importing with CLIs" for more information.

Example

```
198.200.41.64,pl1cilas1,qtek.com,0800094b72fc,,,198.200.41.1,Manual_Bootp,et
   hernet,,
198.200.35.147,pl1bolas3,qtek.com,0800095dc634,,,198.200.35.1,Manual_Bootp,e
   thernet,,
198.200.35.146,pl1bolas2,qtek.com,0800095d0199,,,198.200.35.1,Manual_Bootp,e
   thernet,,
198.200.35.145,pl1bolas1,qtek.com,0800095d01d9,,,198.200.35.1,Manual_Bootp,e
   thernet,,
```

Table 3-11 Output file definitions

Value in the record	Field name	Derived from	Bootptab file tag	Comments
198.200.41.64	ObjectAddress	Bootptab file	ip	Required
pllcilas1	Object Name	Bootptab file		Required
qtek.com	DomainName	Bootptab file	dn	Required
0800094b72fc	MACAddress	Bootptab file	ha	Optional
	ObjectClass	Bootptab file	ht	Optional
	TftpServer	Bootptab file	ts	Optional
198.200.41.1	DefaultRouters	Bootptab file	gw	Optional
Manual_Bootp	DynamicConfig			Optional. This is assumed to be Manual Bootp.
ethernet	HardwareType	Bootptab file	ht	Optional
	BootFileName	The CLI qip-bootptabcsv command	hd and bf	Required

Output File 3 - qip-bootpnet.csv

The *qip-bootpnet.csv* file is the file that contains the network information, which are then imported using **enternetwork**. Table 3-12 explains the first row in a *qip-bootpnet.csv* file. There are more "Optional" fields, which can be imported. Refer to Chapter 4 for more information.

Example

```
161.251.0.0,24,root@unilever.com,net_161.251.0.0,Y
```

Table 3-12 Output file definitions

Record value	Field name	Derived from	Comments
161.251.0.0	Network Address	Bootptab file ip and sm bootptab file tags.	Required
24	Mask bit length	Bootptab file (sm tag)	Required
root@unilever.com	e-mail address	System-generated*	Required
net_161.251.0.0	Network Name	System-generated*	Optional
Y	Cidr indicator	System-derived	Optional

Output File 4 - qip-bootpsn.csv

The *qip-bootpsn.csv* file is the file that contains the subnet information, which is then imported by using **entersubnet**. Table 3-13 explains the first row in a *qip-bootpsn.csv* file. There are more "Optional" fields, which can be imported. Refer to Chapter 4 for more information.

Example

```
161.251.6.0,255.255.255.0,161.251.0.0,subnet_161.251.6.0,,,,,,,,,,qtek.com
```

Table 3-13 Output file definitions

Value in the record	Field name	Derived from	Comments
161.251.6.0	Subnet Address	Bootptab file ip and sm bootptab file tags.	Required
255.255.255.0	Subnet Mask	Bootptab file (sm tag)	Required
161.251.0.0	Network Address	System-generated	Required
subnet_161.251.6.0	Subnet Name	System-generated	Optional

Value in the record	Field name	Derived from	Comments
qtek.com	Domain Name	Bootptab file or the **-d** parameter of the **qip-bootpcsv** process	Optional

Note: The subnet address in the Network Address and Subnet Name fields is derived from the sm tag and the object IP address.

Importing Bootptab files

Network information (*qip-bootpnet.csv*) is imported into VitalQIP by using **enternetwork**. For more detailed information, refer to "enternetwork " (p. 4-28).

Subnet information (*qip-bootpsn.csv*) is imported into VitalQIP by using **entersubnet**. For more detailed information, refer to "entersubnet" (p. 4-52).

Manual Bootp object information (*qip-bootpobject.fmt* and *qipbootpobject.csv*) is imported into VitalQIP by using **entersimpleobj**. For more detailed information, refer to "entersimpleobj" (p. 4-47).

4 Exporting and importing with CLIs

Overview

VitalQIP offers commands to export and import data into or from the VitalQIP database. Commands are available to export data for domains, OSPF areas, subnet organizations, subnets, MAC address pools, or objects. Commands also allow the importation of data for domains, OSPF areas, subnet organizations, subnets, MAC address pools, or objects. These import/export commands are available on UNIX and Windows platforms and can be used for upgrades or new installations.

Note: In general, import/export commands should NOT be used to modify existing data. The qip-get*/qip-set* CLIs may be more appropriate.

Note: Alcatel-Lucent recommends that you test the format of the import of your files with just a few records before attempting an import of many objects.

Importing files

This section discusses the file formats required for data input using the Import function in the VitalQIP interface. You can import data for domains, OSPF areas, subnet organizations, subnets, MAC address pools, or objects.

The import file format is in CSV format. Throughout tables in this chapter, the following notation is used for input values.

M = Mandatory, O = Optional

> **Note:** If optional information (for example, domain, contact, location, servers) is entered as part of the data, it must already exist in the VitalQIP database.

Order of importing

Alcatel-Lucent recommends a logical order of importing data into your database.

1. Servers (**enterserver**)
2. Domains (**enterdomain**)
3. Networks (**enternetwork**)
4. Subnets (**entersubnet**)
5. Objects (**entersimpleobj**)

During the import, it is recommended that you redirect error messages to a file to trap the potential error messages issued by the VitalQIP, Sybase and/or Oracle database servers. For example, when you run **enterdomain** to import domain information, run:

```
enterdomain -f datafile.dat –e err.dat
```

enteraddrrange

enteraddrrange is the CLI command for defining managed address ranges for networks, and for defining managed object ranges within subnets. These ranges can then be assigned an administrator to manage them. The input must be in VitalQIP CSV (comma delimited) format, as described in Table 4-1, "Address range data in VitalQIP CSV format" (p. 4-3).

Synopsis

```
enteraddrrange -t address_type -f input_filename [-g loginserver]
   [-s servername] [-o organization] [-u username] [-p password]
   [-r reject_file] [-e errmsg_file]
```

Parameters

enteraddrrange recognizes the following parameters:

-t ***address_type***	Defines a range within a network or subnet. Options are **network** (define a range of subnets within a network) or **subnet** (define a range of objects within a subnet).
-f ***input_filename***	Specifies the directory and filename of the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-r ***reject_file***	Specifies the directory and filename for the rejected records.
-e ***errmsg_file***	Specifies the filename to which this CLI command writes error messages, if they occur. The default is *STDERR*.

Table 4-1 Address range data in VitalQIP CSV format

Field	Value type	Description
Network/Subnet IP Address	[M] Numeric with decimals (15 char)	Enter the IP address of the network or subnet (for example, 100.12.0.0, which must already be defined in VitalQIP).
First Address	[M] Numeric with decimals (15 char)	Enter the beginning IP address for this range. (for example, 100.12.1.1)

Field	Value type	Description
Last Address	[M] Numeric with decimals (15 char)	Enter the ending address for this range. (for example, 100.12.1.255)

Note: Data lines must end with a carriage return, or they cannot be imported.

Command line input examples

With the **-t** option as network:

```
144.144.0.0,144.144.144.100,144.144.144.150
144.144.0.0,144.144.144.151,144.144.144.200
144.144.0.0,144.144.144.201,144.144.144.250
```

With the **-t** option as subnet:

```
144.144.144.0,144.144.144.1,144.144.144.100
```

enteralias

enteralias is the CLI command for importing object aliases.

> **Note:** The object included in the alias csv must exist in the database for the alias to be imported. Use **entersimpleobj** or **qipbulkload** to import the object information first.

VitalQIP validates data to prevent CNAME conflicts in the same organization by default. If a CNAME conflict occurs when adding alias records via the enteralias CLI, an error message displays and the add fails. You can disable this feature by setting the Validate CNAME Records policy to False. See the Object Policies and Profiles section of the *VitalQIP User Guide* for more information on the policy. When you add aliases, validation checks on the alias data are made against the following:

- Object Names
- Domain Names
- Aliases
- Object Resource Records
- Domain Resource Records
- Reverse Zone Resource Records

Synopsis

```
enteralias -f input_filename [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-r reject_file]
   [-e errmsg_file]
```

Parameters

enteralias recognizes the following parameters:

-f ***input_filename***	Specifies the directory and filename of the input data. The format is described in Table 4-2.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the database server name. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the organization name.
-u ***username***	Specifies a VitalQIP username.
-p ***password***	Specifies a VitalQIP password.
-r ***reject_file***	Specifies the directory and filename for the rejected records.
-e ***errmsg_file***	Specifies the filename to which this CLI command writes error messages, if they occur. The default is *STDERR*.

Table 4-2 enteralias file format

Field	Value type	Description
Object IP address	Dotted decimals (up to 15 chars length)	The ip address of an existing object with which the aliases are associated.
alias	Alphanumeric (up to 92 chars)	Alias name, can be fully qualified. If more than one exists, each should be separated with a comma (the maximum line length is OS dependent.)

Note: All data lines must end with a carriage return, or they cannot be imported.

Command line input examples

```
199.232.138.33,dev1.qtek.com,dev2.qtek.com,dev3.qtek.com,dev4.qtek.com
199.232.138.99,tst1.qtek.com
```

entercontact

entercontact is the CLI command for importing contact information. The input must be in VitalQIP CSV (comma delimited) format.

Synopsis

```
entercontact -f input_filename [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-r reject_file]
   [-e errmsg_file]
```

Parameters

entercontact recognizes the following parameters:

-f ***input_filename***	Specifies the directory and filename of the input data. The format is described in Table 4-3.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-r ***reject_file***	Specifies the directory and filename for the rejected records.
-e ***errmsg_file***	Specifies the filename to which this CLI command writes error messages, if they occur. The default is *STDERR*.

Table 4-3 Contact file format

Field	Value type	Description
Contact Last Name	[M]	Last name of the contact (for example, Smith).
Contact First Name	[M]	First name of the contact (for example, John).
Contact e-mail address	[O]	Email address of the contact (for example, jsmith@quadritek.com).
Contact phone number	[O]	Contact's phone number (for example, 610-555-1212).
Contact pager number	[O]	Contact's pager number (for example, 610-722-1111).

Note: All data lines must end with a carriage return, or they are not imported.

Command line input examples

```
Smith,John,john.smith@qip.com,888-555-1212,888-800-8888
Alberts,Judy,judy.alberts@qip.com,888-555-1213,888-800-8889
Doe,John,john.doe@qip.com,888-555-1214,888-800-8890
Collins,Steven
Majors,Maureen,maureen.majors@qip.com
```

enterdnsobj

enterdnsobj is the CLI command for importing A, MX, HINFO, and CNAME information in DNS BIND format to a VitalQIP dataserver. An A record creates an object for defined subnets. The HINFO record fills in the **Object Description** field of the **Object** tab (in the Object Profile) for an existing object. The CNAME record fills in the **Aliases** tab for an existing object.

VitalQIP validates data to prevent CNAME conflicts in the same organization by default. If a CNAME conflict occurs when adding an object and alias record via the enterdnsobj CLI, an error message displays and the add fails. You can disable this feature by setting the Validate CNAME Records policy to False. See the Object Policies and Profiles section of the *VitalQIP User Guide* for more information on the policy. When you add objects and aliases, validation checks on the object hostname alias data are made against the following:

- Object Names
- Domain Names
- Aliases
- Object Resource Records
- Domain Resource Records
- Reverse Zone Resource Records

For details on importing specific record formats, refer to *DNS and BIND, 4th Edition* by Paul Albitz & Cricket Liu, published by O'Reilly & Associates. The input must be in space delimited format.

Note: The External Comment, External Timestamp and Tombstoned flags are only set on an external object type and cannot be manually set when adding an object. These flags can be added to resource records.

Synopsis

```
enterdnsobj -f input_filename [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] -d domain
   [-r reject_file] [-w] [-e errmsg_file]
```

Parameters

enterdnsobj recognizes the following parameters:

-f ***input_filename***	Specifies the directory and filename of the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.

-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-d *domain*	Specifies the domain name of the domain where you want these objects to be.
-r *reject_file*	Specifies the directory and filename for the rejected records.
-w	Overwrites with warning if the addresses already exist.
-e *errmsg_file*	Specifies the filename to which this CLI command writes error messages, if they occur. The default is *STDERR*.

Input file content example

```
;==========================================================================;
   Addresses and other host information for zone: quadritek.com
;*************
; A records
;*************
suppaix IN A 198.200.201.167
test4 0 IN A 198.200.138.7
;****************
; CNAME records
;****************
alias1 IN CNAME suppaix1.quadritek.com
full-alias.quadritek.com IN CNAME suppaix1.quadritek.com
;************************
; Other resource records
;************************
suppaix1.quadritek.com IN MX 20 test4.quadritek.com
suppaix1.quadritek.com IN HINFO AIX system
```

enterdnsrr

enterdnsrr is the CLI command for importing DNS resource record information in VitalQIP format. The associated objects, domain, or reverse zone must already exist. Also, the input data must be in VitalQIP CSV (comma delimited) format.

VitalQIP validates data to prevent CNAME conflicts in the same organization by default. If a CNAME conflict occurs, an error message displays, and you cannot add or modify the profile. You can disable the feature by setting the Validate CNAME Records policy to False. See the Object Policies and Profiles section of the *VitalQIP User Guide* for more information on the policy. When you add or modify records other than CNAME records, validation checks are made against:

- Object aliases
- Object resource records
- Domain resource records
- Reverse zone resource records

When you add or modify CNAME records, validation checks are made against:

- Object names
- Domain names
- Any Object Profile resource records
- Any Domain Profile resource records
- Any Reverse Zone resource records
- ENUM NAPTR resource records
- IPv6 node names
- Mail servers

Synopsis

```
enterdnsrr -t owner_type -f input_filename [-g loginserver]
   [-s servername] [-o organization] [-u username] [-p password]
   [-r reject_file] [-e errmsg_file] [-a]
```

Parameters

enterdnsrr recognizes the following parameters:

-t ***owner_type***	Specifies the owner type with which the resource records are associated; for example, an object, a domain, or a reverse zone.
-f ***input_filename***	Specifies the directory and filename of the input data. The format is described in Table 4-4.

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-r *reject_file*	Specifies the directory and filename for the rejected records.
-e *errmsg_file*	Takes the directory and filename provided in the command line to output error messages during the process. An example output would be "rec10: the subnet for object 144.144.144.10 does not exist!" The corresponding record #10 will be listed in the reject file as is. The default is *STDERR*.
-a	Creates a PTR record for an entered A record.

Table 4-4 DNS resource record file input file format

Field	Value type	Description
Owner IP/Name	[M] text	Owner of the Resource Record (for example, IP address, if associated with an object; domain name if associated with a domain; and the IP address and mask length if associated with a reverse zone).
Resource Record Owner	[M]text (up to 255 char)	The owner field of the resource record as defined in RFC 1035.
Resource Record Class	[O] text (6 char)	The type of network or software for each class of records. Options are: IN, CS, CH, HS.
Resource Record Type	[O]text (10 char)	The type for the resource record (for example, CNAME, SOA, NS, WKS, etc.)

Field	Value type	Description
Minimum Time to live	[O]numeric	Specifies the time interval (time to live, in seconds) that the resource record can be cached before the source of the information should again be consulted. A zero value is interpreted to mean that the resource record can only be used for the transaction in progress, and should not be cached. For example, SOA records are always distributed with a zero TTL to prohibit caching. Zero values can also be used for extremely volatile data. If no TTL is desired (for example, it never expires), then this field should be left blank. (Refer to RFC 1035).
Resource Record Text	[M]text (up to 255 char)	The <rdata> of the resource record, as defined in RFC 1035. **Note:** If the text field consists of double quotes, each double quote should be paired with another double quote, and the entire field should be enclosed in a set of double quotes. For example, the text field entry for **"N3"** is **""""N3""""**
Place in	[O] R or F	Forward or reverse zone. Enter R for Reverse zones or F for Forward zones. The default is F (Forward). Applicable for object resource records only.
Action Flag	[O] A or D	Add or delete the record. The default is Add.
Note: The following four fields are used when adding resource records to the domain or reverse zones *only*. These flags are ignored when adding a resource record to an object.		
External Flag	[O] 0 or 1	0 indicates it is not an EDUP record, 1 indicates that is is an EDUP record.
Tombstoned	[O] 0 or 1	1 indicates that this is a tombstoned external record. 0 indicates that it is an active record.
External Comment	[O] text (up to 255 characters)	A comment indicating the history of this external add.
External Timestamp	[O] numeric	Specifies the time in seconds since January 1, 1970 that an external DNS authority mentioned this record.

Note: All data lines must end with a carriage return, or they are not imported.

Input file format examples

- For ***owner_type*** OBJECT (two examples):

```
150.1.6.4,dns02.lu.com,IN,TXT,200,"""Location""Floor 3""",F,A
150.183.140.2,lab2,IN,A,,150.183.140.2,F,A,O,O,,
```

- For ***owner_type*** DOMAIN (two examples):

```
qtek.com,dns.usa.com,IN,TXT,200,"""Location""Floor 3""",F,A
plo.com,lab2,IN,A,,150.183.140.2,F,A
```

- For ***owner_type*** Reverse Zone (two examples):

```
150.246.40.0/24,us.test.com,IN,HINFO,,""""Apple Macintosh""""OS
   8.1, FileMaker Pro""",R,A
113.33.0.0/16,113.33.132.in-
   addr.arpa,IN,PTR,,g01c.qns.qtek.com,R,A
```

enterdomain

enterdomain is the CLI command for importing domain and DNS zone option information. The input must be in VitalQIP CSV (comma delimited) format.

> **Note:** The DNS server included in the domain csv must exist in the database for the domain and associated DNS zone options to be imported. Use **enterserver** to import the DNS server information first.

When you import domains using the enterdomain CLI, the order of the domains becomes critical for the delegation to work correctly. For example, if you import a domain called **`test.xxx.mil`** first and then the next domain you import is **`xxx.mil`**, **`xxx.mil`** will have a higher domn_id than **`test.xxx.mil`**. And if **`xtest.xxx.mil`** is delegated to another server, the server that is authoritative for **`xxx.mil`** should have NS records listed in the **`b.xxx.mil`**. The problem is that the NS records for **`test.xxx.mil`** are not there.

VitalQIP validates data to prevent CNAME conflicts in the same organization by default. If a CNAME conflict occurs when adding a domain name via the enterdomain CLI, an error message displays and the domain add fails. You can disable this feature by setting the Validate CNAME Records policy to False. See the Object Policies and Profiles section of the *VitalQIP User Guide* for more information on the policy. When you add domains, validation checks on the domain name are made against the following:

- Object Names
- Domain Names
- Aliases
- Object Resource Records
- Domain Resource Records
- Reverse Zone Resource Records

Synopsis

> **Note:** You must specify **`-df`** and an appropriate value.

`enterdomain -n` [***`folder_name]`*** **`-w`** [**`-g`** ***`loginserver]`*** [**`-s`** ***`dataserver`***] [**`-e`** ***`errorfile`***] [**`-u`** ***`username]`*** ***`[`***`-p` ***`password]`*** [**`-o`** ***`organization]`*** **`-i`** ***`input_file`*** **`-df`** ***`dataformat`***

Parameters

enterdomain recognizes the following parameters:

`-n` ***`folder_name`***	Specifies the folder name (full path) to which the domains in the input file (**-i** parameter) are assigned.
`-w`	Overwrite if the record already exists.

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***dataserver***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-i ***input_filename***	Specifies the directory and filename of the input data. The format is described in Table 4-5.
-df ***dataformat***	Specifies the input file format. Possible values are **c** for common header and **sh** for separate header. If **-df c** is specified, a common header is used for all the domains in the input file, and for all the domains, only those fields shown in the Domain Profile in the VitalQIP GUI are included in the input file. If **-df sh** is specified, a separate header is used for each domain and each domain is separated by a space line in the input file. The zone option fields will also be included in the input file. If any DNS server is associated with a domain, that DNS server information will also be included in the input file.
-e ***error_file***	Specifies the file that contains errors generated by this CLI.

Table 4-5 Domain and DNS Zone data file format

Field	Value type	Description
Domain Name	[M] text(190 chars)	Specifies a fully qualified domain name (for example, mydomain.qtek.com).
Refresh Time	[M] numeric (in seconds)	Dictates how often the secondary server should verify its data against the primary server.
Expire Time	[M] numeric (in seconds)	If the secondary server has been able to contact the primary server within the 'expire' period, the secondary server will shut down since its data will be out of date.
Retry Time	[M] numeric (in seconds)	Dictates the interval for attempting to refresh in the event that the primary server is unavailable.

Field	Value type	Description
Default TTL	[M] numeric (in seconds)	Defines the time interval for other servers to cache all resource records in the database file if the TTL is not defined at the object level.
Negative Cache TTL	[M] numeric (in seconds)	Defines the time interval in which to cache all negative responses from the name servers authoritative for that zone.
Zone e-mail address	[M] text (64 char)	Specifies the e-mail address where the errors caused by DNS are posted. If the e-mail address contains an @ character, the text following the @ character must contain at least one period separating two domain segments, such as quadritek.com. Each segment of the data, based on the period, following the @ character must contain only the character A-Z, a-z, 0-9, and the dash (-) character. The first and last characters of each segment cannot be the dash character. If the e-mail address does not contain an @ character, the entire field must adhere to the above validation. If the e-mail address contains an @ character, the data preceding the @ character does not need to be validated against any specific algorithm but must be at least one character in length. The e-mail address may contain a trailing period.
Start Zone Options	[O]	There is never any data in this field, it is a placeholder only. See "Input file layout" (p. 4-21) for details
Extensions	[O]	There is never any data in this field. It is a placeholder only. See "Input file layout" (p. 4-21) for details.
Prefix of zone db file	[O] text	Text is displayed before the domain in the zone file and is added to the domain zone file exactly as it is entered.
Postfix of zone db file	[O] text	Text appears after the domain in the zone file and is added to the domain zone file exactly as it is entered.
allow-query	[O] text	Specifies which hosts are allowed to ask questions of a resolver. The options are: Any, None, localhost, localnets, Use List, Use Server Value.

Field	Value type	Description
ACL Templates	[O] text	This field is only available when its parent parameter is set to “Use List”. It allows you to select a predefined ACL template in VitalQIP. **Note:** This field may appear in any Zone Options (BIND-9.X options, Lucent DNS 4.X options, or Windows 2000 DNS options) where the parent zone option value is set to Use List.
other	[O] text	This field is only available when its parent parameter is set to “Use List” and it allows you to enter free form text. **Note:** This field may appear in any Zone Options (BIND-9.X options, Lucent DNS 4.X options, or Windows 2000 DNS options) where the parent zone option value is set to Use List.
allow-transfer	[O] text	Specifies which hosts are allowed to receive zone transfers from this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-update	[O] text	Specifies which hosts are allowed to submit dynamic DNS updates to this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
check-names	[O] text	Specifies the type of name verification processing. The options are: Warn, Fail, Ignore, Use Server Value.
notify	[O] text	Specifies whether to perform notification. The options are: No, Yes, Use Server Value.
zone block of named.conf	[O] text	Any text typed in this free text field will appear in the zone blocks of the *named.conf* file.
BIND-9.X Options	[O]	There is never any data in this field, it is a placeholder only. See “Input file layout” (p. 4-21) for details.
allow-notify	[O] text	Specifies whether to accept notification messages from name servers other than the configured master name server for a zone. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-query	[O] text	Specifies which hosts are allowed to ask questions of a resolver. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-transfer	[O] text	Specifies which hosts are allowed to receive zone transfers from this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.

Field	Value type	Description
allow-update	[O] text	Specifies which hosts are allowed to submit dynamic DNS updates to this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
notify	[O] text	Specifies whether to perform notification. The options are: No, Yes, Explicit, Use Server Value.
zone block of named.conf	[O] text	Any text typed in this free text field will appear in the zone blocks of the *named.conf* file.
Import External Updates	[O] text	Specifies whether this domain is EDUP enabled. The options are true or false.
allow-query	[O] text	Specifies which hosts are allowed to ask questions of a resolver. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-transfer	[O] text	Specifies which hosts are allowed to receive zone transfers from this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-update	[O] text	Specifies which hosts are allowed to submit dynamic DNS updates to this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
check-names	[O] text	Specifies the type of name verification processing. The options are: Warn, Fail, Ignore, Use Server Value.
notify	[O] text	Specifies whether to perform notification. The options are: No, Yes, Use Server Value.
zone block of named.conf	[O] text	Any text typed in this free text field will appear in the zone blocks of the *named.conf* file.
LUCENT DNS 4.X Options	[O]	There is never any data in this field. It is a placeholder only. See "Input file layout" (p. 4-21) for details
Import External Updates	[O] text	Specifies whether this domain is EDUP enabled. The options are true or false.
allow-notify	[O] text	Specifies whether to accept notification messages from name servers other than the configured master name server for a zone. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-query	[O] text	Specifies which hosts are allowed to ask questions of a resolver. The options are: Any, None, localhost, localnets, Use List, Use Server Value.

Field	Value type	Description
allow-transfer	[O] text	Specifies which hosts are allowed to receive zone transfers from this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-update	[O] text	Specifies which hosts are allowed to submit dynamic DNS updates to this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
notify	[O] text	Specifies whether to perform notification. The options are: No, Yes, Explicit, Use Server Value.
zone block of named.conf	[O] text	Any text typed in this free text field will appear in the zone blocks of the *named.conf* file.
WINDOWS 2000 DNS Options	[O]	There is never any data in this field, it is a placeholder only. See "Input file layout" (p. 4-21) for details.
aging	[O] text	Specifies whether the aging and scavenging are enabled for this zone. The options are: false, true, Use Server Value.
allow-transfer	[O] text	Specifies which hosts are allowed to receive zone transfers from this server. The options are: Any, None, Name Servers Only, Use List, Use Server Value.
allow-update	[O] text	Specifies which hosts are allowed to submit dynamic DNS updates to this server. The options are: no, yes, Use Server Value.
no-refresh-interval	[O] numeric (in hours)	Specifies the no refresh interval for scavenging. If the value is set to 0, the value will not be pushed.
notify	[O] text	Specifies whether to perform notification. The options are: No, Yes, Use List, Use Server Value.
refresh-interval	[O] numeric (in hours)	Specifies the refresh interval for scavenging. If the value is set to 0, the value will not be pushed.
zone-options	[O] text	This is free form text where information can be entered in dnscmd format.
End Zone Options	[O]	There is never any data in this field. It is a placeholder only. See "Input file layout" (p. 4-21) for details.
Start DNS Servers	[O]	There is never any data in this field. It is a placeholder only. See "Input file layout" (p. 4-21) for details.
Primary DNS Server Name	[O] text	Specifies the fully qualified name of a primary DNS server for this domain.

Field	Value type	Description
DNS Server Name	[O] text (253 char)	Specifies the fully qualified name of an authoritative DNS server for this domain.
DNS Server Type	[O] text	Specifies whether the server is a primary or a secondary server for the zone. The Options are: P, S.
Sec DNS Update Flag	[M] numeric	Specifies whether the name server is secure and if it is secure, then the dynamic updates to the server for this zone must be secure. The options are: 1 , 0.
End DNS Servers	[O]	There is never any data in this field. It is a placeholder only. See "Input file layout" (p. 4-21) for details.

Input file layout

The input file for **enterdomain** is data driven. If any Zone Options are specified in the input file, the 'Start Zone Options' and the 'End Zone Options' placeholder columns are required. The fields between these tags may vary based on the setting of other fields.

For example, the 'ACL Templates' and 'other' fields are only applicable if their parent's value (for example, the option 'allow query') is set to a value of 'Use List'. Consequently, if an option is to be set, it should come before the child fields or an error will be generated. If you want any Prefix/Postfix of zone db files, the 'Extensions' column must be placed before its prefix and postfix of the zone db file field.

Similarly, if you want to enter any BIND-9.X Options, the 'BIND-9.X Options' placeholder column must be placed before any of its actual option columns. The same rule applies to any LUCENT DNS 4.X Options and WINDOWS 2000 DNS Options.

If any DNS server is associated with a domain, the 'Start DNS Servers'and the 'End DNS Servers' columns are required (again, to act as placeholders). Additionally, if any DNS server is associated with a domain, the server type of the first server has to be a primary server.

Note: All data lines must end with a carriage return, or they are not imported.

Command line input example

```
enterdomain -n boston -i domn.txt -df c
```

Note: A correct input file format is very important to ensure the **enterdomain** CLI runs successfully. Users should run **exportdomain** to get the correct input file format and modify the input file data as needed before running **enterdomain**. You can get both input file formats from **exportdomain** whether or not you use the **-sh** parameter in the command line. Refer to "exportdomain" (p. 4-80) for further information.

enterlocalobj

enterlocalobj is the CLI command for importing object information in Local format. VitalQIP automatically creates appropriate object records in the specified domain and in the appropriate subnets. The object class of these objects is set to undefined. The format of the input file (specified by the **-f** parameter) is the same as that of */etc/hosts*, as shown below.

Note: The input file is space-delimited, not comma-delimited.

VitalQIP validates data to prevent CNAME conflicts in the same organization by default. If a CNAME conflict occurs when adding an object and alias data via the enterlocalobj CLI, an error message displays and the object add will fail. You can disable this feature by setting the Validate CNAME Records policy to False. See the Object Policies and Profiles section of the *VitalQIP User Guide* for more information on the policy. When you add an object and alias data, validation checks on the object name and aliases are made against the following:

- Object Names
- Domain Names
- Aliases
- Object Resource Records
- Domain Resource Records
- Reverse Zone Resource Records

Before you begin

- All data lines must end with a carriage return, or they cannot be imported.
- Within the VitalQIP GUI, the default address option for importing an object is "Overwrite if address exists". **enterlocalobj** defaults to a warning that duplicates exist, and then skip that address.
- You can add multiple Alias Names. Continue to add Alias names with spaces (or tabs) between them.

Synopsis

```
enterlocalobj -f input_filename -d domain_name [-g loginserver]
   [-s servername] [-o organization] [-u username] [-p password]
   [-r reject_file] [-e errmsg_file] [-w]
```

Parameters

enterlocalobj recognizes the following parameters:

-f *input_filename* Specifies the directory and filename of the input data. The format is described in Table 4-6. Note that host files and their directory are used by **enterlocalobj** to input data (for example, **-f /etc/hosts**). If other files are used, an error message is displayed.

-d *domain_name* Specifies the domain name belonging to the object.

-g *loginserver* Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.

-s *servername* Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.

-o *organization* Specifies the VitalQIP organization (corporation) name.

-u *username* Specifies the VitalQIP administrator account to be used in establishing the database connection.

-p *password* Specifies the password for the associated administrator account.

-r *reject_file* Specifies the directory and filename for the rejected records.

-e *errmsg_file* Specifies the filename to which this CLI command writes error messages, if they occur. The default is *STDERR*.

-w Overwrites with warning if the addresses already exist.

Table 4-6 Data in local file format

Field	Value type	Description
IP Address	[M]	The IP address of the object.
Host Name	[M]	The host name of the object.
Alias Name1	[O]	The alias of the object. If a fully qualified hostname is supplied, the parsing rule will be based on the policy setting of the ALLOW_DOTTED_HOSTNAME policy in the Global Policies portion of the GUI.
Alias Name2	[O]	An additional alias of the object.
Alias Name3	[O]	An additional alias of the object.

Input file example

```
199.16.12.4 ws00050sw johnsws domnws
```

enterlocation

enterlocation is the CLI command for importing location information. The input must be in VitalQIP CSV (comma delimited) format.

Synopsis

```
enterlocation -f input_filename [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-r reject_file]
   [-e errmsg_file] [-h] [-v]
```

Parameters

enterlocation recognizes the following parameters:

-f ***input_filename***	Specifies the directory and filename of the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-r ***reject_file***	Specifies the directory and filename for the rejected records.
-e ***errmsg_file***	Specifies the filename to which this CLI command writes error messages (if any). The default is *STDERR*.

Table 4-7 Location file format

Field	Value type	Description
Room ID or Address 1	[M]	The first line of the address of the location. (for example, 400 Lapp Road).
Address 2	[O]	The second line of the address of the location (for example, Suite 20).
City	[O]	The city of the location (for example, Malvern).
State	[O]	The state of the location (for example, PA).
Zip	[O]	The zip code for the location (for example, 19355).
Country	[O]	The country of the location (for example, US).

Note: All data lines must end with a carriage return, or they are not imported.

Input file examples

```
400 Lapp Road,Suite 20,Malvern,PA,19355,US
1 Sycamore Lane,Mailstop 123245AS23,Malvern,PA,19355
```

entermanufacturer

entermanufacturer is the CLI command for importing manufacturer information to the VitalQIP database. The input must be in VitalQIP CSV (comma delimited) format.

Synopsis

```
entermanufacturer -f input_filename [-g loginserver]
   [-s servername] [-o organization] [-u username]
   [-p password][-r reject_file] [-e errmsg_file] [-h] [-v]
```

Parameters

entermanufacturer recognizes the following parameters:

-f ***input_filename***	Specifies the directory and filename of the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-r ***reject_file***	Specifies the directory and filename for the rejected records.
-e ***errmsg_file***	Specifies the filename to which this CLI command writes error messages if they occur. The default is *STDERR*.

Table 4-8 Manufacturer data file format

Field	Value type	Description
Manufacturer Name	[M] Text (up to 48 char)	Enter the manufacturer name of the model being defined, it must already exist in the VitalQIP database for a given object class.
Object Class Description	[M] Text (20 char)	Object Class.
Preferred Prefix	[M] Hexadecimal (6 char)	The prefix of the manufacture's MAC address.
Model Name	[O] Text (20 char)	The model name. (If specified, then it must be followed by the tag name and value.)
Tag Name	[O] Text (30 char)	Model Tag name.

Field	Value type	Description
Tag Value	[O]	The value associated with the Tag Name. (You can have multiple pairs of Tag Name and Tag Value for a given model.)

Note: All data lines must end with a carriage return, or they are not imported.

Input file examples

```
3Com,Workstation,001122,mod1,mtag1,tagval1,mtag2,tagval2
3Com,Workstation,001122,mod2,mtag21,tagval21,mtag22,tagv22
```

enternetwork

enternetwork is the CLI command for importing network information. The input must be in VitalQIP CSV (comma delimited) format.

Synopsis

```
enternetwork -f input_filename [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-r reject_file]
   [-e errmsg_file] [-v] [-h]
```

Parameters

enternetwork recognizes the following parameters:

-f *input_filename* Specifies the directory and filename of the input data.

-g *loginserver* Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.

-s *servername* Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.

-o *organization* Specifies the VitalQIP organization (corporation) name.

-u *username* Specifies the VitalQIP administrator account to be used in establishing the database connection.

-p *password* Specifies the password for the associated administrator account.

-r *reject_file* Specifies the directory and filename for the rejected records.

-e *errmsg_file* Specifies the filename to which this CLI command writes error messages if they occur. The default is *STDERR*.

Table 4-9 Network data file format

Field	Value type	Description
Network Address	[M]numeric with decimals (15 char)	The network address you are assigning subnets to (for example, 100.12.128.0).
Network mask length	[M] numeric (up to 32 characters)	The number of bits in the network portion of the address.
E-mail Address	[M]text (63 characters)	The e-mail address for the reverse zone.
Network Name	[O] text (32 characters)	A name signifying the network you are building.
CIDR	[O] text (1 characters)	Whether to support CIDR (Y = yes, N= no).

Field	Value type	Description
Warning Percent	[O] numeric	A warning is posted if the percentage of allocated IP addresses is greater than the percentage defined in this field. Values are 0-100.
Warning Type	[O] numeric	The type of warnings to receive: 0 = none 1 = e-mail 2 = visual 3 = e-mail & visual
DNS Server Name	[O] text (253 characters)	The fully qualified server name authoritative for the reverse zone you created.

Note: All data lines must end with a carriage return, or they are not imported.

Input file examples

```
172.17.0.0,16,root@nsroot.qip.com,QIPNET-1,N,10,2,nsroot.qip.com
172.18.0.0,16,root@nsroot.qip.com,QIPNET-2,N,10,2,nsroot.qip.com
172.19.0.0,16,root@nsroot.qip.com,QIPNET-3,N,10,2,nsroot.qip.com
192.168.0.0,16,root@nsroot.qip.com,QIP-CIDR-NET-4,Y,10,2,nsroot.qip.com
10.0.0.0,8,root@nsroot.qip.com,QIPNET-5,N,O,O
```

enterorganization

enterorganization is the CLI command for importing organization information. The input must be in VitalQIP CSV (comma delimited) format. This command inputs the maximum object count.

Synopsis

```
enterorganization -f input_filename [-g loginserver]
   [-s servername] [-u username] [-p password] [-r reject_file]
   [-e errmsg_file]
```

Parameters

enterorganization recognizes the following parameters:

-f *input_filename* Specifies the directory and filename of the input data. The format is described in Table 4-10.

-g *loginserver* Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.

-s *servername* Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.

-u *username* Specifies the VitalQIP administrator account to be used in establishing the database connection.

-p *password* Specifies the password for the associated administrator account.

-r *reject_file* Specifies the directory and filename for the rejected records.

-e *errmsg_file* Specifies the filename to which this CLI command writes error messages if they occur. The default is *STDERR*.

Table 4-10 Organization data file format

Field	Value type	Description
Organization ID	[M]numeric	The Organization ID is exported by *exportorganization* for reference purposes. Therefore, *enterorganization* expects an Organization ID. However, the Organization ID is not used by *enterorganization* because of potential conflicts with previously existing Organization IDs in the database. When you manually create the datafiles, therefore, you can enter any number for the Organization ID.

Field	Value type	Description
Organization Name	[M]text (32 char.)	The Organization name.
Description	[O]text (255 char.)	The description of the organization.
Maximum Number of Objects Allowed	[M]numeric	Indicates the maximum number of objects allowed in the organization. Zero indicates that there is no limit.

Note: All data lines must end with a carriage return, or they cannot be imported.

Input file examples

```
2,Lucent,”Description of the Lucent Corporation”,0
3,VitalQIP,”Description of the VitalQIP Organization”,10000
```

enterospf

enterospf is the CLI command for importing OSPF information. The input must be in VitalQIP CSV (comma delimited) format.

Synopsis

```
enterospf -f input_filename [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-r reject_file]
   [-e errmsg_file] [-w]
```

Parameters

enterospf recognizes the following parameters:

-f ***input_filename***	Specifies the directory and filename of the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-r ***reject_file***	Specifies the directory and filename for the rejected records.
-e ***errmsg_file***	Specifies the filename to which this CLI command writes error messages if they occur. The default is *STDERR*.
-w	Overwrites warning messages.

Table 4-11 OSPF area data file format

Field	Value Type	Description
OSPF Area Name	[M]text (32 char)	The unique name applied to this OSPF area (for example, OSPF Area 1).
OSPF Area ID	[O] numeric, with decimals (15 char)	A unique ID that represents the backbone and has already been established (for example, 100.12.128.0).
OSPF Compliant	[O]text (1 char)	Whether OSPF Compliant (1=yes, 2=no). The default is 0.

Field	Value Type	Description
Area Warning Percent	[O] numeric,0-100	The threshold for area warnings. Delivers an alarm when the number of managed addresses put into service for this OSPF area reaches the defined threshold.
Area Warning Type	[O] numeric, 0-3	The type of warnings to receive: 0 = none 1 = e-mail 2 = visual 3 = e-mail & visual
Managed Subnets	[O]	List of the managed subnet IP address surrounded by double quotes.
Start Subnet Range	[O] numeric, with decimals	The start address of the subnet for this OSPF area (for example, 100.12.128.0).
OSPF Area Mask or End Subnet Range	[O]numeric, with decimals (15 char)	The above Start Subnet Range field must be coupled with either an OSPF Area Mask or an End Subnet Range. There can be multiple sets of these pairs listed with each record.

Note: All data lines must end with a carriage return, or they are not imported.

Input file examples

- The following example is based on using the Start Subnet Range and End Subnet Range pairs:

```
OSPF-
   In1,0.0.0.2,N,85,3,"172.16.64.0,172.16.128.0",172.16.64.1,172.16.64.255,
   172.16.128.1,172.16.128.255
```

- The following example is based on using the Start Subnet Range and OSPF Area Mask pairs:

```
OSPF-In2,0.0.0.2,Y,80,2,"172.16.192.0",172.16.192.0,255.255.192.0
```

enterreversezone

enterreversezone is the CLI command for importing reverse zones (split zones) information to the VitalQIP database. The input must be in VitalQIP CSV (comma delimited) format.

Synopsis

```
enterreversezone -df dataformat -i input_filename [-bps]
   [-g loginserver] [-s servername] [-o organization]
   [-u username] [-p password] -w
```

Parameters

enterreversezone recognizes the following parameters:

-df ***dataformat***	Specifies the input file format: If **-df c** is specified, then a common header is used for all the reverse zones in the input file, and for all the reverse zones, only those fields shown in the reverse zone profile window in VitalQIP GUI will be included in the input file. If **-df sh** is specified, then a separate header is used for each reverse zone and each reverse zone record is separated by a space line in the input file. All of the zone option fields will be included in the input file. If any DNS server is associated with the reverse zones, that DNS server information will also be included in the input file.
-i ***input_filename***	Specifies the directory and filename of the input data. Each field in the input file is described in Table 4-12.
-**bps**	Specifies that for each reverse zone specified in the input file, its associated zone options and DNS server list will be inherited from its parent reversezone. **Note:** When **-bps** is specified in the command line, the input file must be in "common headers" format, that is, **-df c** must be specified in the command line.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.

-w Overwrites an existing reverse zone.

Table 4-12 Reverse zone data file format

Field	Value type	Description
Reversezone IP Address	[M] dotted decimal	The IP address of the reverse zone.
Reversezone Address Mask	[M] numeric	Number of Mask bits for the reverse zone IP address.
Network Address	[M] dotted decimal	The IP address of the network the zone is on.
Parent Address	[M] dotted decimal	The IP address of the parent zone to split.
Parent Address Mask	[M] numeric	Number of Mask bits for the parent zone's address.
Refresh Time	M] numeric (in seconds)	Dictates how often the secondary server should verify its data against the primary server.
Expire Time	[M] numeric (in seconds)	If the secondary server has been able to contact the primary server within the 'expire' period, the secondary server will shut down since its data will be out of date.
Retry Time	[M] numeric (in seconds)	Dictates the interval for attempting to refresh in the event that the primary server is unavailable.
Default TTL	[M] numeric (in seconds)	Defines the time interval for other servers to cache all resource records in the database file if the TTL is not defined at the object level.
Negative Cache TTL	[M] numeric (in seconds)	Defines the time interval in which to cache all negative responses from the name servers authoritative for that zone.

Field	Value type	Description
Zone e-mail address	[M] text (62 char)	Specifies the e-mail address where the errors caused by DNS are posted. If the e-mail address contains an @ character, the text following the @ character must contain at least one period separating two domain segments, such as quadritek.com. Each segment of the data, based on the period, following the @ character must contain only the character A-Z, a-z, 0-9, and the dash (-) character. The first and last characters of each segment cannot be the dash character. If the e-mail address does not contain an @ character, the entire field must adhere to the above validation. If the e-mail address contains an @ character, the data preceding the @ character does not need to be validated against any specific algorithm but must be at least one character in length. The e-mail address may contain a trailing period.
Start Zone Options	[O]	There is never any data in this field, it is a placeholder only. See Notes * below for details
Extensions	[O]	There is never any data in this field. It is a placeholder only. See Notes * for details.
Prefix of zone db file	[O] text	Text appears before the domain in the zone file and is added to the domain zone file exactly as it is entered.
Postfix of zone db file	[O] text	Text appears after the domain in the zone file and is added to the domain zone file exactly as it is entered.
Bind-8.x Options	[O]	There is never any data in this field. It is a placeholder only. See Notes * for details.
allow-query	[O] text	Specifies which hosts are allowed to ask questions of a resolver. The options are: Any, None, localhost, localnets, Use List, Use Server Value.

Field	Value type	Description
ACL Templates	[O] text	This field is only available when its parent parameter is set to “Use List” and it allows you to select a predefined ACL template in VitalQIP. **Note:** This field may appear in any Zone Options (for example, BIND-9.X options, Lucent DNS 4.X options, or Windows 2000 DNS options) where the parent zone option value is set to Use List.
other	[O] text	This field is only available when its parent parameter is set to “Use List” and it allows you to enter free form text. **Note:** This field may appear in any Zone Options (for example, BIND-9.X options, Lucent DNS 4.X options, or Windows 2000 DNS options) where the parent zone option value is set to Use List.
allow-transfer	[O] text	Specifies which hosts are allowed to receive zone transfers from this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-update	[O] text	Specifies which hosts are allowed to submit dynamic DNS updates to this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
check-names	[O] text	Specifies the type of name verification processing. The options are: Warn, Fail, Ignore, Use Server Value.
notify	[O] text	Specifies whether to perform notification. The options are: No, Yes, Use Server Value.
zone block of named.conf	[O] text	Any text typed in this free text field will appear in the zone blocks of the *named.conf* file.
BIND-9.X Options	[O]	There is never any data in this field, it is a placeholder only. See Notes* for details.

Field	Value type	Description
allow-notify	[O] text	Specifies whether to accept notification messages from name servers other than the configured master name server for a zone. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-query	[O] text	Specifies which hosts are allowed to ask questions of a resolver. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-transfer	[O] text	Specifies which hosts are allowed to receive zone transfers from this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-update	[O] text	Specifies which hosts are allowed to submit dynamic DNS updates to this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value
notify	[O] text	Specifies whether to perform notification. The options are: No, Yes, Explicit, Use Server Value.
zone block of named.conf	[O] text	Any text typed in this free text field will appear in the zone blocks of the named.conf file.
Import External Updates	[O] text	Specifies whether this domain is EDUP enabled. The options are true or false.
allow-query	[O] text	Specifies which hosts are allowed to ask questions of a resolver. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-transfer	[O] text	Specifies which hosts are allowed to receive zone transfers from this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-update	[O] text	Specifies which hosts are allowed to submit dynamic DNS updates to this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.

Field	Value type	Description
check-names	[O] text	Specifies the type of name verification processing. The options are: Warn, Fail, Ignore, Use Server Value.
notify	[O] text	Specifies whether to perform notification. The options are: No, Yes, Use Server Value.
zone block of named.conf	[O] text	Any text typed in this free text field will appear in the zone blocks of the named.conf file.
LUCENT DNS 4.X Options	[O]	There is never any data in this field. It is a placeholder only. See Noted * below for details
Import External Updates	[O] text	Specifies whether this domain is EDUP enabled. The options are true or false.
allow-notify	[O] text	Specifies whether to accept notification messages from name servers other than the configured master name server for a zone. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-query	[O] text	Specifies which hosts are allowed to ask questions of a resolver. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-transfer	[O] text	Specifies which hosts are allowed to receive zone transfers from this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
allow-update	[O] text	Specifies which hosts are allowed to submit dynamic DNS updates to this server. The options are: Any, None, localhost, localnets, Use List, Use Server Value.
notify	[O] text	Specifies whether to perform notification. The options are: No, Yes, Explicit, Use Server Value.
zone block of named.conf	[O] text	Any text typed in this free text field will appear in the zone blocks of the named.conf file.

Field	Value type	Description
WINDOWS 2000 DNS Options	[O]	There is never any data in this field, it is a placeholder only. See Notes * below for details.
aging	[O] text	Specifies whether the aging and scavenging are enabled for this zone. The Options are: false, true, Use Server Value.
allow-transfer	[O] text	Specifies which hosts are allowed to receive zone transfers from this server. The options are: Any, None, Name Servers Only, Use List, Use Server Value.
allow-update	[O] text	Specifies which hosts are allowed to submit dynamic DNS updates to this server. The options are: no, yes, Use Server Value.
no-refresh-interval	[O] numeric (in hours)	Specifies the no refresh interval for scavenging. If the value is set to 0, the value will not be pushed.
notify	[O] text	Specifies whether to perform notification. The options are: No, Yes, Use List, Use Server Value.
refresh-interval	[O] numeric (in hours)	Specifies the refresh interval for scavenging. If the value is set to 0, the value will not be pushed.
zone-options	[O] text	This is free form text where information can be entered in **`dnscmd`** format.
End Zone Options	[O]	There is never any data in this field. It is a placeholder only. See Notes * below for details.
Start DNS Servers	[O]	There is never any data in this field. It is a placeholder only. See Notes * below for details.
Primary DNS Server Name	[O] text	Specifies the fully qualified name of a primary DNS server for this domain.
DNS Server Name	[O] text (253 char)	Specifies the fully qualified name of an authoritative DNS server for this domain.
DNS Server Type	[O] text	Specifies whether the server is a primary or a secondary server for the zone. The Options are: P, S.

Field	Value type	Description
Sec DNS Update Flag	[M] numeric	Specifies whether the name server is secure and if it is secure, then the dynamic updates to the server for this zone must be secure. The options are: 1, 0.
End DNS Servers	[O]	There is never any data in this field. It is a placeholder only. See Notes * below for details.

Input file layout

The input file for **enterreversezone** is data driven. If any Zone Options are specified in the input file, the 'Start Zone Options' and the 'End Zone Options' placeholder columns are required. The fields between these tags may vary based on the setting of other fields.

For example, the 'ACL Templates' and 'other' fields are only applicable if their parent's value (for example, the option 'allow query') is set to a value of 'Use List'. Consequently, if an option is to be set, it should come before the child fields or an error will be generated. If you want any Prefix/Postfix of zone db files, the 'Extensions' column must be placed before its prefix and postfix of the zone db file field.

Similarly, if you want to enter any BIND-9.X Options, the 'BIND-9.X Options' placeholder column must be placed before any of its actual option columns. The same rule applies to any LUCENT DNS 4.X Options and WINDOWS 2000 DNS Options.

If any DNS server is associated with a domain, the 'Start DNS Servers'and the 'End DNS Servers' columns are required (again, to act as placeholders). Additionally, if any DNS server is associated with a domain, the server type of the first server has to be a primary server.

Note: All data lines must end with a carriage return or they are not imported.

enterserver

enterserver allows you to load one or more servers with their definition and parameters into the VitalQIP database.

Before you begin

- If the **enterserver** command is run with a server name that exists in the VitalQIP database and a zone e-mail address is not specified, no indication is given by the **enterserver** command that it does not exist. It continues processing without providing a notice that the zone e-mail address is missing. Ensure the zone e-mail address is specified when you are adding a server.
- The **enterserver** command modifies servers if the server name specified exists in the database. This functionality is different from the other import commands.

Synopsis

```
enterserver -f input_filename [-g loginserver] [-s servername]
   [-u username] [-p password] [-o organization] [-r reject_file]
   [-e errmsg_file]
```

Parameters

enterserver recognizes the following parameters:

-f ***input_filename***	Specifies the directory and filename of the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-r ***reject_file***	Specifies the directory and filename for the rejected records.
-e ***errmsg_file***	Specifies the filename to which this CLI command writes error messages if they occur. The default is *STDERR*.

enterserver parses the file until it finds the "Type =" label. When it finds this, it parses out the server type. Based upon the server type, it knows what parameters to expect.

The **Server Type** consists of one of the following :

DNS:

BIND-9.X

LUCENT DNS 4.X

MICROSOFT-NT 4.0

WINDOWS 2000 DNS

DHCP:

LUCENT DHCP 5.4

LUCENT DHCP 5.5

LUCENT DHCP 5.6

WINDOWS 2003 DHCP

Other Server Classes:

NIS

LOCAL HOST

Bootp

WINDOWS 2000 DC

The format consists of a record identifier "Format=" label. It is followed by a comma delimited list which indicates which parameters you are including in the CLI to send to the database. ***The first two fields must be the server name and domain name.***

Input file format

```
Type=<Server Type><CR>
Format=<Format><CR>
<Data Values><CR>
<Data Values><CR>
Type=<Server Type><CR>
Format=<Format><CR>
<Data Values><CR>
etc….
```

The **`Data Values`** are values you wish to submit to the server in the order of the parameter names listed in the **`Format`** line. If any of those values are incorrect, an error is generated and the CLI fails, for example:

```
Type=Bootp
Host Name, Domain Name, Default Directory,Time Interval
Hostname,quadritek.com,/abc/temp,4500
```

Every parameter and sub-parameter listed in the Server Profiles can be specified as a field in the format, except for multi-line fields such as “Additional Policies “ for DHCP and “Corporate Extensions” for DNS which must be specified with a pipe (|) symbol to denote new lines.

The Lucent DHCP parameters are described in Chapter 4 of the *VitalQIP User’s Guide* in a section titled “Lucent DHCP 5.4 Server Type”. The parameters for all other server Policies are also described in Chapter 4 of the *VitalQIP User’s Guide* .

Input Parameter Rules

Rules for the input of data are described following; also refer to the input examples.

- The first item must be the server name, and the second item must be the domain name, followed by the rest of the server parameters.
- The server parameter and its sub-parameters must be grouped together; parent followed by child, then by grandchild.
- If a parameter has a parent field with no value, the , , placeholder must be used.
- If the value consists of commas, then the value must be enclosed in a pair of double quotes.
- If a parameter consists of multiple lines , each line must be delimited by a | (pipe).

Example 1

The following example contains the format:

```
Type=LUCENT DHCP 5.5
Format=Host Name,Domain Name,Managed Range,Default Directory,DHCP
   Template,Accept Client Names,Additional Policies,Client Class,Debug
   Information,Debug,DebugFile,MaxDebugFileSize,Failover Server
   Type,CtlReqRetryMax,PollDelay,SyncBindRetryMax,SyncBindingBufSize,Use
   Failover
   Server,WaitCtlRetSecs,WaitSyncBindAckSecs,WaitSyncBindUpdateSecs,Remote
   Server Proxy,Description,Scheduled Automatic Updates,Support Bootp,Use
   Server Policy Template,DHCP Server
   Policies,AckRenewForUnusedAddress,ActiveLeaseExpiration,Bootfile,CheckTra
   nsactionID,ClientHostNameProcessing,CompressedLog,DefaultDescentThreshold
   ,DefaultLease,DefaultUnavailableThreshold,DropAllDhcpInformPackets,DropZe
   roMacAddressPackets,Enable LDRM
   Interface,UseLDRMCallout,ExpireAllLeasesOnRestart,ForceClass,HonorRequest
   edLeaseTime,HonorUnqualifiedBootfile,InitRebootAddressShuffle,IssueDropUn
   knownClientTrap,LeaseExpirationSleepTime,LeaveBootpParametersInOptions,Lo
   gLeaseGrantAndRenew,MacWarningsToEventLog,MaxOutgoingDhcpMessageSize,MaxP
   endingSeconds,MaxUnavailableTime,NackDhcpRequestsForDuplicates,NakUnknown
   Clients,OfferOnlyApiRequestedAddress,Option81Support,PadBootpReply,PingAt
   tempts,PingBeforeManualBootp,PingBeforeManualDhcp,PingDelay,PingRetention
   ,PingSendDelay,RegisteredClientsOnly,RenewAddressShuffle,RenewAddressShuf
   fleMaxCount,SearchDynamicFirst,SendRequestedParamsOnly,SendServerIdLast,S
   hareAutoBootpAndDynDhcp,SharedNetworkThresholdProcessing,SupportAutoRelea
```

```
se,SupportBootpAutoRelease,SupportEncodingLongOptions,SupportMultiUserCla
ss,SupportRelayAgentDeviceClass,SupportRelayAgentOption,SupportRelayAgent
ServerOverride,SupportSubnetSelection,ThresholdMonitorSleepTime,Update
QIP
Operations,Autorelease,Bootp,Decline,Delete,Expiration,Grant,Release,Rene
w,UpdatePreclusionDuration,ZeroCiAddr
lucdhcpprim,example.com,Corporation,/opt/qip/dhcp,"general",True,,,,None,dhc
pd.log,-
1,Standalone/Primary,3,60,3,1024,False,5,5,15,,,None,True,False,,False,Of
f,Default,False,Ignore,False,0,7776000,0,False,True,,False,True,None,True
,False,Off,False,60000,False,True,False,1024,10,86400,True,True,False,Sup
press,True,1,False,True,500,0,0,False,Off,0,False,False,False,False,False
,True,True,False,False,False,True,False,False,60,,True,True,True,True,Tru
e,True,True,True,60,False,
Format=Host Name,Domain Name,Managed Range,Default Directory,DHCP
Template,Accept Client Names,Additional Policies,Client Class,Debug
Information,Debug,DebugFile,MaxDebugFileSize,Failover Server
Type,PollRetryMax,SyncBindRetryMax,SyncBindingBufSize,SyncBindings,SyncFa
ilCritical,SyncReqRetryMax,WaitPollRplSecs,WaitSyncBindAckSecs,WaitSyncBi
ndUpdateSecs,WaitSyncStartSecs,Remote Server Proxy,Description,Scheduled
Automatic Updates,Support Bootp,Use Server Policy Template,DHCP Server
Policies,AckRenewForUnusedAddress,ActiveLeaseExpiration,Bootfile,CheckTra
nsactionID,ClientHostNameProcessing,CompressedLog,DefaultDescentThreshold
,DefaultLease,DefaultUnavailableThreshold,DropAllDhcpInformPackets,DropZe
roMacAddressPackets,Enable LDRM
Interface,UseLDRMCallout,ExpireAllLeasesOnRestart,ForceClass,HonorRequest
edLeaseTime,HonorUnqualifiedBootfile,InitRebootAddressShuffle,IssueDropUn
knownClientTrap,LeaseExpirationSleepTime,LeaveBootpParametersInOptions,Lo
gLeaseGrantAndRenew,MacWarningsToEventLog,MaxOutgoingDhcpMessageSize,MaxP
endingSeconds,MaxUnavailableTime,NackDhcpRequestsForDuplicates,NakUnknown
Clients,OfferOnlyApiRequestedAddress,Option81Support,PadBootpReply,PingAt
tempts,PingBeforeManualBootp,PingBeforeManualDhcp,PingDelay,PingRetention
,PingSendDelay,RegisteredClientsOnly,RenewAddressShuffle,RenewAddressShuf
fleMaxCount,SearchDynamicFirst,SendRequestedParamsOnly,SendServerIdLast,S
hareAutoBootpAndDynDhcp,SharedNetworkThresholdProcessing,SupportAutoRelea
se,SupportBootpAutoRelease,SupportEncodingLongOptions,SupportMultiUserCla
ss,SupportRelayAgentDeviceClass,SupportRelayAgentOption,SupportRelayAgent
ServerOverride,SupportSubnetSelection,ThresholdMonitorSleepTime,Update
QIP
Operations,Autorelease,Bootp,Decline,Delete,Expiration,Grant,Release,Rene
w,UpdatePreclusionDuration,ZeroCiAddr
lucdhcpsec,example.com,Corporation,/opt/qip/dhcp,"general",True,,,,None,dhcp
d.log,-
1,Failover/Secondary,3,3,1024,True,False,3,5,5,15,5,,,None,True,False,,Fa
lse,Off,Default,False,Ignore,False,0,7776000,0,False,True,,False,True,Non
e,True,False,Off,False,60000,False,True,False,1024,10,86400,True,True,Fal
se,Suppress,True,1,False,True,500,0,0,False,Off,0,False,False,False,False
,False,True,True,False,False,False,True,False,False,60,,True,True,True,Tr
ue,True,True,True,True,60,False,
```

Example 2

The following example contains the format:

```
Type=LUCENT DHCP 5.4
Format=Host Name,Domain Name,Managed Range,Default Directory,DHCP
   Template,Accept Client Names,Client Class,Failover Server Type,Use Server
   Policy Template,DHCP Server
   Policies,ActiveLeaseExpiration,Option81Support,
   ShareAutoBootpAndDynDhcp,SupportAutoRelease, PingDelay
```

The following example contains the corresponding values. Note the , , placeholder for Server Policies:

```
dhcpsvr,lucent.com,Corporation,d:/qip/dhcp,general,True,,Standalone/Primary,
   False,,Off,Suppress,True,False,0
```

Example 3

```
Type=WINDOWS 2000 DC
Format=Host Name,Domain Name,LDAP Port,Active Directory Domain Name,User
DN,Password,Scheduled Automatic Updates
dc,quadritek.com,389,quadritek.com,"cn=liz
zar,cn=users,dc=quadritek,dc=com",liz,None,
```

entersimpleobj

entersimpleobj is the CLI command for importing object information in VitalQIP format. The input must be in VitalQIP CSV (comma delimited) format.

> **Note:** For a more abbreviated import of objects in large numbers, refer to the **qipbulkload** CLI command.

VitalQIP validates data to prevent CNAME conflicts in the same organization by default. If a CNAME conflict occurs when adding an object and alias data via the entersimpleobj CLI, an error message displays and the object add will fail. You can disable this feature by setting the Validate CNAME Records policy to False. See the Object Policies and Profiles chapter of the *VitalQIP User Guide* for more information on the policy. When you add an object and alias data, validation checks on the object name and aliases are made against the following:

- Object Names
- Domain Names
- Aliases
- Object Resource Records
- Domain Resource Records
- Reverse Zone Resource Records

Synopsis

```
entersimpleobj -f input_filename [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-r reject_file]
   [-e errmsg_file] [-w(overwrite)] [-x] [-i]
```

Parameters

entersimpleobj recognizes the following parameters:

-f ***input_filename***	Specifies the directory and filename of the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-r ***reject_file***	Specifies the directory and filename for the rejected records.

-e *errmsg_file* Specifies the filename to which this CLI command writes error messages if they occur. The default is *STDERR*.

-w(overwrite) Overwrites duplicate addresses.

-i Overwrites GAP addresses.

-x Allows an IP address with a status of “Selected” or “Planned Use” to be overwritten. An address may be marked “Selected” as a result of the qip-getipaddr CLI being run (for example, this CLI is called with the Quick Add function in the Web Client).

Table 4-13 Object data in VitalQIP file format

Field	Value Type	Description
Object IP Address	[M]numeric, with decimals (15 char)	Enter the IP address of the object (for example, 100.12.128.28).
Object(Device) Name	[M]text (63 char)	Enter the object device name for this object. The policy setting ALLOW_DOTTED HOSTNAMES must be turned on if dotted names are used.
Subnet Address	[O]numeric, with decimals (15 char)	Enter the subnet address for this object (for example, 100.12.128.0).
DHCP Server Name	[O]text (253 char)	Enter the fully qualified name of the DHCP Server assigned for this object (for example, *dhcp1.quadritek.com*).
Domain Name	[O]text (190 char)	Enter the fully qualified domain name for this object (for example, *qtek.quadritek.com*).

Field	Value Type	Description
Object Class Description	[O]	refer to the following options: - Workstation - X-terminal - PC - Printer - Server - Wiring_HUB - Router - Bridge - Terminal_Server - Switch - Legacy_System - Gateway - Test_Equipment - Undefined - Others - External -Any user-defined object class
Comment	[O]text (32 char)	Enter a description for this object.
MAC Address	[O]12 or 16, in hexadecimal	Enter the MAC Address for this object.
Contact Last Name	[O]text (20 char)	Enter the last name of the contact for this object. If none exists, the Contact Last Name is created.
Contact First Name	[O]text (20 char)	Enter the first name of the contact for this object. If none exists, the Contact First Name is created.
Contact Phone No.	[O]text (20 char)	Enter the phone number of the contact for this object. If none exists, the Contact Phone Number is created.

Field	Value Type	Description
Dynamic Configuration	[O]numeric	Enter the dynamic configuration for this object. If you do not specify a value, the object is allocated as a static object. refer to the following options: 0=None - Allocates this address without using the Bootp or DHCP Protocol. 1=M-Bootp - Allocates this address using the Bootp protocol where the MAC address is defined. 2=M-DHCP - Allocates this address using the DHCP protocol where the MAC address is defined. 3=A-Bootp - Allocates this address using the Bootp protocol where the MAC address is not known. 4=A-DHCP - Allocates this address using the DHCP protocol to a DHCP template with an infinite lease. 5=D-DHCP - Defines this address using the DHCP protocol to a DHCP template for a specific lease time. -1=Static object or static planned move. -2=Reserves an object.
Hardware Type	[O]numeric	Enter the hardware type that corresponds to this object. Refer to the following options: 0=No hardware type defined 1=Ethernet 2=Token Ring 3=AX.25 4=Pronet 5=Chaos 6=IEEE802 7=Arcnet
Manufacturer	[O]text (31 char)	Enter the manufacturer of this object.
Model Type	[O]text (15 char)	Enter the manufacturer's model number of this object.

Field	Value Type	Description
Time to live	[O]numeric	Enter the amount of time (in seconds) that a DNS server is allowed to cache Resource Record data. If this field is not numeric, it is defaulted to -1.
Alias	[O]text (92 char)	Enter one or more aliases for this object (for example, “Alias1,Alias2,Alias3”).
Allow DHCP Clients to Modify Dynamic Object Resource Records	[O] numeric	Specifies the subnet level setting to keep control and make the DNS updates for DHCP clients.The options are: 0 = False 1 = True 2 = Same as in Global Policies. Defaults to this value if not specified.
Users List	[O]text (20 char)	Enter one or more user’s login ID for this object. Enclose a list of users in double quotes (for example, “user1,user2,user3”).
User Class	[O]text (256 char)	Enter the user class. ***Only applicable for QDHCP server objects.***

Note: All data lines must end with a carriage return, or they are not imported.

Input file examples

- To allocate an address for a work station:

```
198.200.234.2,tst2,198.200.234.0,,usa.world.com,Workstation,,,,,3,,,,,work-
   ny
```

- To reserve the address “144.144.144.2”:

```
144.144.144.2,tst2,,,qtek.com,Workstation,,,,,-2,,,,,work2
```

entersubnet

entersubnet is the CLI command for importing Subnet Profile information to an existing network.

Note: The Contact, Location, TFTP Server Name, Application, and Subnet are optional information that must already exist in the VitalQIP database. The input must be in VitalQIP CSV (comma delimited) format.

Synopsis

```
entersubnet -f input_file [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-r reject_file]
   [-e errmsg_file] [-w(overwrite)] [-b]
```

Parameters

entersubnet recognizes the following parameters:

-f ***input_filename***	Specifies the directory and filename of the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-r ***reject_file***	Specifies the directory and filename for the rejected records.
-e ***errmsg_file***	Specifies the filename to which this CLI command writes error messages if they occur. The default is *STDERR*.
-w(overwrite)	Overwrites warning messages.
-b	Not supported.

Table 4-14 Subnet profile data file format

Field	Value type	Description
Subnet Address	[M]numeric, with decimals (15 char)	Enter the subnet address for this Subnet (for example, 100.12.128.0).
Subnet Mask	[M]numeric, with decimals (15 char)	Enter the Subnet Mask for this subnet (for example, 255.255.255.0).
Network Address	[M]numeric, with decimals (15 char)	Enter the Network Address for this subnet (for example, 100.12.128.0).

Field	Value type	Description
Subnet Name	[O]text (32 char)	Enter the Subnet Name for this Subnet Profile.
Application	[O]text (30 char)	Enter the primary application for this subnet (for example, Engineering).
Contact Last Name	[O]text (20 char)	Enter the last name of the contact for this subnet.
Contact First Name	[O]text (20 char)	Enter the first name of the contact for this subnet.
Street 1	[O]text (40 char)	Enter the Street 1 information of the location of the objects in this subnet.
Street 2	[O]text (40 char)	Enter the Street 2 information of the location of the objects in this subnet.
City	[O]text (20 char)	Enter the city of the location of the objects in this subnet.
State	[O]text (10 char)	Enter the state of the location of the objects in this subnet.
Zip	[O]numeric	Enter the zip code of the location of the objects in this subnet.
Country	[O]text (30 char)	Enter the country of the location of the objects in this subnet.
Hardware Type	[O]numeric	Enter the hardware type for this subnet. Refer to the options below: 1=Ethernet 2=Token Ring 3=AX.25 4=Pronet 5=Chaos 6=IEEE802 7=Arcnet
TFTP Server Name	[O]text (253 char)	Enter the server name that will act as the TFTP server.

Field	Value type	Description
Allow DHCP Clients to Modify Dynamic Object Resource Records	[O] numeric	Specifies the subnet level setting to keep control and make the DNS updates for DHCP clients.The options are: 0 = False 1 = True 2 = Same as in Global Policies. Defaults to this value if not specified.
Domain	[O]text (190 char)	Enter a fully qualified domain name for this subnet (for example, qtek.quadritek.com). May have multiple entries. The first domain in a list becomes the default domain for the subnet.

Note: All data lines must end with a carriage return, or they cannot be imported.

Input file example

```
10.1.1.0,255.255.255.0,10.0.0.0,Subnet-1,,,,,,,,,,,,0,qip.com
192.168.0.0,255.255.192.0,192.168.0.0,Subnet-
   2,,,,,,,,,,,,1,qip.com,north.qip.com
192.168.64.0,255.255.192.0,192.168.0.0,Subnet-
   3,,,,,,,,,,,,2,qip.com,south.qip.com
192.168.128.0,255.255.192.0,192.168.0.0,Subnet-4,,,,,,,,,,,,1,qip.com,
   east.qip.com
192.168.192.0,255.255.192.0,192.168.0.0,Subnet-5,,,,,,,,,,,,2,qip.com,
   west.qip.com
172.17.1.0,255.255.255.224,172.17.0.0,Subnet-
   6,,,,,,,,,,,,1,qip.com,north.qip.com,
   south.qip.com,east.qip.com,west.qip.com
172.17.1.32,255.255.255.224,172.17.0.0,Subnet-7,,,,,,,,,,,,1,qip.com
172.17.1.64,255.255.255.224,172.17.0.0,Subnet-8,,,,,,,,,,,,1,qip.com
172.17.1.96,255.255.255.224,172.17.0.0,Subnet-9,,,,,,,,,,,,0,qip.com
172.17.1.128,255.255.255.224,172.17.0.0,Subnet-10,,,,,,,,,,,,0,qip.com
172.17.1.160,255.255.255.224,172.17.0.0,Subnet-11,,,,,,,,,,,,0,qip.com
172.17.1.192,255.255.255.224,172.17.0.0,Subnet-12,,,,,,,,,,,,0,qip.com
172.17.1.224,255.255.255.224,172.17.0.0,Subnet-13,,,,,,,,,,,,0,qip.com
```

entersubnetorg

entersubnetorg is the CLI command for importing Subnet Organization information. The input must be in VitalQIP CSV (comma delimited) format.

Synopsis

```
entersubnetorg -f input_filename [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-r reject_file]
   [-e errmsg_file]
```

Parameters

entersubnetorg recognizes the following parameters:

-f *input_filename* Specifies the directory and filename of the input data. The input file contains the list of subnets that need to be part of the subnet organization. The CLI obtains the list of subnets from the input file and compares it to a list of subnets that already exist within the subnet organization. It deletes the subnets from the subnet organization, if the subnet is not present in the list supplied by the input file.

-g *loginserver* Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.

-s *servername* Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.

-o *organization* Specifies the organization (corporation) name.

-u *username* Specifies the VitalQIP administrator account to be used in establishing the database connection.

-p *password* Specifies the password for the associated administrator account.

-r *reject_file* Specifies the directory and filename for the rejected records.

-e *errmsg_file* Specifies the filename to which this CLI command writes error messages if they occur. The default is *STDERR*.

Table 4-15 Subnet organization data file format

Field	Value type	Description
Subnet Organization Name	[M]text (32 char)	Specify the name you want to give the Subnet Organization.

Field	Value type	Description
Subnet Address	[O]numeric, with decimals (15 char)	Enter the subnet address for this Subnet Organizer (for example, 100.12.128.0). May have multiples of Subnet Address. Multiple subnet addresses must be separated by a comma (,) and enclosed within a set of double quotations (“ ”).
Address Number (GAP)	[O]numeric	If you assign the number 1, for example, with “Router” in the Object Class field below, the Router Object Class will occupy the first IP address in all subnets in this Subnet Group relative to the subnet. (-1 indicates 1 from the end). Multiple sets of Address Numbers/Object Classes may be used. Pairs must be enclosed within a set of (“ ”), for example, “1, Workstation,2,PC”.
Object Class (GAP)	[O]text	The Object Class you want to associate with the Address Number. The options are shown below: Workstation X-terminal PC Printer Server Wiring_HUB Router Bridge Terminal_Server Switch Legacy_System Gateway Test_Equipment Undefined Others User-defined
Site Name	[M]text (32 char)	Specify the name you want to give the Windows 2000 site.

Field	Value type	Description
Site Name Same as Subnetorg	[O]numeric	Determines if the site has the same name as the subnet organization. 0=false 1=true
Domain Controller Server Names	[M]text (255 char)	Specify the Domain Controllers that are to be associated with the subnet organization and managed as a Windows 2000 site within VitalQIP.
DHCP Server Name	[O]text (255 char)	Specify the DHCP server to be associated with the subnet organization.
DHCP Template Name	[O]text (32 char)	Specify the DHCP Option Template to be associated with the subnet organization.
Warning Type	[O]numeric	The warning type you want to use with this subnet organization. The options are shown below: 0=The warning type is not being sent. 1=The “Visual” check box is being selected. 2=The “Email” check box is being selected. 3=Both the “Visual” and the “Email” check boxes are being selected.
Warning threshold	[O]numeric	The threshold this subnet organization will use to send out the warning message. When the percentage of the managed subnets addressed in the associated subnet organization reach this threshold, a warning message will be sent using the method specified in the Warning Type entry.

Note: All data lines must end with a carriage return, or they are not imported.

The Address Number and Object Class must be entered in pairs, as shown below.

```
1,pc,
2,workstation,
```

The Address Number and Object Class can be repeated as often as necessary.

Note: If using a spreadsheet (for example, Excel) to store **entersubnetorg** data, store each Address Number and Object Class value in its own cell.

If you use the - sign, it indicates one from the end. In the example below, you are assigning the gateway to the 9th subnet address from the end.

Input file example

```
Snorg1,"10.200.110.0","-9,Gateway,23,Printer,45,Workstation",,0,,,,3,67
```

To run this CLI, follow these steps:

1. Run the **exportsubnetorg** CLI for a specific organization to get the list of subnet organizations and their subnet data.
2. Edit the output file, note the subnet organizations you wish to modify, and paste them into an input file.
3. Additional subnets that need to be added to the subnet organizations should be added to the input file.
4. Run the **entersubnetorg** CLI for the specific organization using the following syntax:

```
entersubnetorg -u<username> -p<password> -s<server> -f <input file
   name>
```

enterudf

The **enterudf** CLI command imports user-defined fields to the VitalQIP database. The input must be in VitalQIP CSV (comma delimited) format.

Synopsis

```
enterudf -f input_filename [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-r reject_file]
   [-e errmsg_file]
```

Parameters

enterudf recognizes the following parameters:

-f ***input_filename***	Specifies the directory and filename of the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-r ***reject_file***	Specifies the directory and filename for the rejected records.
-e ***errmsg_file***	Specifies the filename to which this CLI command writes error messages if they occur. The default is *STDERR*.

Table 4-16 UDF data file format

Field	Value type	Description
Owner Type	[M] text (190 char)	The type of user-defined field. Options are: 1=Object, 2=Subnet, 6=Domain, 13=Organization, 16=User, 18=Reverse Zone,
Field name/value	[M] text (1 char)	Specify whether this is a field name (**N**)or the value of a field (**V**).
Field name	[M] text (30 char)	If the field name/value is **N** (for example, field name), specify the field name.
Owner(usernamelIP)	[M] text (32 char)	If the field name/value is **V** (for example, field value), specify the IP address or user name of the field.

Field	Value type	Description
Field value	[M] text (128 char)	If the field name/value is **V** (for example, field value), specify the actual value for the field.

Note: All data lines must end with a carriage return, or they are not imported.

Input file examples

```
1,n,psl-udf2
1,v,psl-udf2,150.1.6.14,This is the field value of 150-1-6-14
```

enteruser

enteruser is the CLI command for importing user information to the VitalQIP database. The input must be in VitalQIP CSV (comma delimited) format.

Synopsis

```
enteruser -f input_filename [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-r reject_file]
   [-e errmsg_file] [-c] [-w]
```

Parameters

enteruser recognizes the following parameters:

-f ***input_filename***	Specifies the directory and filename of the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-r ***reject_file***	Specifies the directory and filename for the rejected records.
-e ***errmsg_file***	Specifies the filename to which this CLI command writes error messages if they occur. The default is *STDERR*.
-w	Overwrites warning messages.
-c	This parameter should only be used if the password is not encrypted. The password will be encrypted with the use of the **-c** parameter.

Table 4-17 User data file format

Field	Value type	Description
Log in Name	[M]text (30 char)	The name the user enters to login.
Last Name	[M]text (20 char)	The last name of the user.
First Name	[M]text (20 char)	The first name of the user.
User Group	[M]text (30 char)	User Group names.
Phone Number	[O]numeric (20 digits)	The user's phone number.
E-mail Address	[O]text (63 char)	The e-mail address of the user.

Field	Value type	Description
Password	[O] text (10 char)	The password the user uses to login.
Pin	[O]text (30 char)	The user's Pin number
Description	[O]text (30 char)	A description for this user.
Street1	[O]text (30 char)	The street address of the user.
Street2	[O]text (30 char)	Additional street address information.
City	[O]text (20 char)	The city the user lives in.
State	[O]text (10 char)	The state the user lives in.
Zip	[O]text (15 char)	The user's zip code.
Country	[O]text (30 char)	The country the user lives in.
Activation Type	[O] numeric (0-2)	Activation type, as follows: Active=1 Inactive=0 Pending=2 The default is 0 (inactive).
Default Subnet	[O]	List of default subnet IP addresses, which are separated by a comma, and placed in double quotes.
Managed Range	[O]	Groups of MAC addresses, Hostnames or IP addresses. You may have multiple types and sets. Place multiples within double quotes.
User Field Name	[O]	The field name of the user field. This must exist if you are going to supply the next field, the User Field Value.
User Field Value	[O]	The value of the preceding User Field Name. If there is a comma in this value, the whole string must be surrounded by double quotes.

Note: All data lines must end with a carriage return, or they cannot be imported.

Input file examples

```
usr5,Jones,"pal,jr","isp_qtek,isp2_qtk",292-
   1212,plo@qtek.com,password,6666,"adding managed ranges of the
   user",,,,,,,1,"144.144.144.0,161.251.6.0","112233445566,test.qtek.com,11.
   22.33.44",UDF-1,"the text pertaining to the user field UDF-1"
```

enterusergrp

enterusergrp is the CLI command for importing User Group information to the VitalQIP database. The input must be in VitalQIP CSV (comma delimited) format.

Synopsis

```
enterusergrp -f input_filename [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password][-r reject_file]
   [-e errmsg_file]
```

Parameters

enterusergrp recognizes the following parameters:

-f ***input_filename***	Specifies the directory and filename of the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-r ***reject_file***	Specifies the directory and filename for the rejected records.
-e ***errmsg_file***	Specifies the filename to which this CLI command writes error messages if they occur. The default is *STDERR*.

Table 4-18 User group data file format

Field	Value type	Description
UserGroupName	[M]text (32 char)	The name of the user group.
ContactLastName	[M]text (20 char)	The last name of the contact.
ContactFirstName	[M]text (20 char)	The first name of the contact.
UserGroupDescription	[O]text (32 char)	The description of the User Group.

Note: All data lines must end with a carriage return, or they are not imported.

Input file examples

```
isp_qtek,qipman,qipman,big time isp
```

enterzoneext

enterzoneext is the CLI command for importing zone extension information to the VitalQIP database. The input must be in VitalQIP CSV (comma delimited) format.

Synopsis

```
enterzoneext -f input_filename [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-r reject_file]
   [-e errmsg_file]
```

Parameters

enterzoneext recognizes the following parameters:

-f ***input_filename***	Specifies the directory and filename of the input data.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-r ***reject_file***	Specifies the directory and filename for the rejected records.
-e ***errmsg_file***	Specifies the filename to which this CLI command writes error messages if they occur. The default is *STDERR*.

Table 4-19 Domain data file format

Field	Value type	Description
Zone name	[M]text (up to 256 char)	The fully qualified name of the domain (for example, mydomain.qtek.com) or the IP address and mask of the reverse zone.
Extension Location	[O]text	The location of the extension. Valid types are: **Prefix of zone db file** - prefix extensions **Postfix of zone db file** - postfix extensions If no extension location is specified, the **Postfix of zone db file** location is used.
Zone Extension	[M] text	Up to 256 characters of additional text for the extension.

Note: All data lines must end with a carriage return, or they are not imported.

Input file example

```
example.com,Postfix of zone db file,$include
   /ipaa/internal/spcl.west
example.com,Prefix of zone db file,$include
   /ipaa/internal/spcl.west
```

qipbulkload

qipbulkload pre-loads IPv4 objects from a comma delimited file that you create. It reads a file in the format shown below and loads the data by using the **qsi-import** routine (refer to the *VitalQIP Installation Guide*). It is, essentially, a condensed version of the **entersimpleobj** import utility. **qipbulkload** is installed in *$QIPHOME/usr/bin* directory on UNIX platforms and in the *%QIPHOME%* directory on Windows platforms.

Note: **qipbulkload** only loads IPv4 objects.

VitalQIP validates data to prevent CNAME conflicts in the same organization by default. If a CNAME conflict occurs when adding an object and alias data via the qipbulkload CLI, an error message displays and the object add will fail. You can disable this feature by setting the Validate CNAME Records policy to False. See the Object Policies and Profiles section of the *VitalQIP User Guide* for more information on the policy. When you add an object and alias data, validation checks on the object name and aliases are made against the following:

- Object Names
- Domain Names
- Aliases
- Object Resource Records
- Domain Resource Records
- Reverse Zone Resource Records

Before you begin

- If you build your infrastructure from scratch, **qipbulkload** cannot be used until forward zones have been bound to Subnet Profiles.
- No other VitalQIP process except for the VitalQIP Login Service (GUI, daemon, CLI, etc.), VitalQIPMessage Service and VitalQIP SSL Tunnel Service should be active when **qipbulkload** is run to avoid objects being added from other sources and causing corruption.
- **qsi-import** must be in the same path as **qipbulkload**.
- If records are rejected by the **qipbulkload** CLI command, they are sent to a file (called **bulkload.rej**) that is created automatically, or the record's associated error message is sent to the log file you specified in the –l ***<logfile_name>*** option.
- **qipbulkload** should only be used for adding objects. Use **qip-setobject** if you wish to modify them.
- Duplicate errors can occur. Do not have duplicates in data as input.

- Before you use **qipbulkload**, ensure that your license covers the total number of IPv4 and IPv6 objects in the input file. VitalQIP does not warn users that the maximum number of objects covered by a license has been reached during a database load, but instead locks them out of the database.

  Important! Even though **qipbulkload** only loads IPv4 objects, you must obtain a count of both IPv4 and IPv6 objects to avoid being locked out of the database.

- To check the number of objects covered by your license, select **Help | About VitalQIP**. Alternatively, if you want to determine how many objects exist before you use **qipbulkload**, execute the following commands:

IPv4 Objects

Sybase:

```
isql -U qipman -P password
1>select count(*) from obj_prof
2>go
```

Oracle:

```
sqlplus qipman/password@ORASID
sql>select count(*) from obj_prof
```

IPv6 Objects

Sybase:

```
isql -U qipman -P password
1>select count(*) from ipaddress
2>go
```

Oracle:

```
sqlplus qipman/password@ORASID
sql>select count(*) from ipaddress
```

If you are already locked out of VitalQIP and have exceeded the number of IPs covered by your current license, contact your sales representative to obtain a temporary license for a larger number of objects so you can access the GUI and delete excess objects.

Synopsis

```
qipbulkload [-g loginserver] [-s servername] [-o organization]
   [-u username] [-p password] -f input_filename [-t O|S]
   [-e errmsgfile] [-z] [-r reject_file]
```

Parameters

qipbulkload recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-f *input_filename*	Specifies the filename from which the information is read. The default is *STDIN*.
-t O|S	Specifies either O=Oracle or S=Sybase.
-e *errmsgfile*	Specifies the name of the file to contain information concerning the records that could not be processed. The default is *STDERR*.
-z	Specifies that qipbulkload runs in test mode. Specifically, generates the import files but prints the import command to the screen instead of executing the import command. Also, leaves the import files in place upon exit, instead of deleting them This gives you a chance to look at the import files before actually loading them.
-r *reject_file*	Specifies the file where records that are not added or modified are placed.

Table 4-20 Input file format

Field	Value Type	Description
Object IP Address	[M]numeric, with decimals (15 char)	Enter the IP address of the object (for example, 100.12.128.28).
Object(Device) Name	[M]text (63 char)	Enter the object device name for this object.

Field	Value Type	Description
Subnet Address	[O]numeric, with decimals (15 char)	Enter the subnet address for this object (for example, 100.12.128.0).
DHCP Server Name	[O]text (253 char)	Enter the fully qualified name of the DHCP Server assigned for this object (for example, dhcp1.quadritek.com).
Domain Name	[O]text (190 char)	Enter the fully qualified domain name for this object (for example, qtek.quadritek.com).
Object Class Description	[O]	Refer to the following options: - Workstation - X-terminal - PC - Printer - Server - Wiring_HUB - Router - Bridge - Terminal_Server - Switch - Legacy_System - Gateway - Test_Equipment - Undefined - Others - External
Comment	[O]text (32 char)	Enter a description for this object.
MAC Address	[O]12 or 16 (in hexadecimal)	Enter the MAC Address for this object.
Contact Last Name	[O]text (20 char)	Enter the last name of the contact for this object.
Contact First Name	[O]text (20 char)	Enter the first name of the contact for this object.

Field	Value Type	Description
Contact Phone No.	[O]text (20 char)	Enter the phone number of the contact for this object.
Dynamic Configuration	[O]numeric	Enter the dynamic configuration for this object. Refer to the following options: 0=None - Allocates this address without using the Bootp or DHCP Protocol. 1=M-Bootp - Allocates this address by using the Bootp protocol where the MAC address is defined. 2=M-DHCP - Allocates this address by using the DHCP protocol where the MAC address is defined. 3=A-Bootp - Allocates this address by using the Bootp protocol where the MAC address is not known. 4=A-DHCP - Allocates this address by using the DHCP protocol to a DHCP template with an infinite lease. 5=D-DHCP - Defines this address by using the DHCP protocol to a DHCP template for a specific lease time.
Hardware Type	[O]numeric	Enter the hardware type that corresponds to this object. Refer to the following options: 1=Ethernet 2=Token Ring 3=AX.25 4=Pronet 5=Chaos 6=IEEE802 7=Arcnet
Manufacturer	[O]text (31 char)	Enter the Manufacturer of this object.
Model Type	[O]text (15 char)	Enter the manufacturer's model number of this object.
Time to live	[O]numeric	Specify the amount of time (in seconds) that a DNS server is allowed to cache Resource Record data.
Alias	[O]text (92 char)	Enter one or more aliases for the object (for example, "Alias1,Alias2,Alias3").

Field	Value Type	Description
Allow DHCP Clients Modify Dynamic Object Resource Records	[O] numeric	Specifies the subnet level setting to keep control and make the DNS updates for DHCP clients. The options are: 0 = False 1 = True 2 = Same as in Global Policies. Defaults to this value if not specified.

Note: All data lines must end with a carriage return, or they are not imported.

Input file examples

```
10.200.90.3,wks1,10.200.90.0,,seg4.qa.quadritek.com,workstation,test
   workstation,484453000000,qipman,qipman,,,1,,,1231,,
10.200.90.4,lpt2,10.200.90.0,,seg4.qa.quadritek.com,switch,test
   switch,490003111111,qipman,qipman,,,2,,ACC,1231,,
10.200.90.5,pc3,10.200.90.0,,seg4.qa.quadritek.com,pc,test
   pc,0000E2222222,qipman,qipman,,,3,,ACER,1231,,
10.200.90.6,ptr4,10.200.90.0,,seg4.qa.quadritek.com,printer,test
   printer,900010333333,edwards,cindy,555-1212,,4,,AT&T,1231,,
10.200.90.7,gtw5,10.200.90.0,,seg4.qa.quadritek.com,gateway,test
   gateway,eiei0eeeeeee,murphy,paula,,,5,,,1231,,
10.200.90.8,bdg6,10.200.90.0,,seg4.qa.quadritek.com,bridge,brooklyn
   bridge,eac4daeeffff,schmidt,alexander,,,6,,ALGORITHMS SOFTWARE PVT.
   LTD.,1231,,
10.200.90.9,dns1,10.200.90.0,,seg4.qa.quadritek.com,server,the
   dns_server,,schmidt,alexander,,,7,,,1231,,
10.200.90.10,hub01,10.200.90.0,,seg4.qa.quadritek.com,wiring_hub,the
   server,712377fffe77,qipman,qipman,,,1,,,1231,,
10.200.90.11,hub02,10.200.90.0,,seg4.qa.quadritek.com,wiring_hub,the
   wiring_hub,87eeddaadd33,murphy,paula,,,2,,,1231,,
10.200.90.12,rtr01,10.200.90.0,,seg4.qa.quadritek.com,router,the
   router,,newman,alfred,,,3,,,1231,,
10.200.90.13,tserver01,10.200.90.0,,seg4.qa.quadritek.com,terminal_server,th
   e terminal_server,98700aaaaadd,qipman,qipman,,,4,,,1231,,
10.200.90.14,usa1,10.200.90.0,,seg4.qa.quadritek.com,legacy_system,the
   legacy_system,ace1234deac7,edwards,cindy,,,5,,,1231,,
10.200.90.15,testpack01,10.200.90.0,,seg4.qa.quadritek.com,test_equipment,th
   e test_equipment,ccaaeedd1122,schmidt,alexander,,,6,,,1231,,
```

qiploadsndomn

qiploadsndomn loads subnet domains from a comma delimited file that you create. It reads a file in the format shown below and loads the data using the **qsi-import** routine (refer to the *VitalQIP Installation Guide* for more information on this routine). **qiploadsndomn** is installed in *$QIPHOME/usr/bin* directory on UNIX platforms, or *%QIPHOME%* directory on Windows platforms. It must be in the same directory as **qsi-import**.

Before you begin

- No other VitalQIP process (for example, GUI, daemon, CLI) should be active when **qiploadsndomn** is run.
- **qsi-import** must be in the same path as **qiploadsndomn**.
- The subnets that are included in the input file must have been defined in VitalQIP.

Synopsis

```
qiploadsndomn [-g loginserver] [-s servername] [-o organization]
   [-u username] [-p password] -f input_filename [-t O|S]
   [-e errmsgfile] -z
```

Parameters

qiploadsndomn recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-f ***input_filename***	Specifies the file from which the information is read. The default is *STDIN*.
-t O|S	Specifies either **O**=Oracle or **S**=Sybase.
-e ***errmsgfile***	Specifies the name of the file to contain information concerning the records that could not be processed. The default is *STDERR*.

-z If this parameter is specified, **qiploadsndomn** runs in test mode. It generates the import file, but prints the import command to the screen instead of executing the import command. It also leaves the import files in place on exit instead of deleting them. This gives you the chance to look at the import files before actually loading them.

Table 4-21 Input file format

Field	Value type	Description
Record Type Identifier	[M] subnet	Specifies the record type as a subnet.
Subnet IP Address	[M] dotted decimal (20 char)	Specifies the IP address of the subnet where the domains are located.
Domain Name	[M] text (up to 253)	The list of domains associated with the subnet (Comma separated).

Note: All data lines must end with a carriage return, or they cannot be imported.

Exporting files with CLIs

This section describes the use of CLI modules for exporting VitalQIP objects to the VitalQIP comma separated file format (CSV). These files can be modified, if desired, and used for importing. The files are automatically exported in the VitalQIP import file formats, as described in the corresponding Import section.

exportaddrrange

exportaddrrange is the CLI command for exporting managed address ranges for networks and managed object ranges within subnets into VitalQIP CSV format.

Synopsis

```
exportaddrrange [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-f filename]
   -t range_type [-a ip_address]
```

Parameters

exportaddrrange recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-f *filename*	Specifies the directory and filename of the output data. The default is *STDOUT*.
-t *range_type*	Specifies a network or subnet (object) range, as follows: **network** or **subnet**.
-a *ip_address*	Specifies a subnet or network address for exporting only the ranges within a single subnet or network.

Command line input example 1

- To export a network address range:

```
exportaddrrange -t network
```

Output example 1

Refer to Table 4-1, "Address range data in VitalQIP CSV format" (p. 4-3) for an explanation of the field layout.

```
148.94.0.0,148.94.0.1,148.94.0.20
```

Command line input example 2

- To export a subnet object range:

```
exportaddrange -t subnet
```

Output example 2

Refer to Table 4-1, “Address range data in VitalQIP CSV format” (p. 4-3) for an explanation of the field layout.

```
198.200.234.0,198.200.234.20,198.200.234.50
```

exportcontact

exportcontact is the CLI command for exporting contact information into VitalQIP CSV format.

Synopsis

exportcontact [**-g** ***loginserver***] [**-s** ***servername***] [**-o** ***organization***] [**-u** ***username***] [**-p** ***password***] [**-f** ***filename***]

Parameters

exportcontact recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-f ***filename***	Specifies the directory and filename of the output data. The default is *STDOUT*.

Output example

Refer to Table 4-5, "Domain and DNS Zone data file format" (p. 4-16) for an explanation of the field layout.

```
Day,"Betty",bday@quadritek.com,44-1256-346-344,
Day,"Betty",,44-1256-346-344,44-1256-346-355
Drescher,"Alex",,610-725-8535,800-sky-Page
Klein,"Dave",,,
Reiley,"Jim",,,
Scott,"Erin",escott@world.com,610-725-8535,
Scott,"Erin",,610-725-8535,800-Sky-Page
Swiss,"Bob",,,
qipman,"qipman",,,
```

exportdnsrr

exportdnsrr is the CLI command for exporting all DNS resource record information associated with objects, domains, and reverse zones in VitalQIP CSV format.

Synopsis

```
exportdnsrr -t object|domain|reverse_zone [-g loginserver]
   [-s servername] [-o organization] [-u username] [-p password]
   [-f filename]
```

Parameters

exportdnsrr recognizes the following parameters:

-t object|domain|reverse_zone	The owner type of the resource records (for example, objects, domains, or reverse zones).
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-f *filename*	Specifies the directory and filename of the output data.

Output examples

Refer to Table 4-4, "DNS resource record file input file format" (p. 4-12) for an explanation of the field layout.

- Sample output for object resource record.

```
150.1.0.11,<HOST_NAME>,IN,HINFO,-1,""""QIP DDNS"""" ""BIND 8.x""",F,O,O,,
150.1.0.11,<HOST_NAME>,IN,WKS,-1,150.1.0.11 53 11 22 33 44,F
150.1.4.8,<FULL_NAME>,IN,HINFO,-1,""QIP DDNS"" ""BIND 4.x""",F
```

- Sample output for domain resource record:

```
usa.world.com,www.usa.world.com,IN,CNAME,-
   1,webserver.usa.world.com
```

- Sample output for reverse zone resource record:

```
198.200.138.0/24,138.200.198.in-addr.arpa,CH,AAAA,0,321:0:1:2:3:4:567:89AB,R
   ,,0,0,"","""
```

exportdomain

exportdomain is the CLI command for exporting domain and DNS zone option information into VitalQIP CSV format.

Synopsis

```
exportdomain [-f output_filename] [-sh] -nh [-g loginserver]
   [-s servername] [-o organization] [-u username] [-p password]
```

Parameters

exportdomain recognizes the following parameters:

-f ***output_filename***	Specifies the directory and file name of the output data. Each field in the output file is described in Table 4-5.
-sh	If **-sh** is not specified, a common header is used for all the domains in the output file, and for all the domains, only those fields shown in the domain profile window in the VitalQIP GUI will be included in the output file. If **-sh** is specified, a separate header is used for each domain and each domain is separated by a space line in the output file. The zone option fields are also included in the output file. If any DNS server is associated with a domain, that DNS server information is also included in the output file.
-nh	Specifies that no header appears in the output file. This data file format is specifically required for importing domains from GUI. and it only contains those fields shown in the domain profile GUI.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.

Output examples

Refer to Table 4-5, "Domain and DNS Zone data file format" (p. 4-16) for an explanation of the field layout.

```
bank.com,21600,604800,3600,86400,bank@qtek.com,0,0,"any","any","any", ,""…..
```

Note: The output files from **exportdomain** can be used directly as input files for the **enterdomain** CLI. Users should run **exportdomain** to get the correct input file format and modify the input file data as needed before running the **enterdomain** CLI. You can get both input file formats from **exportdomain** whether or not you use the **`-sh`** parameter in the command line.

exportlocation

exportlocation is the CLI command for exporting location information into VitalQIP CSV format.

Synopsis

exportlocation [**-g** ***loginserver***] [**-s** ***servername***] [**-o** ***organization***] [**-u** ***username***] [**-p** ***password***] [**-f** ***filename***]

Parameters

exportlocation recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-f ***filename***	Specifies the directory and filename of the output data.

Output example

Refer to Table 4-7, "Location file format" (p. 4-24) for an explanation of the field layout.

```
,,Berne,,,Switzerland
,,Basingstoke,,,UK
,,New York,NY,,USA
,,Philadelphia,PA,,USA
,,Malvern,PA,,USA
```

exportmanufacturer

exportmanufacturer is the CLI command for exporting manufacturer information into VitalQIP CSV format.

Synopsis

```
exportmanufacturer [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-f filename]
```

Parameters

exportmanufacturer recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-f ***filename***	Specifies the directory and filename of the output data.

Output example

Refer to Table 4-8, "Manufacturer data file format" (p. 4-26) for a description of the field layout.

exportnetwork

exportnetwork is the CLI command for exporting network information into VitalQIP CSV format.

Synopsis

exportnetwork [**-g** ***loginserver***] [**-s** ***servername***] [**-o** ***organization***] [**-u** ***username***] [**-p** ***password***] [**-f** ***filename***]

Parameters

exportnetwork recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-f ***filename***	Specifies the directory and filename of the output data.

Output example

Refer to Table 4-9, "Network data file format" (p. 4-28) for a description of the field layout.

```
148.94.0.0,16,root@dns1.world.com,Network1,N,0,0,dns1.world.com
150.1.0.0,16,root@dns1.world.com,Network2,N,0,0,dns1.world.com
150.150.0.0,16,root@dns1.world.com,Network3,N,50,2,dns4.uk.world.
   com
155.55.0.0,16,root@dns1.world.com,Network4,N,0,0,
198.200.234.0,24,root@dns1.world.com,Network5,N,0,0,dns1.world.co
   m
```

exportorganization

exportorganization is the CLI command for exporting organization information into VitalQIP CSV format. This command exports the maximum object count.

Synopsis

exportorganization [**-g** ***loginserver***] [**-s** ***servername***] [**-u** ***username***] [**-p** ***password***] [**-f** ***filename***]

Parameters

exportorganization recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-f ***filename***	Specifies the directory and filename of the output data.

Output example

Refer to Table 4-10, "Organization data file format" (p. 4-30) for a description of the field layout.

```
1,"VitalQIP Organization","Default Organization",0
```

exportospf

exportospf is the CLI command for exporting OSPF information into VitalQIP CSV format.

Synopsis

```
exportospf [-g loginserver] [-s servername] [-o organization]
   [-u username] [-p password] [-f filename]
```

Parameters

exportospf recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-f *filename*	Specifies the directory and filename of the output data.

Output example

Refer to Table 4-11, "OSPF area data file format" (p. 4-32) for a description of the field layout.

```
England,000.000.000.001,1,50,2,"150.150.1.0,150.150.1.64",150.150.000.000,
150.150.031.255
```

exportreversezone

exportreversezone is the CLI command for exporting reverse zone information into VitalQIP CSV format.

Synopsis

```
exportreversezone [-f output_filename] [-sh] [-g loginserver]
   [-s servername] [-o organization] [-u username] [-p password]
```

Parameters

exportreversezone recognizes the following parameters:

-f ***output_filename***	Specifies the directory and file name of the output data. Each field in the output file is described in Table 4-12.
-sh	If **-sh** is not specified, a common header is used for all the reverse zones in the output file, and for all the reverse zones, only those fields shown in the reverse zone profile window in the VitalQIP GUI will be included in the output file. If **-sh** is specified, a separate header is used for each reverse zone and each reverse zone record is separated by a space line in the output file. All of the zone option fields are included in the output file. If any DNS server is associated with the reverse zones, that DNS server information is also included in the output file.
-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u **username**	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.

exportserver

exportserver is the CLI command for exporting server information into VitalQIP CSV format.

Synopsis

exportserver [**-g** ***loginserver***] [**-s** ***servername***] [**-o** ***organization***] [**-u** ***username***] [**-p** ***password***] [**-f** ***filename***]|[**-c** ***class***] [**-t** ***type***]

Parameters

exportserver recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-f ***filename***	Specifies the directory and filename of the output data.
-c ***serverClass***	Specifies the class of the server .
-t ***serverType***	Specifies the type of server from the following list: Bootp, Lucent DNS 4.x, DHCP, NIS, Local Host, Windows 2000 **-t** always overrides anything entered in for the **-c** parameter. If neither is included, all servers are returned.

Output example

Refer to the **enterserver** CLI for a description of the field layout.

```
Type=<Server_Class><CR>
<Format><CR>
<Data_Values><CR>
```

exportsimpleobj

exportsimpleobj is the CLI command for exporting object information in VitalQIP format into VitalQIP CSV format.

Synopsis

exportsimpleobj [**-g** ***loginserver***] [**-s** ***servername***] [**-o** ***organization***] [**-u** ***username***] [**-p** ***password***] [**-f** ***filename***]

Parameters

exportsimpleobj recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-f ***filename***	Specifies the directory and filename of the output data.

Output Example

Refer to Table 4-13, "Object data in VitalQIP file format" (p. 4-48) for a description of the field layout.

```
212.0.0.1,wsp000078wss,,,qtek.com,Workstation,,,,,,-
   1,0,,,0,,,,1,,,Workstation,yes,1
212.0.0.2,pcp000012pcs,,,qtek.com,PC,,,,,,-1,0,,,-1,,,,1,,,PC,1
212.0.0.3,wsp000108wss,,,qtek.com,Workstation,,,,,,-1,0,,,-1,"asdf.qtek.com.
,bdaf.qtek.com.,gfz.qtek.com.,qwerqew.qtek.com.",,,1,,,Workstation,2
212.0.0.4,worksta,,,qtek.com,Workstation,,,,,,-1,0,,,-
   1,,,,1,,,Workstation,0,1
212.0.0.5,wsp000112wss,,,qtek.com,Workstation,,,,,,-1,0,,,
   -1,"asdfafs.qtek.com.,asdfsa.qtek.com.",,,1,,,Workstation,1,1
198.102.15.1,itsawonderfullife,,,northpole.com,Server,,,,,,-1,0,,,-
   1,,,,1,,,Server,1,1
198.102.15.2,santalaptop,,,northpole.com,Workstation,santa's
laptop,,,,,-1,0,,,-1,,,,1,,,Workstation,1,1
198.102.15.3,santadesktop,,,northpole.com,Workstation,santa's,1
desktop,,,,,-1,0,,,-1,"stella.q.com.",,,1,,,Workstation,1
198.102.15.4,wsp000113wss,,,northpole.com,Workstation,,,,,,-1,0,,,-1,,,,1,,,
Workstation,1
211.111.110.1,w1,,,qtek.com,Undefined,,,,,,-1,0,,,-1,,,,1
```

```
211.111.110.2,w2,,,qtek.com,Undefined,,,,,-1,0,,,-1,,,,1
211.111.110.3,w3,,,qtek.com,Undefined,,,,,-1,0,,,-1,,,,0
211.111.110.4,w4,,,qtek.com,Undefined,,,,,-1,0,,,-1,,,,1
211.111.110.5,wsp000115wss,,test-dhcp.qtek.com,qtek.com,Workstation,,,,,5,0
,,,-1,,,"user_class1,user_class2",1,,,Workstation,1
211.111.110.6,wsp000116wss,,test-dhcp.qtek.com,qtek.com,Workstation,,,,,5,0
,,,-1,,,"user_class1,user_class2",2
211.111.110.7,wsp000117wss,,test-dhcp.qtek.com,qtek.com,Workstation,,,,,5,0
,,,-1,,,"user_class1,user_class2",1
211.111.110.8,wsp000118wss,,test-dhcp.qtek.com,qtek.com,Workstation,,,,,5,0
,,,-1,,,"user_class1,user_class2",1
211.111.110.9,udp000085uds,,first.one.com,qtek.com,Undefined,,,Test,John,123
-1234,3,0,,,-1,,,,1,,,Undefined,1
211.111.110.10,whp000033whs,,,qtek.com,Wiring_HUB,,,Test,John,123-1234,-
   1,0,,,
-1,"a1.qtek.com.,a2.qtek.com.",,,1,,,Wiring_HUB,1
211.111.110.11,router1,,,qtek.com,Router,,,,,-1,0,,,-1,,,,1,,,Router,1
211.111.110.12,wsp000122wss,,,qtek.com,Workstation,,,,,-1,0,,,-
   1,,,,1,,,Workstation,0
```

exportsubnet

exportsubnet is the CLI command for exporting Subnet Profile information into VitalQIP CSV format.

Synopsis

exportsubnet [**-g** ***loginserver***] [**-s** ***servername***] [**-o** ***organization***] [**-u** ***username***] [**-p** ***password***] [**-f** ***filename***]

Parameters

exportsubnet recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-f ***filename***	Specifies the directory and filename of the output data.

Output example

Refer to Table 4-14, "Subnet profile data file format" (p. 4-52) for a description of the field layout.

```
148.94.1.0,255.255.255.0,148.94.0.0,,,,,,,,,,1,,,,1,world.com
148.94.0.0,255.255.255.0,148.94.0.0,,,,,,,,,,1,,,,1,world.com
150.1.40.0,255.255.252.0,150.1.0.0,,,,,,,,,,1,,,,2,usa.world.com,uk.world.c
   om
150.1.6.128,255.255.255.192,150.1.0.0,,,,,,,New York,NY,,USA,1,
   Bootp Server,,,0,usa.world.com
150.1.6.64,255.255.255.192,150.1.0.0,New York,Sales,Scott,Erin,,,
   New York,NY,,USA,1,Bootp Server,,,1,usa.world.com
150.1.6.0,255.255.255.192,150.1.0.0,Philadelphia,Sales,Scott,Erin,,,
   Philadelphia,PA,,USA,2,Bootp Server,,,1,usa.world.com
150.1.0.0,255.255.255.0,150.1.0.0,Data Center,Network,Scott,Erin,,,Malvern,
   PA,,USA,1,Bootp Server,,,2,usa.world.com,world.com
150.1.1.20,255.255.255.252,150.1.0.0,SL-New York,Network,Scott,Erin,,,
   Malvern,PA,,USA,1,Bootp Server,,,1,usa.world.com,world.com
150.1.1.16,255.255.255.252,150.1.0.0,SL-Alps,Network,Scott,Erin,,,Malvern,
   PA,,USA,1,Bootp Server,,,0,usa.world.com,world.com
150.1.1.12,255.255.255.252,150.1.0.0,SL-
   Geneva,Network,Scott,Erin,,,Malvern,PA,,USA,1,Bootp
   Server,,,1,usa.world.com,world.com
```

```
150.1.1.8,255.255.255.252,150.1.0.0,SL-Winchester,Network,Scott,Erin,,,
   Malvern,PA,,USA,1,Bootp Server,,,1,usa.world.com,world.com
```

exportsubnetorg

exportsubnetorg is the CLI command for exporting Subnet Organization information into VitalQIP CSV format.

Synopsis

exportsubnetorg [**-g** ***loginserver***] [**-s** ***servername***]
[**-o** ***organization***] [**-u** ***username***] [**-p** ***password***] [**-f** ***filename***]

Parameters

exportsubnetorg recognizes the following parameters:

-g ***loginserver***	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s ***servername***	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o ***organization***	Specifies the VitalQIP organization (corporation) name.
-u ***username***	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p ***password***	Specifies the password for the associated administrator account.
-f ***filename***	Specifies the directory and filename of the output data.

Output example

Refer to Table 4-15, "Subnet organization data file format" (p. 4-55) for a description of the field layout.

```
SubTest1,"10.200.60.0","10,Server",,0,,,microsoft_clients,2,85
SubOrg,"10.12.12.0,198.200.138.0,198.200.138.64,198.200.138.128,198.200.138.
   160,198.200.138.192","-1,X-
   terminal,1,Terminal_Server,2,Switch,3,Legacy_System,4,Gateway,5,Test_Equi
   pment,6,Undefined,7,Partially_Managed",SubOrgC,0,"3ibm.seg2.qa.quadritek.
   com",vqaw2ka01.seg1.qa.quadritek.com,general,3,24
SubTestMoreSubnets,"198.200.138.32","6,Bridge",,0,"3ibm.seg2.qa.quadritek.co
   m",,,1,40
test12,"10.58.208.0","-
   5,Workstation,3,Workstation",,0,,srt08.seg5.qa.quadritek.com,general,2,25
Subnetorgtest,,"-9,Server",,0,,vqaw2ka01.seg1.qa.quadritek.com,general,0,33
```

exportudf

exportudf is the CLI command for exporting user-defined field names or values into the VitalQIP CSV format.

Synopsis

```
exportudf -t field_name|value [-g loginserver] [-s servername]
   [-o organization] [-u username] [-p password] [-f filename]
```

Parameters

exportudf recognizes the following parameters:

-t field_name|value	Exports the user-defined field name or value.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-f *filename*	Specifies the directory and filename of the output data. Output is in CSV format.

Output Example

Refer to Table 4-16, "UDF data file format" (p. 4-59) for a description of the field layout.

```
object,n,Field_1
object,n,Field_2
object,n,Field_3
object,n,Field_4
object,n,Field_5
organization,n,AUTHORITY
domain,n,AUTHORITY
revzone,n,AUTHORITY
#Export UDF Values - Example: export -t value
organization,v,AUTHORITY,150.1.0.19,James Mitchell
domain,v,AUTHORITY,150.1.0.19,William Gates
domain,v,AUTHORITY,150.1.0.19,Tom Cruise
domain,v,AUTHORITY,150.1.0.19,Pierce Brosnan
```

exportuser

exportuser is the CLI command for exporting user information to the VitalQIP CSV format.

Synopsis

```
exportuser [-g loginserver] [-s servername] [-o organization]
   [-u username] [-p password] [-f filename]
```

Parameters

exportuser recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-u *username*	Specifies the VitalQIP administrator account to be used. in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-f *filename*	Specifies the directory and filename of the output data.

Output example

Refer to Table 4-17, "User data file format" (p. 4-61) for a description of the field layout. Exported files contain encrypted passwords. The record will contain one or more records similar to this:

```
johndoe,"Doe","John","Group1",,,01228529f529bd8ae71006ce2857b0a12
   1eb,,,,,,,,,1,,"200.200.200.38",,
```

exportusergrp

exportusergrp is the CLI command for exporting user group information into the VitalQIP CSV format.

Synopsis

```
exportusergrp [-g loginserver] [-s servername] [-o organization]
   [-u username] [-p password] [-f filename]
```

Parameters

exportusergrp recognizes the following parameters:

-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.
-f *filename*	Specifies the directory and filename of the output data.

Output example

Refer to Table 4-18, "User group data file format" (p. 4-63) for a description of the field layout.

```
VitalQIP Admins,,,Administrators of VitalQIP
user_group_1,,,Default User Group
```

exportzoneext

exportzoneext is the CLI command for exporting zone extension information from domains, reverse zones, and root zones. The output is in VitalQIP CSV (comma delimited) format.

Synopsis

```
exportzoneext -t domain|reverseZone|rootZone -f input_filename
   [-g loginserver] [-s servername] [-o organization]
   [-u username] [-p password]
```

Parameters

exportzoneext recognizes the following parameters:

-t domain| reverseZone| rootZone	Specifies the type of export source (domain, reverseZone, or rootZone). The default is *all*. **domain** exports data comes from the **Domain Extension** tab of the Domain Profile. **reverseZone** exports data comes from the **Reverse Zone Extension** tab of the Reverse Zone Profile. **rootZone** exports data comes from the **Root Zone Extension** field in the DNS Server Profile.
-f *input_filename*	Specifies the directory and filename of the input data.
-g *loginserver*	Specifies the VitalQIP login server's IP address. This value is the equivalent of the LOGIN environment variable.
-s *servername*	Specifies the VitalQIP database server. This value is the value of the QIPDATASERVER environment variable.
-o *organization*	Specifies the VitalQIP organization (corporation) name.
-u *username*	Specifies the VitalQIP administrator account to be used in establishing the database connection.
-p *password*	Specifies the password for the associated administrator account.

Output examples

Refer to Table 4-19, "Domain data file format" (p. 4-64) for a description of the field layout.

- Sample export of domain extensions:

```
world.com,us.personnel IN A 192.249.249.44, postfix
world.com, IN MX 10 us.personnel.pa.com., postfix
world.com,employeedb.personnel IN CNAME us.personnel.pa.com.,
   postfix
```

- Sample export of reverse zone extensions:

```
148.94.0.0/16,samplesystem.tstsvr IN A 192.253.252.10, postfix
```

```
148.94.0.0/16, IN MX 10 samplesystem.tstsvr.uk.com., postfix
148.94.0.0/16, IN MX 100 backupsystem.tstsvr.uk.com., postfix
148.94.0.0/16,qip.tstsvr IN CNAME samplesystem.tstsvr.uk.com.,
  postfix
```

5 Web Client CLIs

Overview

Web Client CLIs allow you to perform functions that you typically perform with the Web Client. These CLIs are in a format that differs from other CLIs used in VitalQIP® (QIP), and their use and syntax also differ.

About Web Client CLIs

Usage

The format of the CLIs in this chapter is the **`qip-cli`** command followed by a series of options. All CLIs start with a **`--command`** or **`-c`** option to specify an object type (such as "rule") and an action (such as "add"). Subsequent arguments are specific to the object.

The user, password and organization can be specified on the command line, as shown in the following examples, or they can be set in a configuration file. If you configure them in a configuration file, you do not have to enter them each time you use a CLI. See Set login options for details on configuring these options.

All CLIs in this chapter can be invoked with long-format options, and many can be invoked with short-format options. The long options are preceded with a double-dash (--), and the short options are preceded by a single dash (-).

For example, the following two examples perform exactly the same function:

```
qip-cli --command rule,add --user admin1 --password password
   --organization VitalQIP --inputfile ruleparms.xml --batch
   --rejectfile ruleparmreject.xml

qip-cli -c rule,add -u admin1 -p password -o VitalQIP
   -I ruleparms.xml --batch -r ruleparmreject.xml
```

As you become familiar with the various options, you may find it quicker to use the shorter version when one is available. Additionally, long options may be abbreviated to their shortest unique string.

> **Note:** Not all options have a short version, and not all short versions of an option start with the same letter as their corresponding long version. Check the documentation of the individual CLI for all available options.
>
> Both short and long options are case-sensitive.
>
> Some CLIs allow you to specify User-Defined Attributes (UDAs). The syntax for these options differs from standard CLI syntax. UDA syntax is documented in the CLIs where it is used.

Exit codes

CLIs return the following codes upon completion:

- `0` - success
- `1` - failure

Set login options

Each CLI is shown here with options for user, password and organization. To avoid the need to enter these options each time you invoke a CLI, the software searches for default values in the following files in the order listed:

1. *$HOME/cli.properties*
2. *$QIPHOME/conf/cli.properties*
3. *$QIPHOME/qip.pcy* (the global section)

Note: Passwords entered in these files must be encrypted by *qip-crypt*. See "qip-crypt" (p. 1-35) for information about *qip-crypt*.

The *cli.properties* file also allows you to specify a log directory, *log4j* file, and number of rows to print for CLIs that support **`--format text`**.

Both UNIX and Windows accept a forward slash in file names, for example *C:/myFile.xml* Only Windows accepts a backslash in a filename.

As soon as the software finds a default value, it stops searching the remaining files for that value.

An example *$HOME/cli.properties* follows. By using a personal *cli.properties* file you can run **`qip-cli`** as a regular (non-root) user. The one constraint is choosing a log directory to which your ID has write access.

```
cli.properties:
  User=myuserid
  Password=<your encrypted password goes here>
  Organization=VitalQIP Organization
  cli.logdir=${HOME}/log
  cli.log4j=YourLog4jConfigurationFile# Optional
  cli.prettyPrintNumRows=1000
```

Console Mode

VitalQIP® (QIP) provides a Console mode for invoking Web Client CLIs. This mode is useful for stringing together a series of CLIs in a script and for interactively entering a group of commands. Syntax is as follows:

`qip-cli --console[=`*`option1,option2,...`*`]`

Options are:

- **`echo`** - Used to echo input commands; helpful for debugging scripts.
- **`noprompt`** - Turns off qip-cli prompts; useful for using **`qip-cli`** with other scripts.
- **`exitonerror`** Stops processing when the first error occurs. Useful for using **`qip-cli`** with other scripts.

A sample script that takes advantage of console mode is as follows:

```
qip-cli --console < getmaintainersfile
```

The *getmaintainers* file contains the following:

```
# Collect arin maintainers and put each in an output file
-c maintainer,get -u admin1 -p password -o "MyOrg" \
   -O maintjim.xml -n jim --type arin
-c maintainer,get -u admin1 -p password -o "MyOrg" \
   -O maintmidge.xml -n midge --type arin
exit
```

- When you are in console mode, you begin entering commands with the **-c** argument. You do not enter **qip-cli**.
- Input syntax allows using '#' for comments, and '\' for line continuation.

Automated CLI Options

Users who want to automate CLI actions, for example, with scripting, have several choices available.

1. A script can invoke one CLI at a time. Though easiest, this is the slowest method. When doing scripted CLI access, batch together as many operations as possible. When adding consecutive records, use the **--batch** feature. To run multiple CLIs at one time use **--console** (**--console** is not just for interactive commands.) For example:

   ```
   qip-cli --console <<!
         -c rule,get --outputfile=file1.xml
         -c rule,add --inputfile=file2.xml
   !
   ```

2. The CLI application is written in Java, and provides an entry point available to other Java applications. If you are familiar with Python scripting language, you may be interested in the Jython implementation. The internal CLI entry point is:

   ```
   com.lucent.qip.cli.Main.main(InputStream stdin, PrintStream
   stdout, PrintStream stderr, String[] argv)
   ```

3. There is a direct SOAP interface available that bypasses the CLIs. See the *VitalQIP Web Service Interface Specification* for details.

Input and Output Files

Many CLIs in this chapter can use an input file. Sometimes the input file is used to perform batch processing, other times an input file can contain just one record. Many CLIs can generate an output file which typically contains the results of a *search* or *get* operation. Output files are generated in XML, CSV or sometimes text (TXT) format, and you can use any of these formats for input files used with these CLIs.

XML input and output

A *search* action is designed to return multiple result records. The CLI uses *<object-stream>* and *</object-stream>* tags as a container to hold the results, even if only one record is returned. By comparison, a *get* action only returns a single record and does not use *<object-stream>* tags as a container. By default the output prints to the screen, but it can be redirected to a file by the **-O** (**--outputfile**) option. Postprocessing the output through a stylesheet is optionally selected using **--outputxsl**.

Neither *get* nor *search* CLIs return every XML element. They do not output those XML elements that do not have a value. The *get* action outputs all data related to the requested object. In contrast, the *search* action limits output only about the item being searched.

The CLI *modify* action can only process a single record. Therefore it cannot work directly with the output from a search command unless you edit it to ensure it contains only a single record without *<object-stream>* and *</object-stream>* tags. Alternately, you could use a *get* CLI to obtain a clean copy of the one desired record. The CLI *modify* action affects only the parameters specified in the input file.

When using a *modify* action, the **--replace** option allows you to replace the item you are modifying with a new item with the name you specify in the XML file. The following processing applies when modifying an element:

- If you modify a network component, such as a rule, you must include all values for the network component in your input file (such as a list of rule elements). If you do not, the values you do not include are set to null.
- Modify operates by overlaying the original element values with new ones read from your input file. If an XML element, for example **<description>**, is omitted from your input file then the original value for **<description>** remains untouched.
- If an XML element is present in your input, but contains no value, for example **<description></description>**, then the value for that element is set to null.
- If you try to set a required value to null, the CLI fails.
- To avoid problems, use a *get* CLI to obtain a clean copy of the element you want to modify, edit the values you want to change, and submit this file as an input record for your *modify* CLI.

A CLI *add* action (for example) is more flexible because it offers the option **--batch**. If **--batch**, is not specified, *add* behaves the same as modify and expects a single record, but with "--batch" specified, *add* processes the multiple records contained under an *<object-stream>* node.

Note: Using the wrong type of XML object in an input file prevents the CLI from running. When the wrong type of XML object is used, the following error occurs:

```
Error: Parsing XML: No such property
```

Use a standard XML editor to view and modify output files that you want to use as input for another CLI. Note that a rejectfile, which shows failed operations on some *add* CLIs, is also in XML format and can also be viewed with an XML editor.

Special characters in XML files, such as apostrophes and hyphens are represented according to HTML/XML standards. For example, an apostrophe (‘) is represented as **&apos.**

XML File Structure

Input and output files are based on the *VQIPTypes.xsd* file. This is XML Schema Definition for the “qip-cli” group of CLIs and all other SOAP clients that must connect to the VitalQIP Web Service. The file defines the structure for an XML document, the elements it uses, and the allowed content for each element. For example, an element’s definition may require you pick from among a predefined set of values, such as an object’s class. Other elements, which are defined as “**xsd:string**”, require you to assign a value, for example a subnet’s description.

The *VQIPTypes.xsd* file is located in the *xsd* subdirectory of the *$qiphome* directory.

When you run a get CLI, you can place the output in an XML file. The following is an example of XML output from the **pool,get** CLI. The command that generated the output is as follows:

```
qip-cli -c pool,get -u userid -p password -o VitalQIP --pool pool1
```

The output is as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SEED_POOL_REC>
  <allowPendingStatus>false</allowPendingStatus>
  <blockAddedNotify>true</blockAddedNotify>
  <childAllocationNotify>false</childAllocationNotify>
  <contact>admin1@lucent.com</contact>
  <freePendingBlockDays>0</freePendingBlockDays>
  <freePendingBlockWhen>FREE_AFTER_WHOIS</freePendingBlockWhen>
  <name>pool1</name>
  <optionalAttributeList>
    <udas>
      <name>das</name>
      <value>ed</value>
    </udas>
  </optionalAttributeList>
  <registry>ARIN</registry>
  <v4Algorithm>BEST_FIT_FROM_START</v4Algorithm>
  <v4MaxAllocation>0</v4MaxAllocation>
  <v4MinimumSparseSize>0</v4MinimumSparseSize>
  <v6Algorithm>BEST_FIT_FROM_START</v6Algorithm>
```

```
  <v6MaxAllocation>0</v6MaxAllocation>
  <v6MinimumSparseSize>0</v6MinimumSparseSize>
</SEED_POOL_REC>
```

You can use the output from this *get* record to add or modify a new pool. While you can edit the file manually, the preferred editing method is to use an XML editor with the schema set to *VQIPTypes.xsd*. Using the XML editor allows you to choose a value for each element, or, for elements to which you assign values, prompts you to enter a value. After modifying the file, you can save it and specify it as an inputfile for those CLIs that support one.

An XML document is defined as having exactly one root element. **qip-cli** does not attempt to parse anything in the file after that. For example, if you are working with a batch input file, anything after the **</object-stream>** is ignored. Similarly if you are working with a single record, then (for example), anything after </NODE_REC> is ignored. Therefore, forgetting to use **--batch**, and forgetting to use **<object-stream>** means that you can add multiple records to an input file and all but the first one is ignored.

> **Note:** If you do not specify a value for a required element, the CLI fails. If you do not specify a value for an optional element, the element's value is not changed. If you specify a value that does not match the datatype (for example, specifying a letter in a numeric field), the CLI generates an XML parsing error. The following example shows the input and the error when an empty string is used where an integer is required. The TTL field requires an integer value, but was input with a null value (**<TTL><TTL/>**).

```
$ qip-cli -c node,modify -IC:/N.xml
Error: Parsing XML: For input string: ""
---- Debugging information ----
required-type       : java.lang.Integer
cause-message       : For input string: ""
class               : com.lucent.qip.nb.server.NODE_REC
message             : For input string: ""
line number         : 9
cause-exception     : java.lang.NumberFormatException
```

Correct input is **<TTL>0<TTL/>**.

Namespace prefixes

Some XML editors may automatically insert namespace prefixes into the user's file. The XML parser used by **qip-cli** does not support namespaces, so input with namespace prefixes is rejected by **qip-cli**.

When namespaces are added by an XML editor, the XML contains the **qipws:** namespace prefix before each element as shown below:

```
<?xml version="1.0" encoding="UTF-8" ?>
<qipws:RULE_REC xmlns:qipws="http://alcatel-lucent.com/qip/nb/ws"
   >
      <qipws:name>myRule</qipws:name>
</qipws:RULE_REC>
```

For XML editors that insert namespace prefixes, it may be possible to disable that behavior by defining a default namespace either by appropriately configuring the tool, or by manually inserting the default namespace declaration: **xmlns="http://alcatel-lucent.com/qip/nb/ws"** as shown in the following example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<RULE_REC xmlns="http://alcatel-lucent.com/qip/nb/ws"
   xmlns:qipws="http://alcatel-lucent.com/qip/nb/ws"  >
<name>myRule</name>
</RULE_REC>
```

CSV input and output

If you specify CSV output from a CLI that creates an output file, the output is placed in a standard ASCII file. You can specify a header row in your output, so the columns are labeled. CSV files are easily importable into most standard spreadsheet programs, where you can then view and edit the data. CSV output, as opposed to XMLoutput, does not contain *<object-stream>* tags.

If you specify **csv**, the following suboptions are available:

- **textqualifier** - character used to surround fields that contain special characters. The default is a doublequote (**"**).
- **escapemode** - This option describes how to handle special characters in the data. Special characters specified with the **textqualifier** suboption are handled by this suboption. Allowable values are as follows:
  - **doubled** - Double up the text qualifier to represent an occurrence of the text qualifier. For example, if the **textqualifier** is a doublequote, and a doublequote appears in your data, the CSV output shows it as two doublequotes (**""**). Conversely, if you are using a CSV file as input, and you want a doublequote to appear in your data, you must ensure that your CSV input file shows the doublequote in your data as two doublequotes (**""**).
  - **backslash**. Use a backslash character before the text qualifier to represent an occurrence of the text qualifier. For example, if the **textqualifier** is a doublequote, and a doublequote appears in your data, the CSV output shows it as a backslash followed by a doublequote(**\"**). Conversely, if you are using a CSV file

as input, and you want a doublequote to appear in your data, you must ensure that your CSV input file shows the doublequote in your data as a backslash followed by a doublequote(**\"**).

- **columndelimiter** - allowable value is any keyboard character. This character signals a column break. The default is a comma (,).
- **recorddelimiter** - allowable value is any keyboard character. This character signals the end of a record. The default is a new line.
- **header** - if specified, then the first record of data contains the column headings that define the contents of the following records.
- **comment** - allowable value is any keyboard character. Lines starting with the specified character are treated as comments, not data. There is no default for this suboption. You can use a comment in your output files to note the csv format options you used to generate the file.
- **utf8** - if specified, then the file uses UTF-8 character encoding.

When using a *modify* action, the **--replace** option allows you to replace the item you are modifying with a new item with the name you specify in the CSV file. The following processing applies when modifying an element:

- If you modify a network component, such as a rule, you must include all values for the network component in your input file (such as a list of rule elements). If you do not, the values you do not include are set to null.
- Modify operates by overlaying the original element values with new ones read from your input file. For CSV files, if no value is specified between the column separators, the value is unchanged. If you want to change a value to null, you must use the special keyword **null**.
- If you try to set a required value to null, the CLI fails.

To avoid problems, use a *get* CLI to obtain a clean copy of the element you want to modify, edit the values you want to change, and submit this file as an input record for your *modify* CLI.

Help Options

CLIs provide several ways to access help on their syntax and allowable values:

- To see a list of syntax and allowable values, use the following syntax:

  qip-cli -c *<object>,<action>* -h provides a list of all available options for the object and action specified. An example is as follows:

  qip-cli -c pool,add -h returns the following:

  ```
  Usage:
  qip-cli -c pool,add [Login Options] --seed|--child
  -I <input_POOL_REC file> [--batch] [--rejectfile <file>]
  ```

- CLIs that have options that are limited to specific values allow you to see a list of those values. Use the following syntax:

  qip-cli -c *<object>,<action>* -*<option>* help

  To see all the available registry types for the **maintainer,get** CLI, use the following example:

  qip-cli -c maintainer,get -T help returns the following:

  `Legal values are: afrinic|arin|ripe|apnic`

- To see each help option on an individual line followed by its description you can use the following syntax:

  qip-cli --help=*<object>,<action>*

  You must use an equals sign (=) after the help argument. You cannot use a space.

  qip-cli --help=list,search returns the following:

```
-c, --command=<object,action>     Selects a specific CLI
   operation.
-h, --help
-o, --organization=<organization> The user's organization.
-O, --outputfile=<output filename>
--outputxsl=<XML stylesheets>     Use with --outputfile
-p, --password=<password>         The user's database password.
-T, --type=<type>
-u, --user=<user>                 The user ID for the database.
```

Typographical Conventions

Table 5-1 contains the typographic conventions for the Synopsis shown for each CLI.

Table 5-1 Typographical conventions

Convention	Meaning	Example
<angle brackets>	Variables for which you must substitute another value.	**qip-cli -c rule,add -I *<inputfile>***
[square brackets]	Variables in square brackets are optional. Variables that are not within square brackets are required.	**qip-cli -c rule,add [-O *<Outputfile>*]**
(parentheses)	Groups variables that work with one another	**(-N *<node ID>* --interfacename *<interface name>* -a *<IPv6 address>*)**

Convention	Meaning	Example
| vertical bar	Indicates a choice between variables or options.	-V4|-V6
\ backslash	Indicates that text continues on the next line	... -N <node ID> \ --interfacename <interfacename>

Address Management and Address Allocation

Address allocation is the process of acquiring and partitioning the address space according to the organization's logical boundaries and also distributing the address to organizations and customers down the hierarchy.

Address management is the process of dividing the address space into smaller, managed spaces (Network, Address Range, Subnet etc), and managing the individual IP addresses within the Subnets. Address Allocation module, as a result of certain block allocation, can create subnets and individual IP addresses in the Address Management areas.

Any address that is currently used by the Address Allocation (AA) module can only be used by the AA module in its entirety. That is if you need to create a subnet, then a block must be allocated and the post-allocation that is a part of the block allocation creates the required subnet. But the individual addresses within the subnet can be managed from the Address Management (AM) module.

However, if you are trying to create just a subnet (without going through the address allocation process), then the address space must be one of the addresses that is currently in Address Management (AM). You cannot create a subnet within the address space that is managed by AA. That generates the exception (error message) that you see below.

```
Error: Internal Error.  Please contact System Administrator.
   [Unable to create subnets.]

cause: [com.lucent.qip.exception.QIPRuntimeException: This
   address space is managed by Address Allocation, and can not be
   modified in Address Management.]
```

acdevice, add

`acdevice,add` allows you to add a device.

Synopsis

```
qip-cli --command acdevice,add --inputfile <filename> [--
   disableupdates] [--batch] [--rejectfile <filename>] <Login
   Options> [--format [xml|csv]]
```

Parameters

qip-cli **`-c acdevice,add`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`acdevice`**. Action is **`add`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing seed block record input for the CLI.
--batch		Enables batch processing on multiple records of the same type in an input file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--rejectfile	-r	File containing the records that failed processing. You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.
--disable updates		Do not send any MAC->User class updates to DHCP. Useful for bulk updates or if the user plans to push at the end of a series of operations.

Command line example

```
qip-cli --command acdevice,add -u qipadmin -p qipadmin -o Senthil
   --inputfile acdevice_add.xml --disableupdates --format xml
```

Sample input example

The following is a sample input file for adding a device:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <AC_DEVICE_REC>
  <macAddress>a1a1a1a1a1a1</macAddress>
  <acSubscriber>VitalQIP</acSubscriber>
  <userClass>Class1</userClass>
  <disableupdates>false</disableupdates>
  </AC_DEVICE_REC>
```

acdevice, modify

`acdevice,modify` allows you to modify a device.

Synopsis

```
qip-cli --command acdevice,modify -inputfile <filename> [--
   disableupdates] <Login Options> [--format [xml|csv]]
```

Parameters

qip-cli **`-c acdevice,modify`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`acdevice`**. Action is **`modify`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing seed block record input for the CLI.
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Long Option	Short Option	Allowable Values
--disable updates		Do not send any MAC->User class updates to DHCP. Useful for bulk updates or if the user plans to push at the end of a series of operations.

Command line example

```
qip-cli --command acdevice,modify -u qipadmin -p qipadmin -o
   Senthil --inputfile acdevice_add.xml --disableupdates --format
   xml
```

Sample input example

The following is a sample input file for modifying a device:

```
<?xml version="1.0" encoding="UTF-8" ?>
<AC_DEVICE_REC>
  <macAddress>a1a1a1a1a1a1</macAddress>
  <acSubscriber>VitalQIP</acSubscriber>
  <userClass>Class2</userClass>
  <disableupdates>false</disableupdates>
  </AC_DEVICE_REC>
```

acdevice, delete

acdevice,delete allows you to delete a device.

Synopsis

```
qip-cli -command acdevice,delete -mac <MAC Address> <Login
   Options>[--disableupdates]
```

Parameters

qip-cli **-c acdevice,delete** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **acdevice**. Action is **delete**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
MAC Address	--mac	MAC Address of the device
--disableupdates		Do not send any MAC->User class updates to DHCP. Useful for bulk updates or if the user plans to push at the end of a series of operations.

Command line example

```
qip-cli --command acdevice,delete -u qipadmin -p qipadmin -o
   Senthil --mac 192.201.63.251 --disableupdates
```

acdevice, get

`acdevice,get` allows you to get an ac device from a seed block.

Synopsis

```
qip-cli -c acdevice,get -mac <MAC Address> [--
   outputfile=<filename>] <Login Options> [--format=xml|csv]
```

Parameters

qip-cli **`-c acdevice,get`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`acdevice`**. Action is **`get`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
MAC Address	--mac	MAC Address of the device
--outputfile	-I	Name of the file containing record output for the CLI.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Command line example

```
qip-cli --command acdevice,get -u qipadmin -p qipadmin -o Senthil
   --mac 192.201.63.251 --output VitalQIP --format xml
```

Sample input example

The following is a sample input file for getting an ac device from a seed block:

```
<?xml version="1.0" encoding="UTF-8" ?>
<AC_DEVICE_REC>
  <macAddress>a1a1a1a1a1a1</macAddress>
  <acSubscriber>VitalQIP</acSubscriber>
  <userClass>Class1</userClass>
  <disableupdates>false</disableupdates>
  </AC_DEVICE_REC>
```

acdevice, search

acdevice,search allows you to search for an ac device from a seed block.

Synopsis

```
qip-cli -c acdevice,search [-mac <MAC Address>]
   [-a <IPv4 Address>]
     [--acsubscriber=<login>] [--host=<name>] [-d <domain>]
   [--userclass=<name>]
     [--uda=[group/]name[=value]] [--outputfile=<filename>] <Login
   Options>
   [--format=xml|csv]
```

Parameters

qip-cli **<user name, password> -c acdevice,search** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **acdevice**. Action is **search**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
Base Options		
MAC Address	--mac	MAC address of the device
IPv4 Address	-a	IPv4 address of the device which you are searching. You can use an asterisk (*) as a wildcard character to return a range of results.
acsubscriber		AC subscriber login (if the device is associated with a subscriber).
host		Host name of the object (if the device has acquired a lease)
domain	-d	Domain name of the object (if the device has acquired a lease)
userclass		Userclass associated with the device (if it has a device level user class)

Long Option	Short Option	Allowable Values
--uda	-U	**`--uda`** [[***`/component`***/]***`group`***/]***`name`***[=***`value`*** The following explains how UDAs are handled in the Web GUI compared to how they are handled when using them in a CLI. **Component** ***Optional.*** The component you specify must be defined under the infrastructure element for which this CLI is searching (for example, node, pool, or block). **Group** ***Optional***. The attribute group you specify must be defined within the infrastructure or the search fails. • When the GUI indicates **<NONE>** for a group, omitting a value for a group in the CLI means "not in a group". Example: **`--uda /node//state=NJ`**, or **`--uda state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute is not in a group. The first example specifies that the attribute is associated with a node. The second example does not specify what component the UDA is associated with. • When the GUI specifies **<ANY>**, using an asterisk (**`*`**) to specify the group in the CLI means that the attribute must be in a group, regardless of the group's name. Example: **`--uda */state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute must belong to a group, but the group is not specified. • When the GUI specifies **<IGNORE>**, for a group, using a dash (**`-`**) in the CLI means it does not matter if the UDA is in a group, or not. Example: **`--uda -/state=NJ`**. In this case, the attribute is "state" and the value is NJ. The attribute will be found regardless of whether it is in a group. **Name** This argument contains the name of the attribute, followed by an equals sign (=), followed by the value for which you are searching.

Long Option	Short Option	Allowable Values
--outputfile	-I	Name of the file containing record output for the CLI.
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Command line example

```
qip-cli --command acdevice,search -u qipadmin -p qipadmin -o
   Senthil --mac 192.201.63.251 --output VitalQIP --format xml
```

acsubscriber, add

`acsubscriber,add` allows you to add a subscriber.

Synopsis

```
qip-cli --command acsubscriber,add --inputfile <filename> [--
   disableupdates] [--batch]
   [--rejectfile <filename>] <Login Options> [--format [xml|csv]]
```

Parameters

qip-cli **`-- command acsubscriber, add`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`acsubscriber`**. Action is **`add`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
Base Options		
--inputfile	-I	Name of the file containing seed block record input for the CLI.
--disable updates		Do not send any MAC->User class updates to DHCP. Useful for bulk updates or if the user plans to push at the end of a series of operations.
--batch		Enables batch processing on multiple records of the same type in an input file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--rejectfile	-r	File containing the records that failed processing. You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.

Command line example

```
qip-cli --command --inputfile acsubscriber_add.xml -u qipadmin -p
   qipadmin -o Senthil --disableupdates --format xml
```

Sample input example

The following is a sample input file for adding a subscriber:

```
<?xml version="1.0" encoding="UTF-8" ?>
<AC_SUBSCRIBER_REC>
  <acSubscriber>JohnSmith</acSubscriber>
  <acSubscriberFirstName>John</acSubscriberFirstName>
  <acSubscriberLastName>Smith</acSubscriberLastName>
<acSubscriberPassword>02c8b256429d3f5ae5ca428b4af02f2cc9d1
   </acSubscriberPassword>
  <maxDevices>0</maxDevices>
  <acSubscriberStatus>ACTIVE</acSubscriberStatus>
  <disableupdates>false</disableupdates>
  </AC_SUBSCRIBER_REC>
```

acsubscriber, modify

`acsubscriber,modify` allows you to modify a subscriber.

Synopsis

```
qip-cli --command acsubscriber,modify
   -inputfile <filename> [--disableupdates] <Login Options>
   [--format [xml|csv]]
```

Parameters

qip-cli **`-c acsubscriber,modify`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`acsubscriber`**. Action is **`modify`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
Base Options		
--inputfile	-I	Name of the file containing seed block record input for the CLI.
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Long Option	Short Option	Allowable Values
--disable updates		Do not send any MAC->User class updates to DHCP. Useful for bulk updates or if the user plans to push at the end of a series of operations.

Command line example

```
qip-cli --command acsubscriber,modify -u qipadmin -p qipadmin -o
   Senthil --inputfile acsubscriber_modify.xml --disableupdates --
   format xml
```

Sample input example

The following is a sample input file for modifying a subscriber:

```
<?xml version="1.0" encoding="UTF-8" ?>
<AC_SUBSCRIBER_REC>
  <acSubscriber>JohnSmith</acSubscriber>
  <acSubscriberFirstName>John</acSubscriberFirstName>
  <acSubscriberLastName>Smith</acSubscriberLastName>
<acSubscriberPassword>02c8b256429d3f5ae5ca428b4af02f2cc9d1
   </acSubscriberPassword>
  <maxDevices>5</maxDevices>
  <acSubscriberStatus>ACTIVE</acSubscriberStatus>
  <disableupdates>false</disableupdates>
  </AC_SUBSCRIBER_REC>
```

acsubscriber, delete

`acsubscriber,delete` allows you to delete a subscriber.

Synopsis

```
qip-cli -c acsubscriber,delete <Login Options> -acsubscriber =
   <login>
   [--disableupdates]
```

Parameters

qip-cli **`-c acsubscriber,delete`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`acsubscriber`**. Action is **`delete`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
login		AC Subscriber login (if the device is associated with a subscriber)
--disable updates		Do not send any MAC->User class updates to DHCP. Useful for bulk updates or if the user plans to push at the end of a series of operations.

Command line example

```
qip-cli --command acsubscriber,delete -u qipadmin -p qipadmin -o
   Senthil --acsubscriber Vitalqip --disableupdates
```

acsubscriber, get

acsubscriber,get allows you to get the details of a subscriber.

Synopsis

```
qip-cli -c acsubscriber,get -mac <MAC Address> [--
   outputfile=<filename>] <Login Options>[--format=xml|csv]
```

Parameters

qip-cli **-c acsubscriber,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **acsubscriber**. Action is **get**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
Base Options		
MAC Address	--mac	MAC Address of the device
--outputfile	-I	Name of the file containing record output for the CLI.
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **--format** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **xml** (default) **csv** If you specify **csv**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Command line example

```
qip-cli --command acsubscriber,get -u qipadmin -p qipadmin -o
   Senthil --mac 192.201.63.251 --output VitalQIP --format xml
```

Sample output example

```
<?xml version="1.0" encoding="UTF-8" ?>
<AC_SUBSCRIBER_REC>
  <acSubscriber>senthilks</acSubscriber>
  <acSubscriberFirstName>Senthil</acSubscriberFirstName>
  <acSubscriberLastName>Kumar</acSubscriberLastName>
  <acSubscriberPassword>02c8b256429d3f5ae5ca428b4af02f2cc9d1
   </acSubscriberPassword>
  <maxDevices>0</maxDevices>
  <acSubscriberStatus>ACTIVE</acSubscriberStatus>
  <disableupdates>false</disableupdates>
  </AC_SUBSCRIBER_REC>
```

acsubscriber, list

`acsubscriber,list` lists all the unique user classes associated with a subscriber including any user classes of devices associated with that subscriber.

Synopsis

```
qip-cli -c acsubscriber,list --acsubscriber=<login>
     [--outputfile=<filename>] <Login Options> [--format=xml|csv]
```

Parameters

qip-cli **`-c acsubscriber,list`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`acsubscriber`**. Action is **`list`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-I	Name of the file containing record output for the CLI.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Command line example

```
qip-cli --command acsubscriber,list -u qipadmin -p qipadmin -o
   Senthil --output VitalQIP --format xml
```

acsubscriber, search

`acsubscriber,search` allows you to search for a subscriber .

Synopsis

```
qip-cli -c acsubscriber,search [-acsubscriber <Subscriber Login>]
   [-lastname <AC Subscriber Last Name>][--userclass<userclass>]
     [--status=[active|inactive]] [--uda [group/]name[=value]] --
   outputfile <filename> <Login Options> [--format [xml|csv]]
```

Parameters

qip-cli **`-c acsubscriber,search`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`acsubscriber`**. Action is **`search`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
Base Options		
Subscriber Login		AC Subscriber login (if the device is associated with a subscriber)
userclass		Userclass associated with the device (if it has a device level user class)
--status		Indicates the status of the subscriber — Active or Inactive.

Long Option	Short Option	Allowable Values
--uda	-U	**`--uda`** [[***`/component`***/]***`group`***/]***`name`***[=***`value`***

The following explains how UDAs are handled in the Web GUI compared to how they are handled when using them in a CLI.

Component

Optional. The component you specify must be defined under the infrastructure element for which this CLI is searching (for example, node, pool, or block).

Group

Optional. The attribute group you specify must be defined within the infrastructure or the search fails.

- When the GUI indicates **<NONE>** for a group, omitting a value for a group in the CLI means "not in a group". Example: **`--uda /node//state=NJ`**, or **`--uda state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute is not in a group. The first example specifies that the attribute is associated with a node. The second example does not specify what component the UDA is associated with.
- When the GUI specifies **<ANY>**, using an asterisk (**`*`**) to specify the group in the CLI means that the attribute must be in a group, regardless of the group's name. Example: **`--uda */state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute must belong to a group, but the group is not specified.
- When the GUI specifies **<IGNORE>**, for a group, using a dash (**`-`**) in the CLI means it does not matter if the UDA is in a group, or not. Example: **`--uda -/state=NJ`**. In this case, the attribute is "state" and the value is NJ. The attribute will be found regardless of whether it is in a group.

Name

This argument contains the name of the attribute, followed by an equals sign (=), followed by the value for which you are searching.

Long Option	Short Option	Allowable Values
--outputfile	-I	Name of the file containing record output for the CLI.
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Command line example

```
qip-cli --command acsubscriber,search -u qipadmin -p qipadmin -o
   Senthil --acsubscriber Vitalqip
   --status ACTIVE --output VitalQIP --format xml --userclass TEST
```

addressrange,add

`addressrange,add` allows you to add an address range to a seed pool.

Synopsis

```
qip-cli -c addressrange,add <login options>
   -I <input_ADDRESS_RANGE_REC_file>.xml|.csv
   [--batch]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er]
   [,textqualifier=<char>][,utf8]])
   [--rejectfile <filename.xml|.csv>] [-v] [-h|?]
```

Parameters

qip-cli -c **`addressrange,add`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`addressrange`**. Action is **`add`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing seed block record input for the CLI.
--batch		Enables batch processing on multiple records of the same type in an input file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--rejectfile	-r	File containing the records that failed processing. You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.

Command line example

```
qip-cli -c addressrange,add -u admin1 -p password -o VitalQIP
   -I newpools.csv --format csv,comment="#" --batch
```

You can use an output file from the **`addressrange,get`** CLI as a model for an input file for this CLI.

addressrange,delete

`addressrange,delete` allows you to delete an address range from a seed block.

Synopsis

```
qip-cli -c addressrange,delete <login options>
   --parentblock <parentblock_address> -A <address_range/prefix>
   [-v] [-h|?]
```

Parameters

qip-cli -c **`addressrange,delete`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`addressrange`**. Action is **`delete`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.
Specific Options		
--address	-A	Start address and prefix of the first subnet in the address range. Use address/prefix notation.
--parentblock		Address of the address range's parent block.

Command line example

```
qip-cli -c addressrange,delete -u admin1 -p password -o testorg
   --parentblock 2300:11:: -A 2300:11::/40
```

addressrange,get

`addressrange,get` allows you to get an address range from a seed block.

Synopsis

```
qip-cli -c addressrange,get <login options>
   [-O <outputfile>.xml|csv] [--outputxsl <file1,file2>]
   --parentblock <parentblock_address> -A <address_range/prefix>
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`addressrange,get`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`addressrange`**. Action is **`get`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing the output from the CLI.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--address	-A	Start address and prefix of the first subnet in the address range. Use address/prefix notation.
--parentblock		Address of the address range's parent block.

Command line example

XML

```
qip-cli -c addressrange,get -u admin1 -p password -o VitalQIP
   -O addrange.xml --parentblock 2300:11:: -A 2300:11::/40
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<ADDRESS_RANGE_REC>
  <name>2300:11::</name>
  <parentBlockAddr>2300:11::</parentBlockAddr>
  <prefixLength>40</prefixLength>
  <startAddress>2300:11::</startAddress>
</ADDRESS_RANGE_REC>
```

Command line example

CSV

```
qip-cli -c addressrange,get -u admin1 -p password -o VitalQIP
   -O addrange.csv -A 2001:db8::/33 --parentblock 2001:db8::
```

Sample Output File

```
2001:db8::,33,DocTestAddrRange1,2001:db8::
```

addressrange,modify

`addressrange,modify` allows you to modify an address range.

Synopsis

```
qip-cli -c addressrange,modify <login options> -I <inputfile>
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er][,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`addressrange,modify`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`addressrange`**. Action is **`modify`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing address range input for the CLI.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.

Command line example

```
qip-cli -c addressrange,modify -u admin1 -p password -o testorg
   -I addrange.xml
```

You can use output from the **`addressrange,get`** CLI as a template for your input file for this CLI.

amseedblock,add

amseedblock,add allows you to add a seed block. This CLI uses the address management function, which allows you to work with seed blocks without having address allocation enabled.

Synopsis

```
qip-cli -c amseedblock,add <login options>
   -I <input_V6_SEED_BLOCK_REC_file> [--batch]
   [--rejectfile <filename.xml|.csv>]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er][,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **amseedblock,add** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **amseedblock**. Action is **add**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing seed block record input for the CLI.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--batch		Enables batch processing on multiple records of the same type in an input file.
--rejectfile	-r	File containing the records that failed processing. You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.

Command line example

```
qip-cli -c amseedblock,add -u admin1 -p password -o VitalQIP
   -I seedblocks.xml --batch
```

You can use output from the **`amseedblock,get`** CLI as a template for your input file for this CLI.

amseedblock,delete

amseedblock,delete allows you to delete a seed block. This CLI uses the address management function, which allows you to work with seed blocks without having address allocation enabled.

Synopsis

```
qip-cli -c amseedblock,delete <login options>
   --address <address/prefix> [-v] [-h|?]
```

Parameters

qip-cli -c **amseedblock,delete** recognizes the following parameters:

Long Option	Short Option	Allowable Values
Base Options		
--command	-c	Syntax is <object>,<action>. Object is **amseedblock**. Action is **delete**.
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.
Specific Options		
--address	-A	Start address and prefix of the seed block. Use address/prefix notation.

Command line example

```
qip-cli -c amseedblock,delete -u admin1 -p password -o VitalQIP
   --address 2300:11::/40
```

amseedblock,get

amseedblock,get allows you to get a seed block. This CLI uses the address management function, which allows you to work with seed blocks without having address allocation enabled.

Synopsis

```
qip-cli -c amseedblock,get <login options> -A <address/prefix>
   [-O <outputfile.xml>] [--outputxsl <file1,file2>]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header][,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **amseedblock,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **amseedblock**. Action is **get**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing the output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.
Specific Options		
--address	-A	Start address and prefix of the fseed block. Use address/prefix notation.

Command line example

```
qip-cli -c amseedblock,get -u admin1 -p password -o VitalQIP
   -A 1400:11::/32 -O seedblock.xml
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<V6_SEED_BLOCK_REC>
  <prefixLength>32</prefixLength>
  <startAddress>1400:11::</startAddress>
</V6_SEED_BLOCK_REC>
```

block,addtopool

`block,addtopool` allows you to add a block to a pool.

Synopsis

```
qip-cli -c block,addtopool <login options>
   -I <input_SEED_BLOCK_REC_file>.xml|.csv [--batch]
   [--rejectfile <filename.xml|.csv>]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er][,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`block,addtopool`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`block`**. Action is **`addtopool`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing seed block record input for the CLI.
--batch		Enables batch processing on multiple records of the same type in an input file.
--rejectfile	-r	File containing the records that failed processing. You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.

Command line example

```
qip-cli -c block,addtopool -u admin1 -p password -o VitalQIP
   -I newpools.xml --batch
```

You can use output from the block,get command as a basis for input for this command.

block,allocate

`block,allocate` allows you to allocate a block.

When this command successfully runs, a message displays confirming that the block was allocated.

Synopsis

```
qip-cli -c block,allocate <login options>
   (-I <input_child_AllocateBlockToPoolRequest_file.xml>) | \
   (-E -I <input_child_ExplicitBlockAllocRequest_file>.xml|.csv>)
   | \ (-Q -I <input_QuickBlockAllocRequest_file>.xml|.csv>)
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`block,allocate`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`block`**. Action is **`allocate`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing input for the CLI.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.
Specific Options		
--explicitblockallocation	-E	Explicit block allocation. Specify this option if you want to allocate the exact size of the block.
--quickblocallocation	-Q	Quick block allocation. Specify this option when you want to allocate a block and child pools using this CLI.

Command line example

```
qip-cli -c block,allocate -u admin1 -p password -o VitalQIP -E
   -I blockallocspecs.xml
```

Sample input example

The following is a sample input file for explicit block allocation

```
<?xml version="1.0" encoding="UTF-8"?>
<ExplicitBlockAllocRequest>
            <startAddress>aa:bb:cc:dd::</startAddress>
            <length>128</length>
            <pool>
                  <name>v6ChildPool</name>
                  <parentName>v6SeedPool</parentName>
            </pool>
            <blockStatus>FREE</blockStatus>
            <addressType>IPv6</addressType>
</ExplicitBlockAllocRequest>
```

You can use output from the **`block,get`** CLI as a template for your input file for this CLI.

Important! When using CSV input, columns without headings assigned to them can cause the CLI to fail with the following message:

```
Error: Parsing input file:  : A CSV map definition for column <##>
   is missing.
```

block,deletefrompool

`block,deletefrompool` allows you to delete a block from a pool.

Synopsis

```
qip-cli -c block,deletefrompool <login options>
   --block <IP/Prefix> --pool <poolname> [-v] [-h|?]
```

Parameters

qip-cli -c **`block,deletefrompool`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`block`**.
		Action is **`deletefrompool`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--block	-A	Block address written as <IP address/prefix length>
--pool	-P	Name of the pool to which the block is being added.

Command line example

```
qip-cli -c block,deletefrompool -u admin1 -p password -o VitalQIP
   --block 2300:11::/32 --pool acctng
```

block,expand

`block,expand` allows you to double the size of a block in a pool.

Synopsis

```
qip-cli -c block,expand <login options> --block <IP/Prefix
   --pool <poolname> --parentpool <parentpoolname> [-v] [-h|?]
```

Parameters

qip-cli -c **`block,expand`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`block`**. Action is **`expand`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--block	-A	Block address written as <IP address/prefix length>
--pool	-P	Name of the pool containing the block to be expanded.
--parentpool	-L	Name of the parent pool.

Command line example

```
qip-cli -c block,expand -u admin1 -p password -o VitalQIP
   --block 2300:11::/32 --pool devlpmnt2 --parentpool devlpmnt
```

block,free

`block,free` allows you to free a block from a pool.

Synopsis

```
qip-cli -c block,free <login options>
   --A <startAddr/prefixLength> --pool <poolname>
   --parentpool <parentpoolname> [--returntoparent] [-v] [-h|?]
```

Parameters

qip-cli -c **`block,free`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`block`**. Action is **`free`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--block	-A	Block address written as <IP address/prefix length>
--pool	-P	Name of the pool containing the block to be freed.
--parentpool	-L	Name of the parent pool.
--returntoparent		Specifies that the block is returned to the parent.

Command line example

```
qip-cli -c block,free -u admin1 -p password -o VitalQIP
   --block 2300:11::/32 --pool devlpmnt1 --parentpool devlpmnt
```

block,freepending

`block,freepending` allows you to free blocks that have a status of pending at a specified level .

Synopsis

```
qip-cli -c block,freepending <login options> [--override]
   (--level all|organization) | (--level pool [-P <poolName>
   [--parentpool <poolName>]]) | (--level block [-P <poolName>
   [--parentpool <poolName>] [-A <blockName>]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`pool,freepending`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`block`**.
		Action is **`freepending`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--pool	-P	Name of the pool that contains the pending blocks.
--parentpool	-L	Name of the parent pool.
--block	-A	Block address written as <IP address/prefix length>
--override		

Long Option	Short Option	Allowable Values
--level		Value indicating the level of the pool to be checked for pending blocks. Allowable values are as follows: • **`all`** Checks all blocks through all organizations the user is authorized to access. • **`organization`** Checks all blocks through the organization used at login. • **`pool`** Checks all blocks in the pool specified. Pool must be within the organization used at login. Pool name and parent pool are required. • **`block`** Checks one specific block. A block, pool, (and login org) are used. Parent pool is required.

Command line example

```
qip-cli -c block,freepending -u admin1 -p password -o VitalQIP
   --level all
```

block,get

`block,get` allows you to obtain an XML or CSV representation of the contents of a block..

Synopsis

```
qip-cli -c block,get <login options> [-O <output_file>.xml|.csv]
   [--outputxsl file1,file2] --block <IP/Prefix> --pool <poolname>
   [--parentpool <parentpoolname>]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`block,get`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`block`**. Action is **`get`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h | -?	Displays syntax and Options for this CLI.
Specific Options		
--block	-A	Block address written as <IP address/prefix length>
--pool	-P	Name of the pool containing the block you want to get.
--parentpool	-L	Name of the parent pool.

Command line example

XML

```
qip-cli -c block,get -u admin1 -p password -o VitalQIP
   --parentpool pool1 -P  cpool1 -A 2300:11::/52
   -O blockoutput.xml
```

Output example

```
<?xml version="1.0" encoding="UTF-8" ?>
<ADDRESS_BLOCK_REC>
  <addressType>IPv6</addressType>
  <blockStatus>USED</blockStatus>
  <optionalAttributeList>
```

```
    <groups>
      <udas>
        <name>block_name</name>
      </udas>
      <udas>
        <name>das</name>
      </udas>
      <name>block_group</name>
    </groups>
    <groups>
      <udas>
        <name>das</name>
      </udas>
      <name>das_group</name>
    </groups>
    <udas>
      <name>block_name</name>
    </udas>
  </optionalAttributeList>
  <parentAddressBlock>
    <addressType>IPv6</addressType>
    <prefixLength>52</prefixLength>
    <startAddress>2300:11::</startAddress>
  </parentAddressBlock>
  <pool>
    <name>cpool1</name>
    <parentName>pool1</parentName>
  </pool>
  <prefixLength>52</prefixLength>
  <startAddress>2300:11::</startAddress>
</ADDRESS_BLOCK_REC>
```

CSV

qip-cli -c block,get -u admin1 -p password -o VitalQIP --block 2006:ad1::/32 --pool DOCS --format csv -O blockoutput.csv

Output example

```
2006:ad1::,32,IPv6,DOCS,,FREE,,,,Atlantic/NJ
   Views=None+Atlantic/PA Views=None+NewYor
k/HQ Views=NY+NorthEast/CN Views=None+NorthEast/MA
   Views=None+NorthEast/RI Views=None+ID+ID No,,80
```

block,merge

`block,merge` allows you to merge two blocks within the same parent pool.

Synopsis

```
qip-cli -c block,merge <login options> --block <IP/Prefix>
   --pool <poolname> --toblock <IP/Prefix> --parentpool <poolname>
   [-v] [-h|?]
```

Parameters

qip-cli -c **`block,merge`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`block`**.
		Action is **`merge`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--block	-A	Block address written as <IP address/prefix length>
--parentpool	-L	Name of the block's parent pool.
--pool	-P	Name of the pool containing the blocks to be merged.
--toblock		Target name of the block to which you are merging blocks

Command line example

```
qip-cli -c block,merge -u admin -p password -o VitalQIP
   --block 2300:11::/32 --pool devlpmnt --toblock 1800:11::/32
   --parentpool devlpmnt
```

block,modify

`block,modify` allows you to modify either a seed block or an address block.

Synopsis

```
qip-cli -c block,modify <login options> --seed|--child
   -I <input_SEED_BLOCL_REC_or_ADDERSS_BLOCK_REC_file>.xml|.csv>
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`block,modify`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`block`**. Action is **`modify`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing input for the CLI.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--seed	-S	Indicates the block is a seed block.
--child	-C	Indicates the block is a child block.

Command line example

```
qip-cli -c block,modify -u admin1 -p password -o VitalQIP --child
   -I blockmods.xml
```

You can use output from the **`block,get`** CLI as a template for your input file for this CLI.

block,move

`block,move` allows you to move a block.

Synopsis

```
qip-cli -c block,move <login options> --block <IP/Prefix>
   --pool <poolname> [--parentpool <poolname>]
   --topool <destinationpoolname> [-v] [-h|?]
```

Parameters

qip-cli -c **`block,move`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`block`**.
		Action is **`move`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--block	-A	Block address written as <IP address/prefix length>
--pool	-P	Name of the pool containing the block to be moved.
--parentpool	-L	Name of the parent pool.
--topool		Name of destination pool.

Command line example

```
qip-cli -c block,move -u admin1 -p password -o VitalQIP
   --block 2300:11::/32 --pool humanres1 --topool 1800:11::/32
```

block,recover

`block,recover` allows you to recover a block in a pending state. Upon a successful operation, the block status is changed from “Pending” back to “Used”..

Synopsis

```
qip-cli -c block,recover <login options>
   --inputfile <RecoverBlockRequest> [-v] [-h|?]
```

Parameters

qip-cli -c **`block,recover`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`block`**.
		Action is **`recover`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--inputfile	-I	Input file name

Command line example

```
qip-cli -c block,recover -u admin1 -p password -o VitalQIP
   --inputfile blockstorecover
```

You can use output from the info,get command as a model for the input file for this command. See “info,get” (p. 5-115) for additional information.

block,renumber

block,renumber allows you to change an IPv6 block's IP address.

Note: This CLI can only be used with IPv6 blocks.

Renumbering a block just changes the start address of a Used block. You can only renumber a Used block to a Free or a Reserved block in the same child pool. The resulting block status is Used. For example you have the following IPv6 blocks under a child pool:

```
2001:1000:1::/64(Used)
2001:1000:2::/64(Free)
2001:1000:3:;/64(Reserved)
2001:1000:4::/64(Used)
```

You can renumber the `2001:1000:4::/64(Used)` block, to the following list of address blocks:

```
2001:1000:2::/64(Free)
2001:1000:3::/64(Reserved)
```

If you select `2001:1000:3::/64(Reserved)` then `2001:1000:1::/64(Used)` is removed and your new block is `2001:1000:3::/64(Used)`.

Synopsis

qip-cli -c block,renumber *<login options>* --block *<IP/Prefix>*
--pool *<poolname>* --parentpool *<parentpoolname>*
--toblock *<blockname>* ([--when *<date H:S:M>*|NOW
--jobdescription *<text_description_of_job>*
--sendemail to=*<email1>*,*<email2>*,cc=*<email1>*,*<email2>*]) [-v]
[-h|?]

Parameters

qip-cli -c **block,renumber** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **block**. Action is **renumber**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.

Long Option	Short Option	Allowable Values
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--block	-A	IPv6 block address.
--pool	-P	Name of the pool containing the block to be renumbered.
--parentpool	-L	Name of the parent pool.
--toblock		New IPv6 block address.
--when		Specifies when to run this CLI. Allowable value is as follows: ***<MM/DD/YYYY H:M:S>*** or ***<DD/MM/YYYY H:M:S>***- CLI runs at the date and time specified. Date format is set based on user's location. **NOW** - CLI runs immediately in background.
--jobdescription		A text description of the job. You can only use this option if you have set a value for the **--when** option.
--sendemail		This option allows you to specify both "To" and "CC" recipients who receive an email when the job either completes normally or terminates prematurely. you can only use this option if you have set a value for the **--when** option. This option has the following suboptions: **to** a list of comma-separated email addresses that appear in the "To" address of the email **cc** a list of comma-separated email addresses that appear in the "cc" address of the email.

Command line example

```
qip-cli -c block,renumber -u admin1 -p password -o VitalQIP
   --block 2300:0:c0/42 --pool humanres1 --parentpool humanres
   --toblock 2300:0:80/42 --"when" 05/08/2009 12:12:18
   --jobdescription "Renum 2300 block"
   --sendemail to=user@abcd.com,cc=admin@abcd.com
```

block,return

block,return allows you to return a free block to a pool.

Synopsis

```
qip-cli -c block,return <login options> --block <IP/Prefix>
   --pool <poolname> --parentpool <parentpoolname>
```

Parameters

qip-cli -c **block,return** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **block**. Action is **return**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--block	-A	Block address written as <IP address/prefix length>
--pool	-P	Name of the pool to which the block is being returned.
--parentpool	-L	Name of the parent pool.

Command line example

```
qip-cli -c block,return -u admin1 -p password -o VitalQIP
   --block 2300:11::/32 --pool devlp3 --parentpool devlpmnt
```

block,search

`block,search` allows you to search for blocks.

Synopsis

You can search for blocks either by registry type, or by one or more of the following:

- Block address,
- Pool name
- Block status
- User-defined attribute (UDA)

The following examples show syntax for each.

Search by address:

```
qip-cli -c block,search <login options>
   [-O <output_file>.xml|.csv>] [--outputxsl <file1,file2>]
   --block <IP/Prefix> --ipv4v6 4|6 [--pool <poolname>]
   [--status <status1>,<status2>...]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Search by registry type:

```
qip-cli -c block,search <login options>
   [-O <output_file>.xml|.csv]
   [[--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er]
   [,textqualifier=<char>][,utf8]]) -V 4|6
```

Search by UDA:

```
qip-cli -c block,search <login options>
   [-A <startAddr>/[<prefixLength>]] [-V 4|6]] [-P <pool>]
   [--status <status1>,<status2>...]
   [--uda [[/object/]group/]name[=value]]...)
   [--outputfile <output_file>.xml|.csv]
   [--outputxsl <file1,file2>...]]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`block,search`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`block`**. Action is **`return`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		

Long Option	Short Option	Allowable Values
--block	-A	Block address written as <IP address/prefix length> **Note:** When you are performing a block search, you can use an asterisk (*) as a wildcard character in your searches. The asterisk cannot be the first character of the block name, or the only character specified. You can specify additional characters after the asterisk. For IPv6 addresses, you cannot use both the asterisk and compressed address notation.
--ipv4v6	-V	Type of block. Allowable values are: 4 - IPv4 block 6 - IPv6 block
--pool	-P	Name of the pool where you are searching for a block. **Note:** When you are performing a pool search, you can use an asterisk (*) as a wildcard character in your searches. The asterisk cannot be the first character of the pool name, or the only character specified. You can specify additional characters after the asterisk. For IPv6 addresses, you cannot use both the asterisk and compressed address notation.
--status	-s	Status of the block. Allowable values are as follows: • Allocated • Free • Next • Origin • Pending • Reserved • Selected • Site • Used You can use more than one status. You can enter multiple statuses either as a comma-separated list(**`--status=free,next`**), or as multiple options (**`--status=free --status=next`**)

Long Option	Short Option	Allowable Values
--uda		**`--uda`** [[***`/component`***/]***`group`***/]***`name`***[=***`value`***

The following explains how UDAs are handled in the Web GUI compared to how they are handled when using them in a CLI.

Component

Optional. The component you specify must be defined under the infrastructure element for which this CLI is searching (for example, node, pool, or block).

Group

Optional. The attribute group you specify must be defined within the infrastructure or the search fails.

- When the GUI indicates **<NONE>** for a group, omitting a value for a group in the CLI means "not in a group". Example: **`--uda /node//state=NJ`**, or **`--uda state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute is not in a group. The first example specifies that the attribute is associated with a node. The second example does not specify what component the UDA is associated with.
- When the GUI specifies **<ANY>**, using an asterisk (**`*`**) to specify the group in the CLI means that the attribute must be in a group, regardless of the group's name. Example: **`--uda */state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute must belong to a group, but the group is not specified.
- When the GUI specifies **<IGNORE>**, for a group, using a dash (**`-`**) in the CLI means it does not matter if the UDA is in a group, or not. Example: **`--uda -/state=NJ`**. In this case, the attribute is "state" and the value is NJ. The attribute will be found regardless of whether it is in a group.

Name

This argument contains the name of the attribute, followed by an equals sign (=), followed by the value for which you are searching.

Command line example

The following example searches for a block by IP Address

```
qip-cli -c block,search -u admin -p password
   -o "VitalQIP Organization" --block 192.0.2.0/24 --ipv4v6 4
```

Output example


```
<?xml version="1.0" encoding="UTF-8" ?>
<object-stream>
<SEED_BLOCK_REC>
  <startAddress>192.0.2.0</startAddress>
  <addressType>IPv4</addressType>
  <blockStatus>FREE</blockStatus>
  <optionalAttributeList>
    <udas>
      <name>CN Views</name>
      <value>None</value>
    </udas>
    <udas>
      <name>HQ Views</name>
      <value>NY</value>
    </udas>
    <udas>
      <name>ID</name>
      <value>EmployeeID</value>
    </udas>
    <udas>
      <name>ID No</name>
      <value>000002</value>
    </udas>
    <udas>
      <name>MA Views</name>
      <value>None</value>
    </udas>
    <udas>
      <name>NJ Views</name>
      <value>None</value>
    </udas>
    <udas>
      <name>PA Views</name>
      <value>None</value>
    </udas>
    <udas>
      <name>RI Views</name>
      <value>None</value>
    </udas>
  </optionalAttributeList>
  <prefixLength>24</prefixLength>
  <threshold>80</threshold>
  <pool>
    <name>DOCS</name>
```

```
  </pool>
</SEED_BLOCK_REC>
</object-stream>
```

block,split

`block,split` allows you to split an IPv6 block.

Synopsis

```
qip-cli -c block,split <login options> --block <IP/Prefix>
   --pool <poolname> --parentpool <parentpoolname>
   --tolength <block prefix> [-v] [-h|?]
```

Parameters

qip-cli -c **`block,split`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`block`**.
		Action is **`split`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--block	-A	Block address written as <IP address/prefix length>
--pool	-P	Name of the pool that contains the block.
--parentpool	-L	Name of the parent pool.
--tolength		The prefix length of the new block

Command line example

```
qip-cli -c block,split -u admin1 -p password -o VitalQIP
   --block 2300:11::/32 --pool eastcoast --parentpool usa
   --tolength /33
```

cdnssec, add

cdnssec,add adds a key for the zone name.

Synopsis

```
qip-cli -cdnssec,add -inputfile=<filename>
   --keyfilepath=<directory containing key files>[--batch]
   [-dskey]
```

Parameters

qip-cli -c **cdnssec,add** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **cdnssec**. Action is **add**.
Base Options		
--inputfile	-I	Name of the file containing input for the CLI.
--keyfilepath		Key File will be generated in the path specified.
--batch		Enables batch processing on multiple records of the same type in an input file.
--dskey		Specifies the DS record that needs to be added to the key. The DS record is read from a file named **dsset.<zone name>** and should be present in the path specified in **-keyfilepath**.

Command line example

```
qip-cli -cdnssec,add --inputfile=domain.in --keyfilepath=/keys
```

Sample input example

The following is a sample input file for adding a key to the zone name.

```
<?xml version="1.0" encoding="UTF-8" ?>
<DNSSEC_KEY_REC>
<keyName>lucent.com</keyName>
<keyTag>30388</keyTag>
<algorithm>1</algorithm>
<bitSize>512</bitSize>
<privateKeyFileName>Klucent.com.+001+30388.private</privateKeyFil
  eName>
<publicKeyFileName>Klucent.com.+001+30388.key</publicKeyFileName>
<ksk>true</ksk>
<deployDate>2009-05-05 16:40:31.0</deployDate>
<expireDate>2009-12-06 16:40:31.0</expireDate>
</DNSSEC_KEY_REC>
```

cdnssec, delete

`cdnssec,delete` deletes the keys specified in the input file from the VitalQIP database.

Synopsis

```
qip-cli -cdnssec,delete --dnsseckeyname=<keyname> --
   keytag=<keyTag>
```

Parameters

qip-cli -c **`cdnssec,delete`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`cdnssec`**. Action is **`delete`**.
Base Options		
--dnsseckeyname		Retrieves the DNSSEC key for the given keyname.
--keytag		Specifies the key tag generated by the key generation utility for each key. It is the last numeric number following the "+" sign in the output generated or the key file names.

Command line example

```
qip-cli -cdnssec,delete --dnsseckeyname=domain.com --keytag=20510
```

cdnssec, modify

cdnssec,modify modifies the key data for the key name given in the input file or command line option, with data in key files present in **-keyfilepath** or **Synopdata** from **-inputfile**.

Before you begin

Only one of the following is required:

- **--inputfile** parameter
- **--dnsseckeyname** and **--keytag** parameters

Synopsis

```
qip-cli -cdnssec,modify --keyfilepath=< directory containing key
   files > [-inputfile <filename>|--dnsseckeyname=<keyName> --
   keytag <keyTag>] [--dskey]
```

Parameters

qip-cli -c **cdnssec,modify** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **cdnssec**. Action is **modify**.
Base Options		
--keyfilepath		Key File will be generated in the path specified.
--inputfile	-I	Name of the file containing input for the CLI.
--dnsseckeyname		Retrieves the DNSSEC key for the given keyname.
--keytag		Specifies the key tag generated by the key generation utility for each key. It is the last numeric number following the "+" sign in the output generated or the key file names.
--dskey		Specifies the DS record that needs to be added to the key. The DS record is read from a file named **dsset <zone name>** and should be present in the path specified in **-keyfilepath**.

Command line example

```
qip-cli -cdnssec,modify -o --input=domain.in --keyfilepath=/keys
```

Sample input example

The following is a sample input file for modifying a key in the zone name.

```
<?xml version="1.0" encoding="UTF-8" ?>
<DNSSEC_KEY_REC>
<keyName>lucent.com</keyName>
<keyTag>30388</keyTag>
<algorithm>1</algorithm>
<bitSize>512</bitSize>
<privateKeyFileName>Klucent.com.+001+30388.private
   </privateKeyFileName>
<publicKeyFileName>Klucent.com.+001+30388.key</publicKeyFileName>
<ksk>true</ksk>
<deployDate>2009-05-05 16:40:31.0</deployDate>
<expireDate>2009-12-06 16:40:31.0</expireDate>
</DNSSEC_KEY_REC>
```

cdnssec, get

cdnssec,get allows you to retrieve the DNSSEC key for a given keyname.

Synopsis

```
qip-cli -cdnssec,get  --dnsseckeyname=<keyname> --keytag <keyTag>
   --outputfile=<outputfile> --keyfilepath=<directory where key
   files generated > [--format=XML/CSV]  [--dskey]
```

Parameters

qip-cli -c **cdnssec,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **cdnssec**. Action is **get**.
Base Options		
--keyfilepath		Key File will be generated in the path specified.
--dnsseckeyname		Retrieves the DNSSEC key for the given keyname.
--keytag		Specifies the key tag generated by the key generation utility for each key. It is the last numeric number following the "+" sign in the output generated or the key file names.
--outputfile	-O	Name of the file containing output from the CLI.
--dskey		Specifies the DS record that needs to be added to the key. The DS record is read from a file named **dsset <zone name>** and should be present in the path specified in **-keyfilepath**.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **--format** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **xml** (default) **csv** If you specify **csv**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Command line example

```
qip-cli -cdnssec,get --dnsseckeyname=domain.com --keytag 20511 --
   format=XML
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<object-stream>
<DNSSEC_KEY_REC>
<keyName>lucent.com</keyName>
<keyTag>30388</keyTag>
<algorithm>1</algorithm>
<bitSize>512</bitSize>
<privateKeyFileName>Klucent.com.+001+30388.private
   </privateKeyFileName>
<publicKeyFileName>Klucent.com.+001+30388.key</publicKeyFileName>
<ksk>true</ksk>
<deployDate>2009-05-05 16:40:31.0</deployDate>
<expireDate>2009-12-06 16:40:31.0</expireDate>
</DNSSEC_KEY_REC>
</object-stream>
```

cserver, copy

`cserver,copy` allows you to clone a DNS/DHCP server.

Synopsis

```
qip-cli  -cserver,copy [Login Options] [Job Options] -T<type>
   --host <host name of server being cloned> --domain <domain name
   of server being cloned> --newhost <host name of the new server>
   --newdomain <domain name of the new server> [--includezones
   [--makesecondary]]
```

Parameters

qip-cli -c **`server,copy`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`cserver`**. Action is **`copy`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
-T		Specifies the DNS or DHCP server.
--host		Specifies the host name of the server being cloned.
--domain		Specifies the domain name of the server being cloned.
--newhost		Specifies the new host name.
--newdomain		Specifies the new domain name.
--includezones		Indicates that the zone associations will be cloned. This is applicable for DNS servers only.
--makesecondary		Indicates that the cloned server will be the secondary for all zones. It is not applicable if **`-includezones`** is not specified. This is applicable for DNS servers only.
Job Options		

Long Option	Short Option	Allowable Values
[--when <date H:S:M>|NOW -- jobdescription <text_description_of_ job>		Specifies the date, brief job description, and options for sending email.
--sendemail to=<email1>, <email2>, cc=<email1>, <email2>]		
--when		Specifies when to run this CLI. The allowable values are **<MM/DD/YYYY H:M:S>** and **<DD/MM/YYYY H:M:S>** The CLI runs at the date and time specified. The date format is set based on the user's location.
NOW		CLI runs immediately in the background.
--jobdescription		Specifies a text description of the job. You can only use this option if you have set a value for the **-when** option.
--when		Specifies when to run this CLI. Allowable value is as follows: ***<MM/DD/YYYY H:M:S>*** or ***<DD/MM/YYYY H:M:S>***- CLI runs at the date and time specified. Date format is set based on user's location. **NOW** - CLI runs immediately in background.
--jobdescription		A text description of the job. You can only use this option if you have set a value for the **--when** option.
--sendemail		This option allows you to specify both "To" and "CC" recipients who receive an email when the job either completes normally or terminates prematurely. You can only use this option if you have set a value for the **--when** option. This option has the following suboptions: • **to** a list of comma-separated email addresses that appear in the "To" address of the email • **cc** a list of comma-separated email addresses that appear in the "cc" address of the email.

Command line example

```
qip-cli -cserver copy,add -u admin1 -p password -o VitalQIP --when
NOW --jobdescription "VitalQIP Testing" --sendemail to=
johndoe@alcatel-lucent.com
-T DNS --host Host1 --domain Domain1 --newhost Host2 --newdomain
Domain2
```

dnsacltemplate,add

qip-cli -c **dnsacltemplate,add** allows you to add one or more ACL templates.

Synopsis

```
qip-cli -c dnsacltemplate,add <login options>
   -I <input_DNS_ACL_TEMPLATE_REC>.xml [--batch]
   [--rejectfile <filename.xml|.csv>] [-v] [-h|?]
```

Parameters

qip-cli -c **dnsacltemplate,add** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **dnsacltemplate**. Action is **add**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing ACL template record input for the CLI.
--rejectfile	-r	File containing the records that failed processing. You must specify a file name if you want to see this output. **Note:** A reject file must not already exist with the name you specify. If it does, it is not overwritten with the new information.
--batch		Enables batch processing on multiple records of the same type in an input file.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	

Command line example

```
qip-cli -c dnsacltemplate,add -u admin1 -p password -o VitalQIP
   -I ACLtemplateparms.xml --batch
   --rejectfile acltemplateparmreject.xml
```

You can use output from the **dnsacltemplate,get** command as a basis for input for this command.

Note the following about ACL Template input records (ACL_TEMPLATE_REC).

- An <acl> contains 0 or more entries. Any of the elements under <acl> may be repeated in any order. For example, an <acl> may contain several <ipv4_address> instances. Also, as is the nature of an access control list, the sequence is significant.
- One of the elements under <acl> is <other>. The values for <other> are the special keywords: **ANY**, **NONE**, **LOCALHOST** and **LOCALNETS**.
- To define an exclusion, prepend a "!" before the value. For example:**<ipv4_network>!192.168.100.0/24</ipv4_network>** and **<other>!LOCALHOST</other>**.

dnsacltemplate,copy

qip-cli -c **`dnsacltemplate,copy`** allows you to copy an ACL template.

Synopsis

```
qip-cli -c dnsacltemplate,copy <login options>
   --name <ACLtemplatename> --toname <newACLtemplatename> [-v]
   [-h|?]
```

Parameters

qip-cli -c **`dnsacltemplate,copy`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`dnsacltemplate`**.
		Action is **`copy`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--name	-n	Name of the ACL template to be copied.
--toname		Name of the new ACL template.

Command line example

```
qip-cli -c dnsacltemplate,copy -u admin1 -p password -o VitalQIP
   --name adminboxes1 --toname adminboxes2
```

dnsacltemplate,delete

qip-cli -c **`dnsacltemplate,delete`** allows you to delete an ACL template.

Synopsis

```
qip-cli -c dnsacltemplate,delete <login options>
   --name <ACLtemplatename> [--force] [-v] [-h|?]
```

Parameters

qip-cli -c **`dnsacltemplate,delete`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`dnsacltemplate`**.
		Action is **`delete`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--name	-n	Name of the ACL template you are deleting.
--force		Delete this ACL template even if it is referenced by something else.

Command line example

```
qip-cli -c dnsacltemplate,delete -u admin1 -p password -o VitalQIP
   --name adminboxes1
```

dnsacltemplate,get

qip-cli -c **dnsacltemplate,get** allows you to obtain an XML representation of the contents of an ACL template.

Synopsis

```
qip-cli -c dnsacltemplate,get <login options>
   --name <ACLtemplatename> [-O <output_file>.xml]
   [--outputxsl file1,file2] [-v] [-h|?]
```

Parameters

qip-cli -c **dnsacltemplate,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **dnsacltemplate**. Action is **get**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the format of the XML output file.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--name	-n	Name of the ACL template you are getting.

Command line example

```
qip-cli -c dnsacltemplate,get -u admin1 -p password -o VitalQIP
   -n adminboxes -O template1.xml
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<DNS_ACL_TEMPLATE_REC>
  <acl>
    <ipv4_address>10.10.2.5</ipv4_address>
    <ipv4_network>10.10.3.0/24</ipv4_network>
  </acl>
  <name>Adminboxes</name>
</DNS_ACL_TEMPLATE_REC>
```

dnsacltemplate,modify

qip-cli -c **dnsacltemplate,modify** allows you to modify an ACL template. You can modify an existing ACL template, or rename a modified ACL template.

Synopsis

```
qip-cli -c dnsacltemplate,modify <login options>
   -I <DNS_ACL_TEMPLATE_REC>.xml
   [--replace current_acltemplatename] [-v] [-h|?]
```

Parameters

qip-cli -c **dnsacltemplate,modify** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **dnsacltemplate**. Action is **modify**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing input for the CLI.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--replace		Current name of the ACL template. **Note:** The --replace option is only necessary when a new name is specified by the input file.

Command line example

```
qip-cli -c dnsacltemplate,modify -u admin1 -p password -o VitalQIP
   -I acltemplatemods1.xml --replace currentacltemplatename
```

You can use the output from a **dnsacltemplate,get** CLI as a template for the input file for this CLI.

dnsacltemplate,search

dnsacltemplate,search allows you to search for one or more ACL templates. The ACL template is output to a filename you specify.

Synopsis

```
qip-cli -c dnsacltemplate,search <login options>
   [--name <acltemplatename>] [-O <output_file>.xml]
   [--outputxsl file1,file2] [-v] [-h|?]
```

Parameters

qip-cli -c **dnsacltemplate,search** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **dnsacltemplate**. Action is **search**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--rule	-R	Name of the rule for which you are searching.

Command line example

```
qip-cli -c dnsacltemplate,search -u admin1 -p password -o VitalQIP
   --name Admin*
```

Sample output file

```
<?xml version="1.0" encoding="UTF-8" ?>
<object-stream>
<DNS_ACL_TEMPLATE_REC>
  <acl>
    <ipv4_address>10.10.2.5</ipv4_address>
    <ipv4_network>10.10.3.0/24</ipv4_network>
  </acl>
  <name>Adminboxes</name>
</DNS_ACL_TEMPLATE_REC>
<DNS_ACL_TEMPLATE_REC>
  <acl>
    <ipv4_address>!10.10.20.4</ipv4_address>
  </acl>
  <name>Adminboxes1</name>
</DNS_ACL_TEMPLATE_REC>
</object-stream>
```

dnsview,add

dnsview,add allows you to add a DNS View to the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c dnsview,add <login options> -I <DNS_VIEW_REC>.xml
   [--batch] [--rejectfile <filename>.xml [-v] [-h|?]
```

Parameters

qip-cli -c **dnsview,add** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **dnsview**. Action is **add**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing DNS view records for the CLI.
--batch		Enables batch processing on multiple records of the same type in an input file.
--rejectfile	-r	File containing the records that failed processing. The file contains the error message details, and a list of any records not added.You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	

Command line example

```
qip-cli -c dnsview,add -u admin1 -p password -o VitalQIP
   -I dnsviewrecs.xml --batch
```

You can use output from the **dnsview,get** or the **dnsview,search** command as a model for an input file for this command.

dnsview,delete

`dnsview,delete` allows you to delete a DNS view.

Note: The delete fails if the view contains any view-specific or shared zones.

Synopsis

```
qip-cli -c dnsview,delete <login options> --view <DNS_view_name>
   [-v] [-h|?]
```

Parameters

qip-cli -c **`dnsview,delete`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`dnsview`**.
		Action is **`delete`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--view	-V	Name of the DNS view being deleted.

Command line example

```
qip-cli -c dnsview,delete -u admin1 -p password -o VitalQIP
   --view hrdnsview2
```

dnsview,get

dnsview,get allows you to obtain an XML representation of the contents of a DNS view.

Synopsis

```
qip-cli -c dnsview,get <login options> -V <viewname>
   [-O <output_file>.xml]  [--outputxsl file1,file2] [-v] [-h|?]
```

Parameters

qip-cli -c **dnsview,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **dnsview**.
		Action is **get**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--view	-V	Name of the DNS view.

Command line input sample

```
qip-cli -c dnsview,get --user admin1 --password password
   -o VitalQIP --view HumanRes
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<DNS_VIEW_REC>
  <createDbCache>false</createDbCache>
  <dbCacheFileExtension></dbCacheFileExtension>
  <dnsOptionList>
    <option>
      <name>acache-cleaning-interval</name>
      <value>60;</value>
    </option>
    <option>
      <name>allow-query</name>
      <addrMatchValue>
        <template>ACL1</template>
        <ipv4_address>!192.168.1.0</ipv4_address>
      </addrMatchValue>
    </option>
    <option>
      <name>notify</name>
      <value>no;</value>
    </option>
  </dnsOptionList>
  <matchClients>
    <template>ACL1</template>
  </matchClients>
  <matchDestinations>
    <template>ACL1</template>
    <ipv4_address>192.168.1.1</ipv4_address>
  </matchDestinations>
  <matchRecursiveOnly>false</matchRecursiveOnly>
  <name>V1</name>
</DNS_VIEW_REC>
```

The output from this CLI, may contain the element `<dnsOptionList>`, which contains a list of DNS options.:

Each DNS option may take an argument. It may either be a single string, or an address match list. For the case of a single string, put it in the `<value>` element. For the case of an address match list, use the `<addrMatchValue>` element. An address match list uses the following syntax:

- A <dnsview> contains 0 or more entries. Any of the elements under <acl> may be repeated in any order. For example, an <acl> may contain several <ipv4_address> instances. Also, as is the nature of an access control list, the sequence is significant.
- One of the elements under <acl> is <other>. The values for <other> are the special keywords: **ANY**, **NONE**, **LOCALHOST** and **LOCALNETS**.

- To define an exclusion, prepend a "!" before the value. For example:**`<ipv4_network>!192.168.100.0/24</ipv4_network>`** and **`<other>!LOCALHOST</other>`**.

The names of all supported options are defined in *$QIPHOME/web/xml/dns_zone_options.xml* or can be obtained by running "info,dnsoptions" (p. 5-114)

dnsview,list

dnsview,list generates a list of all DNS Views within an organization. The output from this list is used by the **dnsview,sequence** CLI.

Synopsis

```
qip-cli -c dnsview,list <login options>
   [-O <output_file>.xml|csv]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header][,textqualifier=<char>][,utf8]])
   [--outputxsl file1,file2] [-v] [-h|?]
```

Parameters

qip-cli -c **dnsview,list** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **dnsview**. Action is **list**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.

Command line input sample

```
qip-cli -c dnsview,list -u admin1 -p admin1
   -o "VitalQIP Organization" -I dnsviewlist.xml
```

Sample Output File

```
<LIST_REC>
  <names>View1</names>
  <names>HumanRes</names>
</LIST_REC>
```

dnsview,modify

`dnsview,modify` allows you to modify a DNS view in the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c dnsview,modify <login options>
   [--replace <currentname>] -I <DNS_VIEW_REC>.xml [-v] [-h|?]
```

Parameters

qip-cli -c **`dnsview,modify`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`dnsview`**. Action is **`modify`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing DNS view records as input for the CLI.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--replace		DNS view's current name. **Note:** The --replace option is only necessary when a new name is specified by the input file.

Command line input sample

```
qip-cli -c dnsview,modify -u admin1 -p admin1 -o Documentation
   -I dnsmods.xml
```

You can use the output file from the **`dnsview,get`** cli as the template for the input file for this command.

dnsview,optionreport

dnsview,optionreport allows you to view the options associated with a DNS view.

Synopsis

```
qip-cli -c dnsview,optionreport <login options>
   [-O <output_file>.text|xml|.csv] [--outputxsl file1,file2]
   ([--format text|xml|csv[,columndelimiter=<char>]
   [,comment=<char>][,recorddelimiter=<char>]
   [,escapemode=doubled|backslash][,header]
   [,textqualifier=<char>][,utf8]]) -V Viewname [-v] [-h|?]
```

Parameters

qip-cli -c **dnsview,optionreport** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **dnsview**. Action is **optionreport**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.

Long Option	Short Option	Allowable Values
--format		If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`text`** (default) **`csv`** xml If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--view	-V	Name of the DNS view for which you want to see options.

Text

Command line input sample

```
qip-cli -c dnsview,optionreport -u admin1 -p password -o VitalQIP
   -V HumanRes
```

Sample Output File

```
Option Name            Effective Value  View?
=====================  ===============  =====
additional-from-cache  yes;             Y
allow-notify           {10.10.2.4;};    Y
```

CSV

Command line input sample

```
qip-cli -c dnsview,optionreport -u admin1 -p password -o VitalQIP
   -V HumanRes --format csv
```

Sample Output File

```
additional-from-cache,yes;,yes;
allow-notify,{10.10.2.4;};,{10.10.2.4;};
```

XML

Command line input sample

```
qip-cli -c dnsview,optionreport -u admin1 -p password -o VitalQIP
   -V HumanRes --format xml
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<DnsViewOptionReport>
  <option>
    <effectiveValue>yes;</effectiveValue>
    <name>additional-from-cache</name>
    <viewValue>yes;</viewValue>
  </option>
  <option>
    <effectiveValue>{10.10.2.4;};</effectiveValue>
    <name>allow-notify</name>
    <viewValue>{10.10.2.4;};</viewValue>
  </option>
</DnsViewOptionReport>
```

dnsview,search

dnsview,search allows you to search for DNS views. DNS Views are returned in alphabetical order.

Synopsis

```
qip-cli -c dnsview,search <login options> [-O <output_file>.xml]
   [--outputxsl file1,file2] [--view <viewname>]
   [--uda [[/object/]group/]name[=value]]) [-v] [-h|?]
```

Parameters

qip-cli -c **dnsview,search** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **dnsview**. Action is **search**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--view	-V	Name of the DNS view for which you are searching. You can use an asterisk (*) as a wildcard character to return a range of results. If you use the asterisk as the first or only character, you must put this value in either single or double quotes. If you do not specify a view name or pattern, all views are displayed.

Long Option	Short Option	Allowable Values
--uda	-U	**`--uda`** [[***`/component`***/]***`group`***/]***`name`***[=***`value`***

The following explains how UDAs are handled in the Web GUI compared to how they are handled when using them in a CLI.

Component

Optional. The component you specify must be defined under the infrastructure element for which this CLI is searching (for example, node, pool, or block).

Group

Optional. The attribute group you specify must be defined within the infrastructure or the search fails.

- When the GUI indicates **<NONE>** for a group, omitting a value for a group in the CLI means "not in a group". Example: **`--uda /node//state=NJ`**, or **`--uda state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute is not in a group. The first example specifies that the attribute is associated with a node. The second example does not specify what component the UDA is associated with.
- When the GUI specifies **<ANY>**, using an asterisk (**`*`**) to specify the group in the CLI means that the attribute must be in a group, regardless of the group's name. Example: **`--uda */state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute must belong to a group, but the group is not specified.
- When the GUI specifies **<IGNORE>**, for a group, using a dash (**`-`**) in the CLI means it does not matter if the UDA is in a group, or not. Example: **`--uda -/state=NJ`**. In this case, the attribute is "state" and the value is NJ. The attribute will be found regardless of whether it is in a group.

Name

This argument contains the name of the attribute, followed by an equals sign (=), followed by the value for which you are searching.

Command line input sample

```
qip-cli -c dnsview,search -u admin1 -p password -o VitalQIP
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<object-stream>
<DNS_VIEW_REC>
  <createDbCache>false</createDbCache>
  <dbCacheFileExtension></dbCacheFileExtension>
  <matchRecursiveOnly>false</matchRecursiveOnly>
  <name>HumanRes</name>
</DNS_VIEW_REC>
<DNS_VIEW_REC>
  <createDbCache>false</createDbCache>
  <dbCacheFileExtension></dbCacheFileExtension>
  <dnsOptionList>
    <option>
      <name>acache-cleaning-interval</name>
      <value>2;</value>
    </option>
    <option>
      <name>acache-enable</name>
      <value>yes;</value>
    </option>
    <option>
      <name>additional-from-cache</name>
      <value>no;</value>
    </option>
  </dnsOptionList>
  <matchRecursiveOnly>false</matchRecursiveOnly>
  <name>View1</name>
</DNS_VIEW_REC>
</object-stream>
```

The output from this CLI, may contain the element `<dnsOptionList>`, which contains a list of DNS options.:

Each DNS option may take an argument. It may either be a single string, or an address match list. For the case of a single string, put it in the `<value>` element. For the case of an address match list, use the `<addrMatchValue>` element.

The names of all supported options are defined in *$QIPHOME/web/xml/dns_zone_options.xml* or can be obtained by running "info,dnsoptions" (p. 5-114)

dnsview,sequence

dnsview,sequence allows you to change the order of DNS Views in the VitalQIP® (QIP) database. Modify the order of the DNS Views listed in the input file, and then run this CLI to change the order.

Synopsis

```
qip-cli -c dnsview,sequence <login options>
   -I <inputfile>.xml|.csv
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header][,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **dnsview,sequence** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **dnsview**. Action is **sequence**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--inputfile	-I	Name of the file containing DNS view records for the CLI. **Note:** Generate the input file for this CLI using the **`dnsview,list`** command.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.

Command line example

```
qip-cli -c dnsview,sequence -u admin1 -p password -o VitalQIP
   -I dnsviewseqs.xml
```

You can use output from the **`dnsview,list`** command as a model for an input file for this command.

info,dnsoptions

`info,dnsoptions` displays a list of the DNS options available for use with **`dnsview`**, **`zone`**, and **`zoneserver`** CLIs. For details on these options, see Chapter 13 of the *Web Client User's Guide*, under *Zone option management in DNS Views*.

Synopsis

```
qip-cli -c info,dnsoptions <login options> [--outputfile=<file>]
   [-v] [-h|?]
```

Parameters

qip-cli -c **`info,dnsoptions`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`info`**. Action is **`dnsoptions`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--outputfile	-O	Name of the file containing output from the CLI.

Command line example

```
qip-cli -c info,dnsoptions -u admin1 -p password -o VitalQIP
```

info,get

`info,get` allows you to obtain a template for various VitalQIP® (QIP) elements in either XML or CSV format. You can then edit this template to create an input file for another CLI.

Synopsis

```
qip-cli -c info,get --n <Schema_Element>
   --format xml|csv[,header[,columndelimiter=<char>]]
   [--outputfile=<file>] [-v] [-h|?]
```

Parameters

qip-cli -c **`info,get`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`info`**. Action is **`get`**.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		

Long Option	Short Option	Allowable Values
	-n	Type of schema for which you want a template. Allowable values are as follows: • ACL_TEMPLATE_REC • ADDR_TEMPL_REC • ADDRESS_BLOCK_REC • ADDRESS_RANGE_REC • AllocateBlockToPoolRequest • CHILD_POOL_REC • DNS_ACL_TEMPLATE_REC • DNS_SHARED_ZONE_REC • DNS_VIEW_REC • DNS_VIEW_ZONE_REC • DNS_ZONE_SERVER_REC • ExplicitBlockAllocRequest • IPV4_OBJECT_ADDR_REC • MAINTAINER_REC • MANAGED_FILE_REC • NODE_REC • POOL_KEYS • QuickBlockAllocRequest • RecoverBlockRequest • REGISTRY_REC • REV_ZONE_TEMPL_REC • RULE_KEYS • RULE_REC • SCHEDULE_JOB_REC • SEED_BLOCK_REC • SEED_POOL_REC • SUBNET_PROF_TEMPL_REC • V4_SUBNET_REC • V6_SUBNET_REC • V6_SEED_BLOCK_REC • ZONE_REC

Long Option	Short Option	Allowable Values
--format		Type of output in which the template is formatted. Allowable values are: **`xml`** (default) This provides an XML template file as a starting point to author a new XML input file. You must provide values for the XML elements, and if appropriate, remove unwanted XML elements. **`csv`** If you specify csv, you can specify a header and a column delimiter. If you specify a header, each field of the csv output is labeled. The default column delimiter is a comma. Allowable value for the column delimiter is any keyboard character. If you do not use the "header" subption, **`--format=csv`** does not provide an initial template file. Instead, it outputs the assignment of each column in the csv file. Alternatively, this same information can be organized into a format understood by spreadsheet programs, such as Microsoft Excel, by using the "header" suboption: **`--format=csv,header`**. Excel users may find it useful to configure the cell format of the first row to allow wrapping text. When using "header", be careful that you do not later unintentionally input the header row as program data. Depending on the application, you may need to either delete the header row, or configure the application to skip the first header row.
--outputfile	-O	Name of the file containing output from the CLI.

Command line example

```
qip-cli -c info,get -u admin1 -p password -o VitalQIP
   -n MAINTAINER_REC
```

XML output

```
<?xml version="1.0" encoding="UTF-8" ?>
<MAINTAINER_REC>
    <registry></registry>
    <name></name>
    <emailAddress></emailAddress>
    <encryptionType></encryptionType>
    <password></password>
    <prefix></prefix>
    <description></description>
</MAINTAINER_REC>
```

Command line example

```
qip-cli -c info,get -u admin1 -p password -o VitalQIP
   -n MAINTAINER_REC --format csv
```

CSV output

```
# Mapping from XML element name to CSV column number.
MAINTAINER_REC.registry=1
MAINTAINER_REC.name=2
MAINTAINER_REC.emailAddress=3
MAINTAINER_REC.encryptionType=4
MAINTAINER_REC.password=5
MAINTAINER_REC.prefix=6
MAINTAINER_REC.description=7
```

Note: Not all elements support CSV output. If you request CSV output for an element that does not support it, you receive the following message:

```
Error: File does not exist:
   $CLASSPATH/csvMaps/<filetype>.properties
```

job,cancel

`job,cancel` allows you to stop a scheduled job from running. Use this CLI on jobs that have not started to run. This CLI works with a job submitted using either the VitalQIP® (QIP) Web GUI, or a CLI.

Synopsis

```
qip-cli -c job,cancel <login options> --jobid <job_ID> [-v] [-h|?]
```

Parameters

qip-cli -c **`job,cancel`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`job`**. Action is **`cancel`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--jobid		Identification number of the job you want to retrieve. This is the number provided by VitalQIP® (QIP) at the time the job is submitted.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	

Command line example

```
qip-cli -c job,cancel -u admin1 -p password -o VitalQIP
   --jobid 345
```

job,delete

job,delete allows you to delete a job with any status from the database. This CLI works with a job submitted using either the VitalQIP® (QIP) Web GUI, or a CLI.

Synopsis

qip-cli -c job,delete *<login options>* --jobid <job_ID> [-v] [-h|?]

Parameters

qip-cli -c **job,delete** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **job**. Action is **delete**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--jobid		Identification number of the job you want to retrieve. This is the number provided by VitalQIP® (QIP) at the time the job is submitted.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	

Command line example

```
qip-cli -c job,delete -u admin1 -p password -o VitalQIP
   --jobid 345
```

job,get

job,get allows you to obtain an XML or CSV representation of the status of a job submitted using either the VitalQIP® (QIP) Web GUI, or a CLI.

Synopsis

```
qip-cli -c job,get <login options> --jobid <job_ID>
   [-O <output_file>.xml|.csv]  [--outputxsl file1,file2]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **job,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **job**. Action is **get**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is text, xml or csv. Allowable values are: **`text`** (default) **`xml`** **`csv`** If you select text as your output format, the results of this CLI appear on the screen. The number of lines that print at once are controlled by **`cli.prettyPrintNumRows`** in the *cli.properties* file. If no output file is specified, and no **`--format`** option is specified, text is the default. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`text`** (default) **`xml`** **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--jobid		Identification number of the job you want to retrieve. This is the number provided by VitalQIP® (QIP) at the time the job is submitted.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.

Command line example

```
qip-cli -c job,get -u admin1 -p password -o VitalQIP --jobid 1
   -o job.get
```

Output example

```
Job ID Admin Name Name Description Status Server ScheduledTime  StartTime
   CompletedTime
====== ========== ==== =========== ====== ====== =============  =========
   =============
 1      admin1     RENUM_BLK renumber block SCHEDULED   Sat Jun 09 13:47:00
   EDT 2007
```

job,search

job,search allows you to obtain an XML or CSV representation of jobs submitted using either the VitalQIP® (QIP) Web GUI, or a CLI.

Synopsis

```
qip-cli -c job,search <login options> [--admin <admin name>]
   [--status Running|Pending|Delayed|Scheduled|Completed|
   Failed|Cancelled] [--jobname <jobname>]
   [--scheduled begin=<MM/DD/YYYY H:M:S>,end=<MM/DD/YYYY H:M:S>]
   [--completed begin=<MM/DD/YYYY H:M:S>,end=<MM/DD/YYYY H:M:S>]
   [-O <output_file>.xml|.csv]  [--outputxsl file1,file2]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **job,search** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **job**. Action is **search**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is text, xml or csv. Allowable values are: **`text`** (default) **`xml`** **`csv`** If you select text as your output format, the results of this CLI appear on the screen. The number of lines that print at once are controlled by **`cli.prettyPrintNumRows`** in the *cli.properties* file. If no output file is specified, and no **`--format`** option is specified, text is the default. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`text`** (default) **`xml`** **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--admin		User ID of the administrator who submitted the jobs

Long Option	Short Option	Allowable Values
--status		Status of the jobs: Allowable values are as follows: • **Running** - The job is currently executing. • **Pending** - The job is in the process of being run. • **Delayed** - The job was scheduled to run at this time, but cannot because the maximum number of concurrent jobs would be exceeded. • **Scheduled** - The job has been scheduled to run. • **Completed** -The job has completed successfully. • **Failed** - The job has run, but has failed execution. • **Cancelled** - The job was cancelled.
--jobname		Name of the job for which you are searching.
--scheduled		The range of time in which the jobs began. This field contains two times/dates separated by a comma. The first time you enter is the earliest that a job was scheduled to begin, the second date is the last that a job was scheduled to begin.
--completed		The range of time in which the jobs completed. This field contains two times/dates separated by a comma. The first time you enter is the earliest that a job completed, the second date is the latest that a job completed..
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	

Command line example

```
qip-cli -c job,search -u admin1 -p password -o VitalQIP
   --admin jsmith --status Pending -O joboutput.xml
```

Text output example

```
Job IDAdmin NameNameDescriptionStatusServerScheduled TimeStart Time
   CompletedTime
Job ID Admin Name Name                Description
   Status    Server   Scheduled Time              Start Time
   Completed Time
==============================================================================
   =========
```

```
 171    admin1     REPORTS           In Background:Run Report
  COMPLETED qdev3vm1 Fri Jun 08 10:06:04 EDT 2007 Fri Jun 08 10:06:12 EDT
  2007 Fri Jun 08 10:06:13 EDT 2007
 169    admin1     REPORTS           In Background:Run Report
  COMPLETED qdev3vm1 Thu Jun 07 20:39:21 EDT 2007 Thu Jun 07 20:39:24 EDT
  2007 Thu Jun 07 20:39:27 EDT 2007
```

CSV output example

```
171,admin1,REPORTS,In Background:Run Report,COMPLETED,qdev3vm1,Fri Jun 08
  10:06:04 EDT 2007,Fri Jun 08 10:06:12 EDT 2007,Fri Jun 08 10:06:13 EDT 2007
169,admin1,REPORTS,In Background:Run Report,COMPLETED,qdev3vm1,Thu Jun 07
  20:39:21 EDT 2007,Thu Jun 07 20:39:24 EDT 2007,Thu Jun 07 20:39:27 EDT 2007
```

XML output example

```
<?xml version="1.0" encoding="UTF-8" ?>
<object-stream>
<SCHEDULE_JOB_REC>
  <completeTime>Fri Jun 08 10:06:13 EDT 2007</completeTime>
  <description>In Background:Run Report</description>
  <id>171</id>
  <name>REPORTS</name>
  <scheduledTime>Fri Jun 08 10:06:04 EDT 2007</scheduledTime>
  <server>qdev3vm1</server>
  <startTime>Fri Jun 08 10:06:12 EDT 2007</startTime>
  <status>COMPLETED</status>
  <userName>admin1</userName>
</SCHEDULE_JOB_REC>
<SCHEDULE_JOB_REC>
  <completeTime>Thu Jun 07 20:39:27 EDT 2007</completeTime>
  <description>In Background:Run Report</description>
  <id>169</id>
  <name>REPORTS</name>
  <scheduledTime>Thu Jun 07 20:39:21 EDT 2007</scheduledTime>
  <server>qdev3vm1</server>
  <startTime>Thu Jun 07 20:39:24 EDT 2007</startTime>
  <status>COMPLETED</status>
  <userName>admin1</userName>
</SCHEDULE_JOB_REC>
</object-stream>
```

list,search

`list,search` allows you to search for items defined in VitalQIP® (QIP) and sometimes needed when using address allocation and address management.

Synopsis

```
qip-cli -c list,search <login options> -T <type>
   [-O <outputfile>.xml|.csv] [--outputxsl <file1,file2,...>]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`list,search`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`list`**.
		Action is **`search`**.
Base Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		

Long Option	Short Option	Allowable Values
--type	-T	Type of list being searched. Allowable values are as follows: • **`dhcp_srv`** - DHCP servers • **`dhcp_opt_templ`** - DHCP option templates • **`dhcp_scope_pol_templ`** - DHCP policy templates • **`dns_srv`** - DNS servers • **`domain`** - Domains • **`obj_class`** - Object classes • **`subnet_organization`** - Subnet organizations • **`user_class`** - User classes • **`vendor_class`** - Vendor classes
--outputfile	-O	Name of the file that contains the search results.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Command line example

```
qip-cli -c list,search -u admin1 -p password -o VitalQIP
   -T dns_srv
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<LIST_REC>
  <names>SERVER1</names>
</LIST_REC>
```

maintainer,add

`maintainer,add` allows you to add a maintainer to the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c maintainer,add <login options> -I <inputfile>.xml|.csv
   [--batch] [--rejectfile <filename>.xml|.csv]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header][,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`maintainer,add`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`maintainer`**. Action is **`add`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing maintainer records for the CLI.
--batch		Enables batch processing on multiple records of the same type in an input file.
--rejectfile	-r	File containing the records that failed processing. The file contains the error message details, and a list of any records not added.You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.

Command line example

```
qip-cli -c maintainer,add -u admin1 -p password -o VitalQIP
   -I maintainerrecs.xml --batch
```

You can use output from the **`maintainer,get`** or the **`maintainer,search`** command as a model for an input file for this command.

maintainer,delete

`maintainer,delete` allows you to delete a maintainer.

Synopsis

```
qip-cli -c maintainer,delete <login options>
   --name <maintainername> -T afrinic|arin|apnic|ripe [-v] [-h|?]
```

Parameters

qip-cli -c **`maintainer,delete`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`maintainer`**. Action is **`delete`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--name	-n	Name of the maintainer being deleted.
--type	-T	Type of registry to which the maintainer is assigned. Allowable values are as follows: • afrinic • arin • apnic • ripe

Command line example

```
qip-cli -c maintainer,delete -u admin1 -p password -o VitalQIP
   --name hrmaintainer2 -T arin
```

maintainer,get

maintainer,get allows you to obtain an XML or CSV representation of the contents of a maintainer.

Synopsis

```
qip-cli -c maintainer,get <login options>
   [-O <output_file>.xml|.csv]  [--outputxsl file1,file2]
   --name <maintainername> --type <AFRINIC|ARIN|APNIC|RIPE>
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header] [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **maintainer,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **maintainer**. Action is **get**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--name	-N	Name of the maintainer.
--type	-T	Type of registry to which the maintainer is assigned. Allowable values are as follows: • afrinic • arin • apnic • ripe

Command line input sample

XML

```
qip-cli -c maintainer,get --user admin1 --password password
   -o VitalQIP --name jim --type arin
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<MAINTAINER_REC>
  <emailAddress>jim@abcd.com</emailAddress>
  <name>jim</name>
  <registry>ARIN</registry>
</MAINTAINER_REC>
```

CSV

qip-cli -c maintainer,get --user admin1 --password password -o VitalQIP --name jim --type arin --format csv

Sample Output File

```
ARIN,jim,jim@abcd.com
```

maintainer,modify

`maintainer,modify` allows you to modify a maintainer in the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c maintainer,modify <login options>
   [--replace <currentname>] -I <inputfile>.xml|.csv
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`maintainer,modify`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`maintainer`**. Action is **`modify`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing maintainer records as input for the CLI.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--replace		Maintainer's current name. **Note:** The --replace option is only necessary when a new name is specified by the input file.

Command line input sample

```
qip-cli -c maintainer,modify -u admin1 -p password -o VitalQIP
   -I maintainermods.xml
```

You can use an output file from the **`maintainer,get`** CLI as a template for the input file for this CLI.

maintainer,search

`maintainer,search` allows you to search for maintainers within a specified Internet registry.

Synopsis

```
qip-cli -c maintainer,search <login options>
   --type afrinic|arin|apnic|ripe [-O <output_file>.xml|.csv]
   [--outputxsl file1,file2]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`maintainer,search`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`maintainer`**. Action is **`search`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--type	-T	Type of registry to which the maintainer is assigned. Allowable values are as follows: • afrinic • arin • apnic • ripe

Command line input sample

```
qip-cli -c maintainer,search -u admin1 -p password -o VitalQIP
   -T ARIN
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<object-stream>
<MAINTAINER_REC>
  <description>DESCRIPTION</description>
  <emailAddress>dave@abcd.com; edna@lucent.com;</emailAddress>
  <name> another new one</name>
  <prefix>prefix</prefix>
  <registry>ARIN</registry>
</MAINTAINER_REC>
<MAINTAINER_REC>
  <emailAddress>daryl@abcd.com</emailAddress>
  <name>daryl</name>
  <prefix>daryl</prefix>
  <registry>ARIN</registry>
</MAINTAINER_REC>
<MAINTAINER_REC>
  <emailAddress>peg@abcd.com</emailAddress>
  <name>arin maint 01</name>
  <registry>ARIN</registry>
</MAINTAINER_REC>
</object-stream>
```

managedfile,add

managedfile,add allows you to add a managed file to the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c managedfile,add <login options>
   --filepath <location_of_file> (-I <inputfile>.xml
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er][,textqualifier=<char>][,utf8]]))
   [--rejectfile <filename>.xml [-v] [-h|?]
```

or

```
qip-cli -c managedfile,add <login options>
   --filepath <location_of_file> (--name <managed_file_name>
   [--uda =[group/]name[=value]]) [--rejectfile <filename>.xml
   [-v] [-h|?]
```

The first synopsis allows you to add a managed file and associate it with a server. The input file contains records for the filename, attributes, and servers. The second synopsis allows you to add a managed file, but if you want to associate it with one or more servers, you must then use the **managedfile,associate** CLI.

Parameters

qip-cli -c **managedfile,add** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **managedfile,**. Action is **add**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing managed file records for the CLI.
--batch		Enables batch processing on multiple records of the same type in an input file.

Long Option	Short Option	Allowable Values
--rejectfile	-r	File containing the records that failed processing. The file contains the error message details, and a list of any records not added.You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Long Option	Short Option	Allowable Values
--uda	-U	**`--uda`** [[***`/component`***/]***`group`***/]***`name`***[=***`value`*** The following explains how UDAs are handled in the Web GUI compared to how they are handled when using them in a CLI. **Component** ***Optional.*** The component you specify must be defined under the infrastructure element for which this CLI is searching (for example, node, pool, or block). **Group** ***Optional***. The attribute group you specify must be defined within the infrastructure or the search fails. • When the GUI indicates **<NONE>** for a group, omitting a value for a group in the CLI means "not in a group". Example: **`--uda /node//state=NJ`**, or **`--uda state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute is not in a group. The first example specifies that the attribute is associated with a node. The second example does not specify what component the UDA is associated with. • When the GUI specifies **<ANY>**, using an asterisk (**`*`**) to specify the group in the CLI means that the attribute must be in a group, regardless of the group's name. Example: **`--uda */state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute must belong to a group, but the group is not specified. • When the GUI specifies **<IGNORE>**, for a group, using a dash (**`-`**) in the CLI means it does not matter if the UDA is in a group, or not. Example: **`--uda -/state=NJ`**. In this case, the attribute is "state" and the value is NJ. The attribute will be found regardless of whether it is in a group. **Name** This argument contains the name of the attribute, followed by an equals sign (=), followed by the value for which you are searching.
--version	-v	Displays the version of this CLI.

Long Option	Short Option	Allowable Values
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--name	-n	Name of the managed file being added.
--filepath		Full path to the managed file being added.

Command line example

Input file

```
qip-cli -c managedfile,add -u admin1 -p password -o VitalQIP
   --filepath=/tmp/yourData.txt  -I rec.xml
```

You can use output from the **managedfile,get** or the **managedfile,search** command as a model for an input file for this command.

Directly add file

```
qip-cli -c managedfile,add -u admin1 -p password -o VitalQIP
   --filepath=/tmp/yourData.txt  -n myFile -U state=NJ
```

If you want to associate the file you added with this syntax, you must run the **managedfile,associate** CLI.

managedfile,associate

managedfile,associate allows you to associate a managed file to either a DNS or DHCP server.

Synopsis

```
qip-cli -c managedfile,associate <login options>
   -n <managedfile_name> --server <servername> --type dns|dhcp
   [-v] [-h|?]
```

Parameters

qip-cli -c **managedfile,associate** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **managedfile**. Action is **associate**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--name	-n	Name of the file being deleted.
--server	-S	Name of the server to which the file is being associated.
--type	-T	Specifies the type of server to which the file is being associated. Allowable values are: • **dns** • **dhcp**

Command line example

```
qip-cli -c managedfile,associate -u admin1 -p password -o VitalQIP
   --name admin2.sh --type dns --server eastcoast1
```

managedfile,delete

managedfile,delete allows you to delete a managed file contained within a DNS view.

Synopsis

```
qip-cli -c managedfile,delete <login options>
   -n <managedfile_name> [--force] [-v] [-h|?]
```

Parameters

qip-cli -c **managedfile,delete** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **managedfile**.
		Action is **delete**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--name	-n	Name of the file being deleted.
--force		Delete the file even if it is being referenced.

Command line example

```
qip-cli -c managedfile,delete -u admin1 -p password -o VitalQIP
   --name admin2.sh
```

managedfile,get

managedfile,get allows you to obtain an XML or CSV list of managed files.

Synopsis

```
qip-cli -c managedfile,get <login options> -n <managedfilename>
   -V <viewname> [--format xml|csv[,columndelimiter=<char>]
   [,comment=<char>][,recorddelimiter=<char>]
   [,escapemode=doubled|backslash][,header]
   [,textqualifier=<char>][,utf8]])[-O <output_file>.xml]
   [--outputxsl file1,file2] [-v] [-h|?]
```

Parameters

qip-cli -c **managedfile,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **managedfile**. Action is **get**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--zone	-Z	Name of the zone being obtained.
--view	-V	Name of the view that contains the zone.

Command line input sample

```
qip-cli -c managedfile,get -u admin1 -p passwd -o VitalQIP
   -n admin2.sh
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<MANAGED_FILE_REC>
  <name>admin2.sh</name>
</MANAGED_FILE_REC>
```

managedfile,modify

managedfile,modify allows you to modify a managed file in the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c managedfile,modify <login options>
   [--replace <currentname>] [--filepath <location_of_file>]
   (-n <name> [--uda=[group/]name[=value]]) -I <inputfile>.xml
   [-v] [-h|?]
```

Parameters

qip-cli -c **managedfile,modify** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **managedfile**. Action is **modify**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing managed file records as input for the CLI.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		

Long Option	Short Option	Allowable Values
--uda	-U	**`--uda`** [[***`/component`***/]***`group`***/]***`name`***[=***`value`*** The following explains how UDAs are handled in the Web GUI compared to how they are handled when using them in a CLI. **Component** ***Optional.*** The component you specify must be defined under the infrastructure element for which this CLI is searching (for example, node, pool, or block). **Group** ***Optional***. The attribute group you specify must be defined within the infrastructure or the search fails. • When the GUI indicates **<NONE>** for a group, omitting a value for a group in the CLI means "not in a group". Example: **`--uda /node//state=NJ`**, or **`--uda state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute is not in a group. The first example specifies that the attribute is associated with a node. The second example does not specify what component the UDA is associated with. • When the GUI specifies **<ANY>**, using an asterisk (**`*`**) to specify the group in the CLI means that the attribute must be in a group, regardless of the group's name. Example: **`--uda */state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute must belong to a group, but the group is not specified. • When the GUI specifies **<IGNORE>**, for a group, using a dash (**`-`**) in the CLI means it does not matter if the UDA is in a group, or not. Example: **`--uda -/state=NJ`**. In this case, the attribute is "state" and the value is NJ. The attribute will be found regardless of whether it is in a group. **Name** This argument contains the name of the attribute, followed by an equals sign (=), followed by the value for which you are searching.

Long Option	Short Option	Allowable Values
--replace		Managed file's current name. **Note:** The --replace option is only necessary when a new name is specified by the input file.
--name	-n	Name of the managed file being modified.

Command line input sample

```
qip-cli -c managedfile,modify -u admin1 -p admin1
   -o Documentation --name admin2.sh -I managedfilemods.xml
```

You can use the output file from the **managedfile,get** cli as the template for the input file for this command.

managedfile,search

managedfile,search allows you to search for managed files and create a list of their names.

Synopsis

```
qip-cli -c managedfile,search <login options>
   --name <managedfilename> (--server <servername>
   --type dns|dhcp|nis) [-O <output_file>.xml|csv]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er][,textqualifier=<char>][,utf8]])[--outputxsl file1,file2]
   [--uda=[group/]name[=value]] [-v] [-h|?]
```

Parameters

qip-cli -c **managedfile,search** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **managedfile**. Action is **search**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Long Option	Short Option	Allowable Values
--uda	-U	**`--uda`** [[***`/component`***/]***`group`***/]***`name`***[=***`value`*** The following explains how UDAs are handled in the Web GUI compared to how they are handled when using them in a CLI. **Component** ***Optional.*** The component you specify must be defined under the infrastructure element for which this CLI is searching (for example, node, pool, or block). **Group** ***Optional***. The attribute group you specify must be defined within the infrastructure or the search fails. • When the GUI indicates **<NONE>** for a group, omitting a value for a group in the CLI means "not in a group". Example: **`--uda /node//state=NJ`**, or **`--uda state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute is not in a group. The first example specifies that the attribute is associated with a node. The second example does not specify what component the UDA is associated with. • When the GUI specifies **<ANY>**, using an asterisk (**`*`**) to specify the group in the CLI means that the attribute must be in a group, regardless of the group's name. Example: **`--uda */state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute must belong to a group, but the group is not specified. • When the GUI specifies **<IGNORE>**, for a group, using a dash (**`-`**) in the CLI means it does not matter if the UDA is in a group, or not. Example: **`--uda -/state=NJ`**. In this case, the attribute is "state" and the value is NJ. The attribute will be found regardless of whether it is in a group. **Name** This argument contains the name of the attribute, followed by an equals sign (=), followed by the value for which you are searching.
--version	-v	Displays the version of this CLI.

Long Option	Short Option	Allowable Values
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--name	-n	Name of the file for which you are searching. You can use an asterisk (*) as a wildcard character to return a range of results. If you use the asterisk as the first or only character, you must put this value in either single or double quotes.
--server	-S	Name of the server where the managed file resides. Defaults to the current server. If you specify this option, you must specify the **`--type`** option.
--type	-T	Type of managed file for which you are searching. You must specify this option if you specify a **`--server`** option. Allowable values are: • **`dns`** • **`dhcp`** • **`nis`**

Command line input sample

```
qip-cli -c managedfile,search -u admin1 -p passwd -o VitalQIP
   -n admin*
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<object-stream>
<MANAGED_FILE_REC>
  <name>admin2.sh</name>
</MANAGED_FILE_REC>
<MANAGED_FILE_REC>
  <name>admin3.sh</name>
</MANAGED_FILE_REC>
</object-stream>
```

node,add

`node,add` allows you to add a node to VitalQIP® (QIP).

> **Note:** When adding an IPv6 node, using this CLI, the CLI allocates the address. When adding an IPv4 node, however, the address must already exist.

When using a CSV file as input to this CLI, modifying the file requires more editing. A CSV file displays nodes as a series of related records. Each record contains common information about the node (for example, Node Name) and unique information about the node's components (for example, IP Address or Interface Name). If you are adding a node, you must add every record for that node.

Synopsis

```
qip-cli -c node,add <login options>
   [--level node|interface|ip|domainname]
   -I <input_NODE_REC_file>.xml|.csv [--batch]
   [--rejectfile <filename>.xml/.csv]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er][,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`node,add`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`node`**. Action is **`add`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing node records for the CLI.
--batch		Enables batch processing on multiple records of the same type in an input file.

Long Option	Short Option	Allowable Values
--rejectfile	-r	File containing the records that failed processing. The file contains the error message details, and a list of any records not added.You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	

Specific Options

Long Option	Short Option	Allowable Values
--level		Value indicating the starting element level where the node is to be added. Allowable values are as follows: • **node** • **interface** • **ip** • **domainname** Level is used to filter a large file. Without this option, the entire NODE_REC input file is processed. If the node already exists, an error occurs. By specifying a level, such as "interface", the CLI skips creating a node, and starts processing with the element level of **interfaces**.

Command line example

```
qip-cli -c node,add -u admin1 -p password -o VitalQIP
   -I addnodereqs.xml --batch
```

You can use output from the **node,get** or **node,search** command as a model for an input file for this command.

node,delete

node,delete allows you to delete a node, or an interface, IPv6 address or domain name associated with that node. It also allows you to disassociate the node from a domain.

Note: This CLI only allows you to delete an IPv6 node. If there is an IPv4 address linked to the node, the link is removed, but the IPv4 object remains.

Synopsis

```
qip-cli -c node,delete <login options>
( -N <node ID> ) | \
( -N <node ID> --interfacename <interface_name> [--nodnsdelete] )
   | \
( -N <node ID> --interfacename <interface_name> -a <IPv6 address>
   ) | \
( -N <node_ID> --interfacename <interface_name> -a <IPv6 address>
   --fqdn <fqdn> ) [-v] [-h|?]
```

Parameters

qip-cli -c **node,delete** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **node**. Action is **delete**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--nodeid	-N	Node's unique ID.
--interfacename		Name of the interface being deleted from the node. If not specified, the entire node is deleted.
--address	-a	IPv6 address to be deleted. If you specify an IPv6 address, you must also specify an interface name. If you specify only a node ID and interface name without specifying an IPv6 address, the entire interface is deleted.

Long Option	Short Option	Allowable Values
--nodnsdelete		Do not update the DNS server when this node or one of its components is deleted.
--fqdn		Fully qualified domain name to be removed from the node. Syntax for this argument is ***<hostname>.<domain_name>***

Command line examples

The following example deletes the node called ***router1***:

```
qip-cli -c node,delete -u username -p password -o VitalQIP
   --node router1
```

The following example deletes the interface called ***Ethernet0*** from the node called router1, but does not update the DNS server with the change:

```
qip-cli -c node,delete -u username -p password -o VitalQIP
   --node router1 --interfacename Ethernet0 --nodnsdelete
```

The following example deletes an IPv6 address from the interface called Ethernet0 on the node called router1:

```
qip-cli -c node,delete -u username -p password -o VitalQIP
   --node router1 --interfacename Ethernet0 -a 2002::120
```

node,get

node,get allows you to obtain an XML or CSV representation of the contents of a node.

When using a CSV file as input to this CLI, modifying the file requires more editing. A CSV file displays nodes as a series of related records. Each record contains common information about the node (for example, Node Name) and unique information about the node's components (for example, IP Address or Interface Name). If you are using a CSV file to obtain information about the node, you must be sure you include all records for that node.

Synopsis

```
qip-cli -c node,get <login options> [-n <nodename>] [-N <nodeid>]
   [-a <IPv6_Address>] [-fqdn <domain_name>]
   [-O <output_file>.xml|.csv]  [--outputxsl file1,file2]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **node,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **node**. Action is **get**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--nodename	-n	Name of the node.
--nodeid	-N	Unique ID of the node.
--address	-a	IPv6 address associated with the node.
--fqdn		Fully qualified domain name associated with the node.

Command line input sample

XML

```
qip-cli -c node,get --user admin1 --password password -o VitalQIP
   -n wsp000055wss -O nodes.xml
```

Sample Output File


```
<?xml version="1.0" encoding="UTF-8" ?>
<NODE_REC>
  <optionalAttributeList>
    <udas>
      <name>das</name>
    </udas>
  </optionalAttributeList>
  <uniqueID>48575</uniqueID>
  <interfaces>
    <optionalAttributeList>
      <udas>
        <name>block_name</name>
      </udas>
    </optionalAttributeList>
    <ipAddrs>
      <subnetStrtAddr>2400:1100::</subnetStrtAddr>
      <optionalAttributeList>
        <udas>
          <name>das</name>
        </udas>
      </optionalAttributeList>
      <domainInfo>
        <fqdn>wsp000055wss.tomain.qa.quadritek.com</fqdn>
        <pubFwdZone>true</pubFwdZone>
        <pubRevZone>true</pubRevZone>
      </domainInfo>

   <ipAddrStr>2400:1100:0000:0000:0000:0000:0000:0000</ipAddrStr>
    </ipAddrs>
    <name>Ethernet0</name>
  </interfaces>
  <interfaces>
    <optionalAttributeList>
      <udas>
        <name>block_name</name>
      </udas>
    </optionalAttributeList>
    <ipAddrs>
      <subnetStrtAddr>2400:1100::</subnetStrtAddr>
      <optionalAttributeList>
        <udas>
          <name>das</name>
        </udas>
      </optionalAttributeList>
```

```
   <ipAddrStr>2400:1100:0000:0000:0000:0000:0000:0017</ipAddrStr>
     </ipAddrs>
     <name>Ethernet1</name>
   </interfaces>
   <nodeType>Workstation</nodeType>
   <name>wsp000055wss</name>
</NODE_REC>
```

CSV

qip-cli -c node,get --user admin1 --password password -o VitalQIP -N DOCTEST00001 --format csv -O nodes.csv

Sample Output File

```
wsp000350wss.example.com,DOCTEST00001,,Workstation,Ethernet0,,200
   1:0db8:0000:0001:0000:0000:0000:00c7,,,,2001:db8:0:1::c7,wsp00
   0350wss.example.com,,true,true,,CN Views=None+HQ Views=NY+ID+ID
   No+MA Views=None+NJ Views=None+PA Views=None+RI
   Views=None,,,ID+ID No
```

node,link

node,link allows you to link an IPv6 node with an IPv4 IP address.

Synopsis

```
qip-cli -c node,link <login options> -N <nodeid>
   --toaddress <IPv4_IP_Address> --interfacename <Interface_name>
   [-v] [-h|?]
```

Note: When using this CLI to create a link to an IPv4 address, the address must exist before running this CLI.

Parameters

qip-cli -c **node,link** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **node**.
		Action is **link**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--nodeid	-N	Unique ID of the node you are linking
--interfacename		Name of the interface associated with the node you are linking.
--toaddress		IPv4 address you want to link with this node.

Command line input sample

```
qip-cli -c node,link -u admin1 -p password -o VitalQIP -N wsp2001
   --toaddress 128.100.20.2 --interfacename Ethernet0
```

node,modify

node,modify allows you to modify a node in the VitalQIP® (QIP) database.

Use **node,add** and **node,delete** to add or remove interfaces, not **node,modify**. When using XML, if you are only modifying a node's description, you need only basic XML:

```
<NODE_REC>
  <uniqueID>3</uniqueID>
  <description>Contact info is John</description>
</NODE_REC>
```

However, to modify information nested under interface, then you must supply all the node's information. This is because the **node,modify** CLI is implemented by deleting all interfaces, then adding all interfaces. So when you use **node,modify** Alcatel-Lucent recommends that you use the **node,get** CLI first to obtain the initial XML file that you edit.

When using a CSV file as input to this CLI, modifying the file requires more editing. A CSV file displays nodes as a series of related records. Each record contains common information about the node (for example, Node Name) and unique information about the node's components (for example, IP Address or Interface Name). If you are modifying information about the node, you must modify it in every record where it appears for that node.

> **Note:** When using this CLI to create or delete a link to an IPv6 address, the CLI allocates or deletes the address. When using this CLI to create a link to an IPv4 address, the address must exist before running this CLI. When using this CLI to delete a link to an IPv4 address, the link is deleted, but the IPv4 address is not deleted.

Synopsis

```
qip-cli -c node,modify <login options> [--replace <current_ID>]
   -I <input_NODE_REC_file>.xml|.csv
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **node,modify** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **node**. Action is **modify**.
Login Options		

Long Option	Short Option	Allowable Values
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing pool records as input for the CLI.
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--replace		Node's current ID. **Note:** The --replace option is only necessary when a new ID is specified by the input file.

Command line input sample

```
qip-cli -c node,modify -u admin1 -p password -o VitalQIP
   -I nodemods.csv
```

You can use output from the **`node,get`** CLI as a template for your input file for this CLI.

node,move

node,move allows you to move a node to another subnet.

Synopsis

```
qip-cli -c node,move <login options> -N nodeId
   ( --interfacename <interface name> --tonodeid <new node ID> )
   | ( --interfacename <interface name> -a <IPv6 Address>
   --tonodeid <new node ID>  --tointerface <new interface name> )
   | --interfacename <interface name> -a <IPv6 Address>
   --tosubnet <prefix> ) [-v] [-h|?]
```

Parameters

qip-cli -c **node,move** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **node**.
		Action is **move**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--address	-A	IPv6 address associated with the node.
--interfacename		Name of the interface you are moving.
--nodename	-n	Name of the node.
--nodeid	-N	Unique ID of the node.
--fqdn		Fully qualified domain name associated with the node.
--tointerface		Target interface.
--tonodeid		Target node ID
--tosubnet		Target subnet

Command line input sample

The following example moves the interface **`Ethernet0`** from the node **`wsp000055wss`** to the node **`wsp000056wss`**.

```
qip-cli -c node,move --user admin1 --password password -o VitalQIP
   -N wsp000055wss --interfacename Ethernet0
   --tonodeid wsp000056wss
```

node,search

node,search allows you to search for nodes. This search returns only a list of matching nodes, not the nodes' subcomponents (interfaces and their IP addresses). To view all the interfaces and IP addresses associated with a node,use the **node,get** CLI.

Note: If you do not specify any search values, you receive a list of all available nodes. Such a large database query can have a noticeable performance impact.

Synopsis

Synopses for both IPv6 and IPv4 node searches are as follows:

IPv6 Node Search

```
qip-cli -c node,search <login options>
   [-O <output_file>.xml|.csv] [--outputxsl file1,file2] [-V6]
   [-n <nodeName>] [-N <nodeID>] [-a <IPv6_Address>]
   [-A <IPv6_Subnet_Address>] [--ipgroupname <name>]
   [--interfacename <name>] [-m <MAC_Address>] [-T <nodeType>]
   [--description <description>] [--fqdn <fqdn>]
   [--dnsview <view>] [--uda [[[/object/]group/]name[=value]]...
   [--outputfile <file>] [--outputxsl <file1>[,<file2>...]]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header] [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

IPv4 Node Search

```
qip-cli -c node,search <login options> [-O output_file>.xml|.csv]
   [--outputxsl file1,file2] [-V4] [-n <Object_Name>]
   [-a <IPv4 Address>] [-m <MAC Address>]
   [--uda [group/]name[=value]]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]])
```

Note: When you perform an IPv4 node search, you can search for legacy UDF values, but UDF values will not be included in the <NODE_REC> output. Since the IPv4 address is a link, only the link itself is visible, not the legacy object.

Parameters

qip-cli -c **node,search** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **node**. Action is **search**.

Long Option	Short Option	Allowable Values
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--ipv4v6	-V4|V6	Specifies whether the search returns IPv4 or IPv6 nodes. If you do not specify either of these options, all nodes are returned regardless of whether they are IPv4 or IPv6 nodes.

Long Option	Short Option	Allowable Values
--name	-n	Name of the node for which you are searching. You can use an asterisk (*) as a wildcard character to return a range of results. If you use the asterisk as the first or only character, you must put this value in either single or double quotes.
--nodeid	-N	Unique ID of the node for which you are searching (IPv6 only).You can use an asterisk (*) as a wildcard character to return a range of results. If you use the asterisk as the first or only character, you must put this value in either single or double quotes.
--address	-a	IP Address of the node for which you are searching. You can use an asterisk (*) as a wildcard character to return a range of results. For IPv6 addresses, you cannot use both the asterisk and compressed address notation.
--subnet	-A	Subnet Address of the node for which you are searching(IPv6 only).You can use an asterisk (*) as a wildcard character to return a range of results.
--ipgroupname		IP Group Name to which the node belongs (IPv6 only). You can use an asterisk (*) as a wildcard character to return a range of results. For IPv6 addresses, you cannot use both the asterisk and compressed address notation.
--interfacename		Name of the node's interface for which you are searching (IPv6 only). You can use an asterisk (*) as a wildcard character to return a range of results. If you use the asterisk as the first or only character, you must put this value in either single or double quotes.
--mac	-m	MAC Address of the node for which you are searching. You can use an asterisk (*) as a wildcard character to return a range of results.

Long Option	Short Option	Allowable Values
--type	-T	Type of node for which you are searching. Allowable values are site-specific. You can obtain values for your site by running the following CLI: **`qip-cli -c list,search -T obj_class`** The following values are supplied at the time VitalQIP® (QIP) is installed. Your list may be different if it has been customized at your site: • Workstation • X-terminal • PC • Printer • Server • Wiring_Hub • Router • Bridge • Terminal_Server • Switch • Legacy_System • Gateway • Test_Equipment • Undefined • Others • Partially_Managed
--description		Text description of the node for which you are searching(IPv6 only). You can use an asterisk (*) as a wildcard character to return a range of results. If you use the asterisk as the first or only character, you must put this value in either single or double quotes.
--fqdn		Fully qualified domain name (that is, ***`objectname.domainname`***) of the node for which you are searching.You can use an asterisk (*) as a wildcard character to return a range of results. If you use the asterisk as the first or only character, you must put this value in either single or double quotes.

Long Option	Short Option	Allowable Values
--uda	-U	**`--uda`** [[***`/component`***/]***`group`***/]***`name`***[=***`value`*** The following explains how UDAs are handled in the Web GUI compared to how they are handled when using them in a CLI. **Component** ***Optional.*** The component you specify must be defined under the infrastructure element for which this CLI is searching (for example, node, pool, or block). **Group** ***Optional***. The attribute group you specify must be defined within the infrastructure or the search fails. • When the GUI indicates **<NONE>** for a group, omitting a value for a group in the CLI means "not in a group". Example: **`--uda /node//state=NJ`**, or **`--uda state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute is not in a group. The first example specifies that the attribute is associated with a node. The second example does not specify what component the UDA is associated with. • When the GUI specifies **<ANY>**, using an asterisk (**`*`**) to specify the group in the CLI means that the attribute must be in a group, regardless of the group's name. Example: **`--uda */state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute must belong to a group, but the group is not specified. • When the GUI specifies **<IGNORE>**, for a group, using a dash (**`-`**) in the CLI means it does not matter if the UDA is in a group, or not. Example: **`--uda -/state=NJ`**. In this case, the attribute is "state" and the value is NJ. The attribute will be found regardless of whether it is in a group. **Name** This argument contains the name of the attribute, followed by an equals sign (=), followed by the value for which you are searching.

Command line input sample

The following example finds any IPv6 nodes with a node ID that starts with 4857. The search results are placed in a file called *locatednodes*.

```
qip-cli -c node,search -u admin1 -p password -o VitalQIP
   -O locatednodes.xml -V6 -N 4857*
```

Sample Output File


```
<?xml version="1.0" encoding="UTF-8" ?>
<object-stream>
<NODE_REC>
  <optionalAttributeList>
    <udas>
      <name>das</name>
    </udas>
  </optionalAttributeList>
  <uniqueID>48575</uniqueID>
  <nodeType>Workstation</nodeType>
  <name>wsp000055wss</name>
</NODE_REC>
<NODE_REC>
  <optionalAttributeList>
    <udas>
      <name>das</name>
    </udas>
  </optionalAttributeList>
  <uniqueID>48576</uniqueID>
  <nodeType>ed</nodeType>
  <name>wsp000056wss</name>
</NODE_REC>
<NODE_REC>
  <optionalAttributeList>
    <udas>
      <name>das</name>
    </udas>
  </optionalAttributeList>
  <uniqueID>48577</uniqueID>
  <nodeType>Workstation</nodeType>
  <name>wsp000057wss.tomain.qa.quadritek.com</name>
</NODE_REC>
<NODE_REC>
  <optionalAttributeList>
    <udas>
      <name>das</name>
    </udas>
  </optionalAttributeList>
  <uniqueID>48578</uniqueID>
  <nodeType>Workstation</nodeType>
  <name>wsp000058wss.tomain.qa.quadritek.com</name>
</NODE_REC>
<NODE_REC>
  <optionalAttributeList>
```

```
      <udas>
        <name>das</name>
        <value>das</value>
      </udas>
    </optionalAttributeList>
    <uniqueID>48579</uniqueID>
    <nodeType>Workstation</nodeType>
    <name>wsp000059wss</name>
  </NODE_REC>
  </object-stream>
```

node,unlink

`node,unlink` allows you to unlink a node from an IPv4 IP address.

Synopsis

```
qip-cli -c node,unlink <login options> -N <nodeid>
   --interfacename <Interface_name> -a <IPv4_IP_Address> [-v]
   [-h|?]
```

Note: When using this CLI to delete a link to an IPv4 address, the link is deleted, but the IPv4 address is not deleted.

Parameters

qip-cli -c **`node,unlink`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`node`**.
		Action is **`unlink`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--nodeid	-N	Unique ID of the node you are linking
--interfacename		Name of the interface associated with the node you are linking.
--address	-a	IPv4 address you want to unlink from this node.

Command line input sample

```
qip-cli -c node,unlink -u admin1 -p password -o VitalQIP
   -N wsp2001 -a 128.100.20.2 --interfacename Ethernet0
```

object,fqdnadd

object,fqdnadd allows you to add a fully-qualified domain name view to the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c object,add <login options> -I <inputfile>.xml|csv
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er][,textqualifier=<char>][,utf8]])[--batch]
   [--rejectfile <filename>.xml [-v] [-h|?]
```

Parameters

qip-cli -c **object,fqdnadd** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **object**. Action is **fqdnadd**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing zone records for the CLI.
--batch		Enables batch processing on multiple records of the same type in an input file.
--rejectfile	-r	File containing the records that failed processing. The file contains the error message details, and a list of any records not added.You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	

Command line example

```
qip-cli -c object,fqdnadd -u admin1 -p password -o VitalQIP
   -I objrecs.xml --batch
```

You can use output from the **object,get** command as a model for an input file for this command.

object,fqdndelete

`object,fqdndelete` allows you to delete a fully-qualified domain name view.

Synopsis

```
qip-cli -c object,fqdndelete <login options>
   --fqdn <hostname.domain> --view <View_name> [-v] [-h|?]
```

Parameters

qip-cli -c **`object,fqdndelete`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`object`**.
		Action is **`fqdndelete`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--fqdn		Fully-qualified domain name.
--view	-V	Name of the view that contains the zone.

Command line example

```
qip-cli -c object,fqdndelete -u admin1 -p password -o VitalQIP
   --fqdn corporate.global --view humanres
```

object,get

object,get allows you to obtain an XML representation of the contents of a fully-qualified domain name view.

Synopsis

```
qip-cli -c object,get <login options> --fqdn <hostname.domain>
   -V <viewname> --address <IP_Address> [-O <output_file>.xml]
   [--outputxsl file1,file2] [-v] [-h|?]
```

Parameters

qip-cli -c **object,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **object**. Action is **get**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--fqdn		Fully qualified domain name.
--address	-a	IP address of the object.
--view	-V	Name of the view that contains the zone.

Command line input sample

```
qip-cli -c object,get -u admin1 -p passwd -o VitalQIP
   --fqdn corporate.global -V HumanRes
```

Sample Output File


```
<?xml version="1.0" encoding="UTF-8" ?>
<DNS_VIEW_ZONE_REC>
  <defaultTtl>86400</defaultTtl>
  <email>user@alcatel-lucent.com</email>
  <expireTime>604800</expireTime>
  <negativeCacheTtl>600</negativeCacheTtl>
  <refreshTime>21600</refreshTime>
  <retryTime>3600</retryTime>
  <zone>
    <ipv4ReverseZone>false</ipv4ReverseZone>
    <sharedIntoView>false</sharedIntoView>
    <viewName>HumanRes</viewName>
    <zoneName>EastCoast</zoneName>
  </zone>
</DNS_VIEW_ZONE_REC>
```

object,modify

`object,modify` allows you to modify a zone in the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c object,modify <login options>
   [--replace <currentname>] -I <inputfile>.xml [-v] [-h|?]
```

Parameters

qip-cli -c **`object,modify`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`object`**. Action is **`modify`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing zone records as input for the CLI.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--replace		Current object name. **Note:** The --replace option is only necessary when a new name is specified by the input file.

Command line input sample

```
qip-cli -c object,modify -u admin1 -p admin1 -o Documentation
   -I objectmods.xml
```

You can use the output file from the **`object,get`** cli as the template for the input file for this command.

pool,add

pool,add allows you to add a pool to the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c pool,add <login options> --seed|--child
   -I <input_POOL_REC_file>.xml|.csv [--batch]
   [--rejectfile <filename.xml|.csv>]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash][,head
   er][,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **pool,add** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **pool**. Action is **add**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing pool records for the CLI.
--batch		Enables batch processing on multiple records of the same type in an input file.
--rejectfile	-r	File containing the records that failed processing. The file contains the error message details, and a list of any records not added.You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--seed	-S	Specifies that the pool being added is a seed pool.
--child	-C	Specifies that the pool being added is a child pool

Command line example

```
qip-cli -c pool,add -u admin1 -p password -o VitalQIP --child
   -I addchildpool.xml --batch
```

You can use output from the **`pool,get`** command as a model for an input file for this command.

pool,delete

pool,delete allows you to delete a pool.

Synopsis

```
qip-cli -c pool,delete <login options> --pool <poolname>
   [--parentpool <parentpool>] [-v] [-h|?]
```

Parameters

qip-cli -c **pool,delete** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **pool**.
		Action is **delete**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--pool	-P	Name of the pool.
--parentpool	-L	Name of the parent pool.

Command line example

```
qip-cli -c pool,delete -u admin1 -p password -o VitalQIP
   -P hrpool2 -L hrparent
```

pool,get

pool,get allows you to obtain an XML or CSV representation of the contents of a pool.

Synopsis

```
qip-cli -c pool,get <login options> [-O <output_file>.xml|.csv]
   [--outputxsl file1,file2] --pool <poolname>
   [--parentpool <poolname>]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **pool,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **pool**. Action is **get**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--pool	-P	Name of the pool.
--parent	-L	Name of the parent pool.

Command line input sample

XML

```
qip-cli -c pool,get -u admin1 -p password -o VitalQIP -P
   seedpool115 -O pool1
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<SEED_POOL_REC>
  <allowPendingStatus>true</allowPendingStatus>
  <blockAddedNotify>true</blockAddedNotify>
  <childAllocationNotify>true</childAllocationNotify>
  <contact>administrator@labbox.com</contact>
  <freePendingBlockDays>0</freePendingBlockDays>
  <freePendingBlockWhen>FREE_AFTER_WHOIS</freePendingBlockWhen>
  <name>seedpool115</name>
  <registry>ARIN</registry>
  <v4Algorithm>BEST_FIT_FROM_START</v4Algorithm>
  <v4HDThreshold>22</v4HDThreshold>
  <v4MaxAllocation>8</v4MaxAllocation>
  <v4MinimumSparseSize>0</v4MinimumSparseSize>
  <v4ReverseZoneTemplate>Seg2 Rev Zone
   Temp</v4ReverseZoneTemplate>
  <v6Algorithm>BEST_FIT_FROM_START</v6Algorithm>
  <v6HDThreshold>43</v6HDThreshold>
  <v6MaxAllocation>17</v6MaxAllocation>
  <v6MinimumSparseSize>0</v6MinimumSparseSize>
</SEED_POOL_REC>
```

Command Line Input Example

```
qip-cli -cpool,get -u admin -p password -o VitalQIP -P
   childpool115a --parentpool seedpool115 -O pool2.xml
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<CHILD_POOL_REC>
  <childAllocationNotify>true</childAllocationNotify>
  <contact>administrator@labbox.com</contact>
  <name>childpool115a</name>
  <parentName>seedpool115</parentName>
  <recurseRequest>true</recurseRequest>
  <recurseRequestNotify>true</recurseRequestNotify>
  <v4Algorithm>BEST_FIT_FROM_START</v4Algorithm>
  <v4HDThreshold>55</v4HDThreshold>
  <v4MaxAllocation>17</v4MaxAllocation>
  <v4MinimumSparseSize>0</v4MinimumSparseSize>
  <v4Rule>reserved-rule-1</v4Rule>
  <v6Algorithm>BEST_FIT_FROM_START</v6Algorithm>
  <v6HDThreshold>87</v6HDThreshold>
  <v6MaxAllocation>16</v6MaxAllocation>
  <v6MinimumSparseSize>0</v6MinimumSparseSize>
</CHILD_POOL_REC>
```

CSV

qip-cli -cpool,get -u admin1 -p password -o VitalQIP -P DOCTEST --parentpool DOCS --format csv

Sample Output File

```
DOCTEST,DOCS,user@abcd.com,false,DOCTESTSUBORG,BEST_FIT_FROM_STAR
   T,0,31,80,BEST_FIT_FROM_START,0,31,80,,,true,true
```

pool,merge

`pool,merge` allows you to merge two pools.

Synopsis

```
qip-cli -c pool,merge <login options> --pool <poolname>
   --topool <destination_pool> [--parentpool <poolname>] [-v]
   [-h|?]
```

Parameters

qip-cli -c **`pool,merge`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`pool`**.
		Action is **`merge`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--pool	-P	Name of the origin pool.
--parentpool	-L	Name of the parent pool.
--topool		Name of the destination pool

Command line input sample

```
qip-cli -c pool,merge -u admin1 -p password -o VitalQIP
   --pool hrpool1 --topool hrpool2
```

pool,modify

pool,modify allows you to modify a pool in the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c pool,modify <login options> --seed|--child
   [--replace <current_name>]
   -I <input_SEED_POOL_REC_or_CHILD_POOL_REC_file>.xml|.csv
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **pool,modify** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **pool**. Action is **modify**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing the pool record as input for the CLI.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--child	-C	Operation applies to a child pool.
--replace		Current pool name **Note:** The --replace option is only necessary when a new name is specified by the input file.
--seed	-S	Operation applies to a seed pool.

Command line input sample

```
qip-cli -c pool,modify -u admin1 -p password -o VitalQIP --child
   -I poolmods.xml
```

pool,search

pool,search allows you to search for pools.

Synopsis

```
qip-cli -c pool,search <login options> [-O output_file]
   [--pool <name> [--child [--parentpool=<name>]]]
   [--outputxsl file1,file2]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **pool,search** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **pool**. Action is **search**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.
Specific Options		
--pool	-p	Name of the pool for which you are searching. **Note:** When you are performing a pool search, you can use an asterisk (*) as a wildcard character in your searches. The asterisk cannot be the first character of the pool name, or the only character specified. You can specify additional characters after the asterisk. For IPv6 addresses, you cannot use both the asterisk and compressed address notation.
--child	-C	Activates a different search mode which only lists child keys of the specified pool. Unlike the other pool searches, wildcards cannot be used with the **`--child`** mode as one specific pool must be specified. For non-seed pools, (that is, child pools), a **`--parentpool`** name is required.

Long Option	Short Option	Allowable Values
-uda		**`--uda`** [[***`/component`***/]***`group`***/]***`name`***[=***`value`***

The following explains how UDAs are handled in the Web GUI compared to how they are handled when using them in a CLI.

Component

Optional. The component you specify must be defined under the infrastructure element for which this CLI is searching (for example, node, pool, or block).

Group

Optional. The attribute group you specify must be defined within the infrastructure or the search fails.

- When the GUI indicates **<NONE>** for a group, omitting a value for a group in the CLI means "not in a group". Example: **`--uda /node//state=NJ`**, or **`--uda state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute is not in a group. The first example specifies that the attribute is associated with a node. The second example does not specify what component the UDA is associated with.
- When the GUI specifies **<ANY>**, using an asterisk (**`*`**) to specify the group in the CLI means that the attribute must be in a group, regardless of the group's name. Example: **`--uda */state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute must belong to a group, but the group is not specified.
- When the GUI specifies **<IGNORE>**, for a group, using a dash (**`-`**) in the CLI means it does not matter if the UDA is in a group, or not. Example: **`--uda -/state=NJ`**. In this case, the attribute is "state" and the value is NJ. The attribute will be found regardless of whether it is in a group.

Name

This argument contains the name of the attribute, followed by an equals sign (=), followed by the value for which you are searching.

Command line input sample

```
qip-cli -c pool,search -u admin -p password -o "VitalQIP
   Organization" --pool DOC*
```

Output sample


```
<?xml version="1.0" encoding="UTF-8" ?>
<object-stream>
<SEED_POOL_REC>
  <allowPendingStatus>false</allowPendingStatus>
  <blockAddedNotify>true</blockAddedNotify>
  <childAllocationNotify>false</childAllocationNotify>
  <contact>doctest@example.com</contact>
  <freePendingBlockDays>0</freePendingBlockDays>
  <freePendingBlockWhen>FREE_AFTER_WHOIS</freePendingBlockWhen>
  <name>DOCS</name>
  <v4Algorithm>BEST_FIT_FROM_START</v4Algorithm>
  <v4MaxAllocation>27</v4MaxAllocation>
  <v4MinimumSparseSize>0</v4MinimumSparseSize>
  <v6Algorithm>BEST_FIT_FROM_START</v6Algorithm>
  <v6HDThreshold>80</v6HDThreshold>
  <v6MaxAllocation>32</v6MaxAllocation>
  <v6MinimumSparseSize>0</v6MinimumSparseSize>
</SEED_POOL_REC>
<CHILD_POOL_REC>
  <childAllocationNotify>false</childAllocationNotify>
  <contact>rwedgwood@lucent.com</contact>
  <name>DOCTEST</name>
  <parentName>DOCS</parentName>
  <recurseRequest>true</recurseRequest>
  <recurseRequestNotify>true</recurseRequestNotify>
  <subnetOrg>DOCTESTSUBORG</subnetOrg>
  <v4Algorithm>BEST_FIT_FROM_START</v4Algorithm>
  <v4HDThreshold>80</v4HDThreshold>
  <v4MaxAllocation>31</v4MaxAllocation>
  <v4MinimumSparseSize>0</v4MinimumSparseSize>
  <v6Algorithm>BEST_FIT_FROM_START</v6Algorithm>
  <v6HDThreshold>80</v6HDThreshold>
  <v6MaxAllocation>31</v6MaxAllocation>
  <v6MinimumSparseSize>0</v6MinimumSparseSize>
</CHILD_POOL_REC>
<CHILD_POOL_REC>
  <childAllocationNotify>true</childAllocationNotify>
  <contact>tsample@example.com</contact>
  <grandParentName>DOCS</grandParentName>
  <name>DocTestChild</name>
  <parentName>DOCTEST</parentName>
  <recurseRequest>true</recurseRequest>
  <recurseRequestNotify>true</recurseRequestNotify>
  <v4Algorithm>BEST_FIT_FROM_START</v4Algorithm>
```

```
  <v4MaxAllocation>0</v4MaxAllocation>
  <v4MinimumSparseSize>0</v4MinimumSparseSize>
  <v6Algorithm>BEST_FIT_FROM_START</v6Algorithm>
  <v6HDThreshold>80</v6HDThreshold>
  <v6MaxAllocation>27</v6MaxAllocation>
  <v6MinimumSparseSize>0</v6MinimumSparseSize>
  <v6Rule>DocTest1</v6Rule>
</CHILD_POOL_REC>
</object-stream>
```

registry,get

qip-cli -c **`registry,get`** allows you to obtain an XML or CSV representation of the contents of an Internet registry.

Synopsis

```
qip-cli -c registry,get <login options> -T <registry_type>
   [-O output_file.xml]  [--outputxsl file1,file2]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`registry,get`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`registry`**. Action is **`get`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the format of the XML output file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--type	-T	Type of Internet registry you are getting. Allowable values are as follows: • **`afrinic`** • **`arin`** • **`apnic`** • **`ripe`**

Command line example

XML

```
qip-cli -c registry,get --user admin1 --password password
   -o VitalQIP -T arin -O arinreg.xml
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<REGISTRY_REC>
  <emailAddress>hostmaster@arin.net</emailAddress>
  <minIPv4BlockSize>29</minIPv4BlockSize>
  <minIPv6BlockSize>48</minIPv6BlockSize>
  <name>ARIN</name>
  <whoisServerName>WHOIS.ARIN.net</whoisServerName>
</REGISTRY_REC>
```

CSV

```
qip-cli -c registry,get --user admin1 --password password
   -o VitalQIP -T arin -O arinreg.csv
```

Sample Output File

```
ARIN,hostmaster@arin.net,,29,48,WHOIS.ARIN.net
```

registry,modify

qip-cli -c **`registry,modify`** allows you to modify elements of an Internet registry.

Synopsis

```
qip-cli -c registry,modify <login options>
   -I <registry_options>.xml|.csv [-v] [-h|?]
```

Parameters

qip-cli -c **`registry,modify`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`registry`**. Action is **`modify`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing input for the CLI.
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Long Option	Short Option	Allowable Values
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	

Command line example

```
qip-cli -c registry,modify --user admin1 --password password
   -o VitalQIP -I regrecord.xml
```

The output file from the **registry,get** command can be used as an input file for this command.

rule,add

qip-cli -c **`rule,add`** allows you to add one or more rules.

Synopsis

```
qip-cli -c rule,add <login options>
   -I <input_RULE_REC_file>.xml|.csv [--batch]
   [--rejectfile <filename.xml|.csv>]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`rule,add`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`rule`**. Action is **`add`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing rule record input for the CLI.
--rejectfile	-r	File containing the records that failed processing. You must specify a file name if you want to see this output. **Note:** A reject file must not already exist with the name you specify. If it does, it is not overwritten with the new information.
--batch		Enables batch processing on multiple records of the same type in an input file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.

Command line example

```
qip-cli -c rule,add -u admin1 -p password -o VitalQIP
   -I ruleparms.xml --batch --rejectfile ruleparmreject.xml
```

You can use output from the **`rule,get`** command as a basis for input for this command.

rule,copy

qip-cli -c **`rule,copy`** allows you to copy a rule.

Synopsis

```
qip-cli -c rule,copy <login options> --rule <rulename>
   --newrule <newrulename> -o <organization> [-v] [-h|?]
```

Parameters

qip-cli -c **`rule,copy`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`rule`**.
		Action is **`copy`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--rule	-R	Name of the rule to be copied.
--newrule		Name of the new rule.

Command line example

```
qip-cli -c rule,copy -u admin1 -p password -o VitalQIP
   --rule acctrule1 --newrule hrrule1
```

rule,delete

qip-cli -c **`rule,delete`** allows you to delete a rule.

Synopsis

```
qip-cli -c rule,delete <login options> --rule <rulename> [-v]
   [-h|?]
```

Parameters

qip-cli -c **`rule,delete`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`rule`**.
		Action is **`delete`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--rule	-R	Name of the rule you are deleting.

Command line example

```
qip-cli -c rule,delete -u admin1 -p password -o VitalQIP
   --rule grouprule2
```

rule,get

qip-cli -c **`rule,get`** allows you to obtain an XML or CSV representation of the contents of a rule.

Synopsis

```
qip-cli -c rule,get <login options> --rule <rulename>
   [-O <output_file>.xml|.csv]  [--outputxsl file1,file2]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`rule,get`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`rule`**. Action is **`get`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the format of the XML output file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--rule	-R	Name of the rule you are getting.

Command line example

XML

```
qip-cli -c rule,get -u admin1 -p password -o VitalQIP -R used-
   rule-1 -O rule1.xml
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<RULE_REC>
  <addrTemplate>NAR1-network-template</addrTemplate>
  <description>A mod example of a used rule</description>
  <name>used-rule-1</name>
  <requestedSize>8</requestedSize>
  <ruleLevel>NORMAL</ruleLevel>
  <ruleType>USED</ruleType>
  <v4V6Rule>IPv4</v4V6Rule>
</RULE_REC>
```

Command line example

qip-cli -c rule,get -u admin1 -p password -o VitalQIP -R site-rule-1 -O rule2.xml

Sample output file

```
<?xml version="1.0" encoding="UTF-8" ?>
<RULE_REC>
  <description>an example of a site rule</description>
  <name>site-rule-1</name>
  <requestedSize>8</requestedSize>
  <ruleComponentList>
    <ruleComponentData>
      <addrTemplate>NAR1-network-template</addrTemplate>
      <ruleType>USED</ruleType>
      <subnetBlockSize>9</subnetBlockSize>
    </ruleComponentData>
  </ruleComponentList>
  <ruleLevel>NORMAL</ruleLevel>
  <ruleType>SITE</ruleType>
  <v4V6Rule>IPv4</v4V6Rule>
```

</RULE_REC>

CSV

qip-cli -c rule,get -u admin1 -p password -o VitalQIP -R DocTest1 -O rule1.csv

Sample Output File

```
DocTest1,NORMAL,USED,DocTest1,,test,32,IPv6
```

rule,modify

qip-cli -c **`rule,modify`** allows you to modify a rule. You can modify an existing rule, or rename a modified rule.

Synopsis

```
qip-cli -c rule,modify <login options> -I <inputfile>.xml|.csv
   [--replace current_rulename]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`rule,modify`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`rule`**. Action is **`modify`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing input for the CLI.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--replace		Current name of the rule. **Note:** The --replace option is only necessary when a new name is specified by the input file.

Command line example

```
qip-cli -c rule,modify -u admin1 -p password -o VitalQIP
   -I rulemods1.xml --replace currentrulename
```

You can use the output from a **`rule,get`** CLI as a template for the input file for this CLI.

rule,search

rule,search allows you to search for one or more rules. The rule is output to a filename you specify.

Synopsis

```
qip-cli -c rule,search <login options> [--rule <rulename>]
   [-T <ruletype>] [-V 4|6] [-O <output_file>.xml|.csv]
   [--outputxsl file1,file2]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **rule,search** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **rule**. Action is **search**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--rule	-R	Name of the rule for which you are searching.
--ipv4v6	-V	Allowable values are: 4 - IPv4 6 - IPv6
--type	-T	Type of rule this CLI is processing. Allowable values are: • Used • Free • Reserved • Site

Command line example

```
qip-cli -c rule,search -u admin1 -p password -o VitalQIP -T used
   -O rule3.xml
```

Sample output file


```
<?xml version="1.0" encoding="UTF-8" ?>
<object-stream>
<RULE_REC>
  <addrTemplate>NAR1-network-template</addrTemplate>
  <description>A mod example of a used rule</description>
  <name>used-rule-1</name>
  <requestedSize>8</requestedSize>
  <ruleLevel>NORMAL</ruleLevel>
  <ruleType>USED</ruleType>
  <v4V6Rule>IPv4</v4V6Rule>
</RULE_REC>
<RULE_REC>
  <addrTemplate>NAR1-network-template</addrTemplate>
  <description>adv admin rule size 11</description>
  <name>used-rule-11</name>
  <requestedSize>11</requestedSize>
  <ruleLevel>ADVANCED</ruleLevel>
  <ruleType>USED</ruleType>
  <subnetProfTemplate>Seg2 Subnet Prof Temp</subnetProfTemplate>
  <v4V6Rule>IPv4</v4V6Rule>
</RULE_REC>
<RULE_REC>
  <addrTemplate>NAR1-network-template</addrTemplate>
  <description>expert admin rule size 8</description>
  <name>used-rule-8</name>
  <requestedSize>8</requestedSize>
  <ruleLevel>EXPERT</ruleLevel>
  <ruleType>USED</ruleType>
  <v4V6Rule>IPv4</v4V6Rule>
</RULE_REC>
<RULE_REC>
  <description>test</description>
  <name>v6test</name>
  <requestedSize>11</requestedSize>
  <ruleLevel>NORMAL</ruleLevel>
  <ruleType>USED</ruleType>
  <v4V6Rule>IPv6</v4V6Rule>
</RULE_REC>
</object-stream>
```

subnet,add

subnet,add allows you to add a subnet to the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c subnet,add <login options>
   -I <input_V6_SUBNET_REC_file>.xml|.csv [--batch]
   [--rejectfile <filename.xml|.csv>]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header][,textqualifier=<char>][,utf8]]) ([
   --when <date H:S:M>|NOW
   --jobdescription <text_description_of_job>
   --sendemail to=<email1>,<email2>,cc=<email1>,<email2>]) [-v]
   [-h|?])
```

Parameters

qip-cli -c **subnet,add** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **subnet**. Action is **add**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing subnet records for the CLI.
--batch		Enables batch processing on multiple records of the same type in an input file.
--rejectfile	-r	File containing the records that failed processing. The file contains the error message details, and a list of any records not added.You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--when		Specifies when to run this CLI. Allowable value is as follows: ***<MM/DD/YYYY H:M:S>*** or ***<DD/MM/YYYY H:M:S>***- CLI runs at the date and time specified. Date format is set based on user's location. **`NOW`** - CLI runs immediately in background.
--jobdescription		A text description of the job. You can only use this option if you have set a value for the **`--when`** option.

Long Option	Short Option	Allowable Values
--sendemail		This option allows you to specify both “To” and “CC” recipients who receive an email when the job either completes normally or terminates prematurely. You can only use this option if you have set a value for the **`--when`** option. This option has the following suboptions: **`to`** a list of comma-separated email addresses that appear in the “To” address of the email **`cc`** a list of comma-separated email addresses that appear in the “cc” address of the email.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.

Command line example

```
qip-cli -c subnet,add -u admin1 -p password -o VitalQIP
   -I addsubnet.xml --batch
```

You can use output from the **`subnet,get`** command as a model for an input file for this command.

subnet,delete

subnet,delete allows you to delete an IPv6 subnet.

Synopsis

```
qip-cli -c subnet,delete <login options> -a <IPv6_Address> [-v]
   [-h|?]
```

Parameters

qip-cli -c **subnet,delete** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **subnet**.
		Action is **delete**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--address	-a	IPv6 address of the subnet being deleted.

Command line example

```
qip-cli -c subnet,delete -u admin1 -p password -o VitalQIP
   -a 2300:11::2
```

subnet,get

subnet,get allows you to obtain an XML or CSV representation of the contents of a subnet.

Synopsis

```
qip-cli -c subnet,get <login options> [-O <output_file>.xml|.csv]
   [--outputxsl file1,file2] -a <IP_Address> [-V 4|6] [-v] [-h|?]
```

Parameters

qip-cli -c **subnet,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **subnet**. Action is **get**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--address	-a	IPv6 address of the subnetl.
--ipv4v6	-V	4 IPv4 subnet 6 IPv6 subnet. If you do not specify this option, the CLI determines the appropriate format based on the the subnet address.

Command line input sample

IPv6

```
qip-cli -c subnet,get --user admin1 --password password
   -o VitalQIP -a 1000:0016:0000:0001:: -O subnet1.xml
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<V6_SUBNET_REC>
  <allocationAlgorithm>NEXT_AVAILABLE</allocationAlgorithm>
  <defaultDomain>tomain.qa.quadritek.com</defaultDomain>
  <optionalAttributeList>
    <groups>
      <udas>
        <name>das</name>
        <value>das444</value>
      </udas>
      <name>das_group</name>
    </groups>
    <udas>
      <name>das</name>
      <value>das333</value>
    </udas>
  </optionalAttributeList>
  <startAddress>1000:16:0:1::</startAddress>
  <subnetLength>64</subnetLength>
</V6_SUBNET_REC>
```

Command line input sample

IPv4

```
qip-cli -c subnet,get --user admin1 --password password
   -o VitalQIP -a 10.0.3.0 -V 4 -O subnet1.xml
```

Sample Output File


```
<?xml version="1.0" encoding="UTF-8" ?>
<V4_SUBNET_REC>
  <warningPercent>0</warningPercent>
  <location>
    <city>Malvern</city>
    <country>USA</country>
    <state>PA</state>
    <street1>400 Lapp Road</street1>
    <zip>19355</zip>
  </location>
  <domains>example.com</domains>
  <mask>255.255.255.128</mask>
  <checkUse>true</checkUse>
  <dhcpOptTempl>general</dhcpOptTempl>
  <allowDHCPClientsModifyDynamicObjectRRs>YES</allowDHCPClientsMo
   difyDynamicObjectRRs>
  <name>test</name>
  <status>USED</status>
  <networkAddress>10.0.0.0</networkAddress>
  <tftpServer>Object2.example.com</tftpServer>
  <showUsed>false</showUsed>
  <subnetLength>25</subnetLength>
  <dhcpServer>doctest_dhcp.example.com</dhcpServer>
  <address>10.0.3.0</address>
  <hardwareType>ETHERNET</hardwareType>
  <warning>NONE</warning>
</V4_SUBNET_REC>
```

subnet,merge

`subnet,merge` allows you to merge two subnets.

Synopsis

```
qip-cli -c subnet,merge <login options> --a <IP_Address>
   --toaddress <destination_subnet> [-v] [-h|?]
```

Parameters

qip-cli -c **`subnet,merge`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`subnet`**.
		Action is **`merge`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--address	-a	Address of the origin subnet.
--toaddress		Address of the destination subnet

Command line input sample

```
qip-cli -c subnet,merge -u admin1 -p password -o VitalQIP
   -a 2004:120:: --toaddress 2004:121::
```

subnet,modify

subnet,modify allows you to modify a subnet in the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c subnet,modify <login options>
   [--replace <currentname>] -I <inputfile>.xml|.csv
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **subnet,modify** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **subnet**. Action is **modify**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing subnet records as input for the CLI.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.
Specific Options		
--replace		Subnet's current name. **Note:** The --replace option is only necessary when a new name is specified by the input file.

Command line input sample

```
qip-cli -c subnet,modify -u admin1 -p password -o VitalQIP
   -I subnetmods.xml
```

You can use the output from a **`subnet,get`** CLI as a template for input for this CLI.

subnet,renumber

subnet,renumber allows you to change an IPv6 subnet's IP address.

Note: This CLI can only be used with IPv6 subnets.

Synopsis

```
qip-cli -c subnet,renumber <login options> --address <IP/Prefix>
   --toaddress <IP/prefix> ([ --when <date H:S:M>|NOW
   --jobdescription <text_description_of_job>
   --sendemail to=<email1>,<email2>,cc=<email1>,<email2>]) [-v]
   [-h|?]
```

Parameters

qip-cli -c **subnet,renumber** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **subnet**. Action is **renumber**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--address	-a	IPv6 subnet address.
--toaddress		New IPv6 subnet address.
--when		Specifies when to run this CLI. Allowable value is as follows: ***<MM/DD/YYYY H:M:S>*** or ***<DD/MM/YYYY H:M:S>***- CLI runs at the date and time specified. Date format is set based on user's location. **NOW** - CLI runs immediately in background.

Long Option	Short Option	Allowable Values
--jobdescription		A text description of the job. You can only use this option if you have set a value for the **`--when`** option.
--sendemail		This option allows you to specify both “To” and “CC” recipients who receive an email when the job either completes normally or terminates prematurely. You can only use this option if you have set a value for the **`--when`** option. This option has the following suboptions: **`to`** a list of comma-separated email addresses that appear in the “To” address of the email **`cc`** a list of comma-separated email addresses that appear in the “cc” address of the email.

Command line example

```
qip-cli -c subnet,renumber -u admin1 -p password -o VitalQIP
   --address 2300:0:c0/42 --toaddress 2300:0:80/42
   --jobdescription “Renum 2300 subnet”
   --sendemail to=user@abcd.com,cc=admin@abcd.com
```

subnet,search

subnet,search allows you to search for subnets.

Synopsis

```
qip-cli -c subnet,search <login options>
   [-O <output_file>.xml|.csv] [-n <subnet_name>]
   [-a <IP_Address>] [--length <prefix>] [-V 4|6]
   [--uda =[group/]name[=value]...] [--udf=name=value]
   [--outputxsl file1,file2]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header][,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **subnet,search** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **subnet**. Action is **search**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
	-n	Name of the subnet for which you are searching. **Note:** When you are performing a search, you can use an asterisk (*) as a wildcard character in your searches. The asterisk cannot be the first character of the field, or the only character specified. You can specify additional characters after the asterisk.
-	-a	Address of the subnet for which you are searching. **Note:** IPv6 addresses must be completely typed out. You cannot use the compressed notation (with double colons to indicate a string of zeros) in this field.
--length		Prefix of the subnet.

Long Option	Short Option	Allowable Values
--ipv4v6	-V	4 IPv4 subnet 6 IPv6 subnet. If you do not specify this option, the CLI determines the appropriate format based on the the subnet address.
--udf		Name and value of a user-defined field (UDF) associated with a subnet. UDFs can be defined by legacy CLIs or the GUI (thick client) for IPv4 subnets only. The syntax is as follows: **`name[=value]`** • **`Name`** is the name of the UDF • **`Value`** is the value contained within it.

Long Option	Short Option	Allowable Values
--uda		**`--uda`** [[***/component***/]***group***/]***name***[=***value***

The following explains how UDAs are handled in the Web GUI compared to how they are handled when using them in a CLI.

Component

Optional. The component you specify must be defined under the infrastructure element for which this CLI is searching (for example, node, pool, or block).

Group

Optional. The attribute group you specify must be defined within the infrastructure or the search fails.

- When the GUI indicates **<NONE>** for a group, omitting a value for a group in the CLI means "not in a group". Example: **`--uda /node//state=NJ`**, or **`--uda state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute is not in a group. The first example specifies that the attribute is associated with a node. The second example does not specify what component the UDA is associated with.
- When the GUI specifies **<ANY>**, using an asterisk (**`*`**) to specify the group in the CLI means that the attribute must be in a group, regardless of the group's name. Example: **`--uda */state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute must belong to a group, but the group is not specified.
- When the GUI specifies **<IGNORE>**, for a group, using a dash (**`-`**) in the CLI means it does not matter if the UDA is in a group, or not. Example: **`--uda -/state=NJ`**. In this case, the attribute is "state" and the value is NJ. The attribute will be found regardless of whether it is in a group.

Name

This argument contains the name of the attribute, followed by an equals sign (=), followed by the value for which you are searching.

Command line input sample

```
qip-cli -c subnet,search -u admin -p password -o "VitalQIP
   Organization" -a 10.0.*
```

Output example


```
<?xml version="1.0" encoding="UTF-8" ?>
<object-stream>
<V4_SUBNET_REC>
  <warningPercent>0</warningPercent>
  <domains>example.com</domains>
  <mask>255.255.255.0</mask>
  <checkUse>true</checkUse>
  <dhcpOptTempl>general</dhcpOptTempl>

   <allowDHCPClientsModifyDynamicObjectRRs>YES</allowDHCPClientsM
   odifyDynamicObjectRRs>
  <status>USED</status>
  <networkAddress>10.0.0.0</networkAddress>
  <showUsed>false</showUsed>
  <subnetLength>24</subnetLength>
  <dhcpServer>doctest_dhcp.example.com</dhcpServer>
  <address>10.0.3.0</address>
  <hardwareType>ETHERNET</hardwareType>
  <warning>NONE</warning>
</V4_SUBNET_REC>
<V4_SUBNET_REC>
  <warningPercent>0</warningPercent>
  <domains>example.com</domains>
  <mask>255.255.255.0</mask>
  <checkUse>true</checkUse>
  <dhcpOptTempl>general</dhcpOptTempl>

   <allowDHCPClientsModifyDynamicObjectRRs>SAME_AS_GLOBAL_POLICY<
   /allowDHCPClientsModifyDynamicObjectRRs>
  <status>USED</status>
  <networkAddress>10.0.0.0</networkAddress>
  <showUsed>false</showUsed>
  <subnetLength>24</subnetLength>
  <dhcpServer>doctest_dhcp.example.com</dhcpServer>
  <address>10.0.5.0</address>
  <hardwareType>ETHERNET</hardwareType>
  <warning>NONE</warning>
</V4_SUBNET_REC>
<V4_SUBNET_REC>
  <warningPercent>0</warningPercent>
  <mask>255.255.224.0</mask>
  <checkUse>false</checkUse>
  <dhcpOptTempl>general</dhcpOptTempl>
  <sharedNetwork>DOCTEST3</sharedNetwork>
  <allowDHCPClientsModifyDynamicObjectRRs>SAME_AS_GLOBAL_POLICY</
```

```
 allowDHCPClientsModifyDynamicObjectRRs>
<primaryInterface>false</primaryInterface>
<status>USED</status>
<networkAddress>10.0.0.0</networkAddress>
<showUsed>false</showUsed>
<subnetLength>19</subnetLength>
<dhcpServer>doctest_dhcp.example.com</dhcpServer>
<address>10.0.32.0</address>
<hardwareType>ETHERNET</hardwareType>
<warning>NONE</warning>
</V4_SUBNET_REC>
</object-stream>
```

subnet,split

subnet,split allows you to divide one subnet into multiple subnets.

Synopsis

```
qip-cli -c subnet,split <login options> -a <IP_Address>
   --tolength </prefix> [--todomain <domain_name>]
   ([--when <date H:S:M>|NOW
   --jobdescription <text_description_of_job>
   --sendemail to=<email1>,<email2>,cc=<email1>,<email2>]) [-v]
   [-h|?]
```

Parameters

qip-cli -c **subnet,split** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **subnet**. Action is **split**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--address	-a	Address of the subnet you are splitting.
--tolength		Prefix of the new subnet. This must be a smaller prefix (higher numerically) than the subnet being split.
--todomain		Domain to which you want to assign the new subnets.

Long Option	Short Option	Allowable Values
--when		Specifies when to run this CLI. Allowable value is as follows: ***<MM/DD/YYYY H:M:S>*** or ***<DD/MM/YYYY H:M:S>***- CLI runs at the date and time specified. Date format is set based on user's location. **NOW** - CLI runs immediately in background.
--jobdescription		A text description of the job. You can only use this option if you have set a value for the **--when** option.
--sendemail		This option allows you to specify both "To" and "CC" recipients who receive an email when the job either completes normally or terminates prematurely. You can only use this option if you have set a value for the **--when** option. This option has the following suboptions: **to** a list of comma-separated email addresses that appear in the "To" address of the email **cc** a list of comma-separated email addresses that appear in the "cc" address of the email.

Command line input sample

```
qip-cli -c subnet,split -u admin1 -p password -o VitalQIP
   -a 2000:abcd::1 --tolength /40
```

template,add

`template,add` allows you to add a template to the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c template,add <login options> -T <type>
   -I <inputfile>.xml|.csv [--batch]
   [--rejectfile <filename.xml|.csv>]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header][,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`template,add`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`template`**. Action is **`add`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing template records for the CLI. Based on the chosen --type, the XML or CSV input will be: **`ADDR_TEMPL_REC, REV_ZONE_TEMPL_REC`**, or **`SUBNET_PROF_TEMPL_REC`**.
--batch		Enables batch processing on multiple records of the same type in an input file.
--rejectfile	-r	File containing the records that failed processing. The file contains the error message details, and a list of any records not added.You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--type	-T	Type of template you are adding. Allowable values are as follows: • **`address`** • **`subnetprofile`** • **`reversezone`**

Command line example

```
qip-cli -c template,add -u admin1 -p password -o VitalQIP
   -T address -I addaddrtemp.xml --batch
```

You can use output from the **`template,get`** command as a model for an input file for this command.

template,copy

qip-cli -c **`template,copy`** allows you to copy a template.

Synopsis

```
qip-cli -c template,copy <login options> --name <templatename>
   --toname <newtemplatename> [-v] [-h|?]
```

Parameters

qip-cli -c **`template,copy`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`template`**. Action is **`copy`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--name	-n	Name of the template to be copied.
--toname		Name of the new template.
--type	-T	Type of template you are copying. Allowable values are as follows: • **`address`** • **`subnetprofile`** • **`reversezone`**

Command line example

```
qip-cli -c template,copy -u admin1 -p password -o VitalQIP
   --name accttemplate --toname accttemplate1 -T address
```

template,delete

template,delete allows you to delete a template.

Synopsis

```
qip-cli -c template,delete <login options> -T <type> -n <name>
   [-v] [-h|?]
```

Parameters

qip-cli -c **template,delete** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **template**. Action is **delete**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--type	-T	Type of template you are deleting. Allowable values are as follows: • **address** • **subnetprofile** • **reversezone**
--name	-n	Name of the template you are deleting.

Command line example

```
qip-cli -c template,delete -u admin1 -p password -o VitalQIP
   -T address --name hrtemplate2
```

template,get

`template,get` allows you to obtain an XML or CSV representation of the contents of a template.

Synopsis

```
qip-cli -c template,get <login options>
   [-O <output_file>.xml|.csv]  [--outputxsl file1,file2]
   -T <type> --name <templatename>
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`template,get`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`template`**. Action is **`get`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h | -?	Displays syntax and Options for this CLI.
Specific Options		
--type	-T	Type of template you are getting. Allowable values are as follows: • **`address`** • **`subnetprofile`** • **`reversezone`**
--name	-n	Name of the template you are getting.

Command line input sample

```
qip-cli -c template,get --user admin1 --password password
   -o VitalQIP -T address -n hrtemp2 -O templts.xml
```

Sample Output File


```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ADDR_TEMPL_REC>
  <description>lateda</description>
  <name>hrtemp2</name>
  <v4V6Type>IPv6</v4V6Type>
  <v6ManagedRanges>
    <managedRangeData>
      <addressType>STATIC</addressType>
      <domain>tomain.qa.quadritek.com</domain>
      <endOffset>0</endOffset>
      <nodeType>Workstation</nodeType>
      <numberOfObjects>23</numberOfObjects>
      <startOffset>0</startOffset>
    </managedRangeData>
    <managedRangeData>
      <addressType>STATIC</addressType>
      <domain>tomain.qa.quadritek.com</domain>
      <endOffset>0</endOffset>
      <intf>Ethernet0</intf>
      <nodeType>Workstation</nodeType>
      <numberOfObjects>2</numberOfObjects>
      <startOffset>0</startOffset>
    </managedRangeData>
    <managedRangeData>
      <addressType>STATIC</addressType>
      <domain>tomain.qa.quadritek.com</domain>
      <endOffset>0</endOffset>
      <nodeType>Workstation</nodeType>
      <numberOfObjects>34</numberOfObjects>
      <startOffset>0</startOffset>
    </managedRangeData>
  </v6ManagedRanges>
</ADDR_TEMPL_REC>
```

template,modify

`template,modify` allows you to modify a template in the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c template,modify <login options>
   -I <intput_file>.xml|.csv -T <type> [--replace <currentname>]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`template,modify`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`template`**. Action is **`modify`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing template records as input for the CLI. Based on the chosen --type, the XML or CSV input will be: **`ADDR_TEMPL_REC, REV_ZONE_TEMPL_REC`**, or **`SUBNET_PROF_TEMPL_REC`**.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the input file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--type	-T	Type of template you are modifying. Allowable values are as follows: • **`address`** • **`subnetprofile`** • **`reversezone`**
--replace		Template's current name. **Note:** The --replace option is only necessary when a new name is specified by the input XML or CSV file.

Command line input sample

```
qip-cli -c template,modify -u admin1 -p password -o VitalQIP
   -T address -I templatemods.xml
```

template,search

`template,search` allows you to search for templates.

Synopsis

```
qip-cli -c template,search <login options>
   [-O <output_file>.xml|.csv] [-T <type>] [-V 4|6]
   [--outputxsl file1,file2]
   ([--format xml|csv[,columndelimiter=<char>][,comment=<char>]
   [,recorddelimiter=<char>][,escapemode=doubled|backslash]
   [,header]
   [,textqualifier=<char>][,utf8]]) [-v] [-h|?]
```

Parameters

qip-cli -c **`template,search`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`template`**. Action is **`search`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--ipv4v6	-V	Allowable values are: **`4`** - IPv4 address **`6`** - IPv6 address
--type	-T	Type of template for which you are searching. Allowable values are as follows: • **`address`** • **`subnetprofile`** • **`reversezone`**

Command line input sample

```
qip-cli -c template,search -u admin1 -p password -o VitalQIP
   -O locatedtemplates.xml -T address -V 6
```

Sample Output File


```
<?xml version="1.0" encoding="UTF-8" ?>
<object-stream>
<ADDR_TEMPL_REC>
  <description>lateda</description>
  <name>hrtemp2</name>
  <v4V6Type>IPv6</v4V6Type>
  <v6ManagedRanges>
    <managedRangeData>
      <addressType>STATIC</addressType>
      <domain>tomain.qa.quadritek.com</domain>
      <endOffset>0</endOffset>
      <nodeType>Workstation</nodeType>
      <numberOfObjects>23</numberOfObjects>
      <startOffset>0</startOffset>
    </managedRangeData>
    <managedRangeData>
      <addressType>STATIC</addressType>
      <domain>tomain.qa.quadritek.com</domain>
      <endOffset>0</endOffset>
      <intf>Ethernet0</intf>
      <nodeType>Workstation</nodeType>
      <numberOfObjects>2</numberOfObjects>
      <startOffset>0</startOffset>
    </managedRangeData>
    <managedRangeData>
      <addressType>STATIC</addressType>
      <domain>tomain.qa.quadritek.com</domain>
      <endOffset>0</endOffset>
      <nodeType>Workstation</nodeType>
      <numberOfObjects>34</numberOfObjects>
      <startOffset>0</startOffset>
    </managedRangeData>
  </v6ManagedRanges>
</ADDR_TEMPL_REC>
</object-stream>
```

uda, add

uda,add allows you to add UDA definitions.

Synopsis

```
qip-cli -c uda,add [Login Options] -I<input UDA_DEFINITION_REC
   file> [--batch] [--rejectfile=<file>]
   [--format=xml|csv[,suboptions]]
```

Parameters

qip-cli -c **uda,add** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **uda**. Action is **add**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Input Filename
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **--format** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **xml** (default) **csv** If you specify **csv**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Long Option	Short Option	Allowable Values
--rejectfile	-r	Output reject filename
--batch		Enables batch processing on multiple records of the same type in an input file.

Command line input sample

```
qip-cli -c uda,add -u admin1 -p password -o VitalQIP
   -I udadefrec.xml --batch
```

uda,associate

uda,associate allows you to manage a UDA's association with different infrastructure types.

Synopsis

```
qip-cli -c uda,associate -T <infrastructureType> -n<uda>...
   [login options]
```

Parameters

qip-cli -c **uda,associate** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **uda**. Action is **associate**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--infrastructureType	-T	Specifies the type of infrastructure. Values can be **ac_device**, **ac_subscriber**, **block**, **dc_server**, **dhcp_server**, **dns_server**, **dns_view**, **interface**, **ipv4_address**, **ipv6_address**, **ipv4_subnet**, **ipv6_subnet**, **managed_file**, **myview**, **network**, **node**, **organization**, **pool**, **subnet_organization**, or **zone**.
--name	-n	UDA's name. If working with multiple UDAs, the user may either specify multiple **-n** options on the same command line (as indicated by the ellipses in the above syntax), or use individual **qip-cli** commands for each UDA.

Command line input sample

```
qip-cli -c uda,associate -T ac_device -n uda1 -u admin1 -p
   password -o VitalQIP
```

uda, countassociations

uda,countassociations allows you to view the number of infrastructure types associated with the UDA.

Synopsis

```
qip-cli -c uda,countassociations -n<uda>...  [login options]
```

Parameters

qip-cli -c **uda,countassociations** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **uda**. Action is **countassociations**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--name	-n	UDA's name. If working with multiple UDAs, the user may either specify multiple **-n** options on the same command line (as indicated by the ellipses in the above syntax), or use individual **qip-cli** commands for each UDA.

Command line input sample

```
qip-cli -c uda,countassociations -n uda1 -u admin1 -p password
   -o VitalQIP
```

uda, delete

`uda,delete` allows you to delete UDA definitions.

Synopsis

```
qip-cli uda,delete -n<udaName> [login options]
```

Parameters

qip-cli -c **`uda,delete`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`uda`**. Action is **`delete`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--name	-n	UDA's name.

Command line input sample

```
qip-cli -c uda,dissociate -T ac_device -n uda1 -u admin1 -p
   password -o VitalQIP
```

uda, dissociate

uda,dissociate allows you to dissociate a UDA with different infrastructure types.

Synopsis

```
qip-cli -c uda,dissociate -T <infrastructureType> -n<uda>...
[login options]
```

Parameters

qip-cli -c **uda,dissociate** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **uda**. Action is **dissociate**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--infrastructureType	-T	Specifies the type of infrastructure. Values can be **ac_device**, **ac_subscriber**, **block**, **dc_server**, **dhcp_server**, **dns_server**, **dns_view**, **interface**, **ipv4_address**, **ipv6_address**, **ipv4_subnet**, **ipv6_subnet**, **managed_file**, **myview**, **network**, **node**, **organization**, **pool**, **subnet_organization**, or **zone**.
--name	-n	UDA's name. If working with multiple UDAs, the user may either specify multiple **-n** options on the same command line (as indicated by the ellipses in the above syntax), or use individual **qip-cli** commands for each UDA.

Command line input sample

```
qip-cli -c uda,dissociate -T ac_device device1 -n uda1 -u admin1
   -p password -o VitalQIP
```

uda, findassociations

uda,findassociations allows you to view the infrastructure types associated with UDAs.

Synopsis

```
qip-cli -c uda,findassociations -n<uda>
[login options]
```

Parameters

qip-cli -c **uda,findassociations** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **uda**. Action is **findassociations**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--name	-n	UDA's name.

Command line input sample

```
qip-cli -c uda,findassociations -n uda1 -u admin1 -p password
   -o VitalQIP
```

uda, get

`uda,get` allows you to add UDA definitions.

Synopsis

```
qip-cli -c uda,get -n<udaName> [--format=xml|csv[,suboptions] \
[login options]
```

Parameters

qip-cli -c **`uda,get`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`uda`**.
		Action is **`get`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--name	-n	UDA's name.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Command line input sample

```
qip-cli -c uda,get -n testuda.xml -u admin1 -p password
   -o VitalQIP
```

uda, list

uda,list allows you to list UDA and UDA groups associated with specified infrastructure types.

Synopsis

```
qip-cli -c uda,list -T <infrastructureType> [login options] \
   [-O outputFile] [--format=xml|csv[,suboptions]]
```

Parameters

qip-cli -c **uda,list** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **uda**. Action is **list**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--infrastructureType	-T	Specifies the type of infrastructure. Values can be **ac_device**, **ac_subscriber**, **block**, **dc_server**, **dhcp_server**, **dns_server**, **dns_view**, **interface**, **ipv4_address**, **ipv6_address**, **ipv4_subnet**, **ipv6_subnet**, **managed_file**, **myview**, **network**, **node**, **organization**, **pool**, **subnet_organization**, or **zone**.
--outputfile	-O	Name of the file containing output from the CLI.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Command line input sample

```
qip-cli -c uda,list -T ac_device --outputxsl=device1.xml --
   format=xml -u admin1 -p password -o VitalQIP
```

uda,modify

uda,modify allows you to modify UDA definitions.

Synopsis

```
qip-cli uda,modify -I<udaFile>  [--format=xml|csv[,suboptions]] \
[login options]
```

Parameters

qip-cli -c **uda,modify** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **uda**. Action is **modify**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Input Filename
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **--format** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **xml** (default) **csv** If you specify **csv**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Command line input sample

```
qip-cli -c uda,modify -I udadefrec.xml -u admin1 -p password
   -o VitalQIP
```

uda,search

uda,search allows you to search UDA definitions.

Synopsis

```
qip-cli uda,search -n<udaPattern>  [login options]
```

Parameters

qip-cli -c **uda,search** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **uda**.
		Action is **search**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--name	-n	UDA's name.

Command line input sample

```
qip-cli -c uda,search -n testuda -u admin1 -p password -o VitalQIP
```

uda, sequence

uda,sequence imposes a specific ordering to UDAs and UDA groups already associated with an infrastructure type.

Synopsis

```
qip-cli -c uda,sequence -T< infrastructureType> \
-I <inputFile containing UDA_DEFINITION_LIST_REC> \
[--format=xml|csv[,suboptions]] [login options]
```

Parameters

qip-cli -c **uda,sequence** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **uda**. Action is **sequence**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Input Filename
--infrastructureType	-T	Specifies the type of infrastructure. Values can be **ac_device**, **ac_subscriber**, **block**, **dc_server**, **dhcp_server**, **dns_server**, **dns_view**, **interface**, **ipv4_address**, **ipv6_address**, **ipv4_subnet**, **ipv6_subnet**, **managed_file**, **myview**, **network**, **node**, **organization**, **pool**, **subnet_organization**, or **zone**.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Command line input sample

```
qip-cli -c uda,sequence -T ac_device -I device1.xml -u admin1 -p
   password -o VitalQIP
```

uda,set

uda,set allows you to modify specific UDA values.

Synopsis

```
qip-cli -c uda,set [Login Options]  -U
   [<group>/]<udaName>[=<value>]... \
(-Tac_device -m <mac>) | \
(-Tac_subscriber -n <name>) | \
(-Tblock [--parentpool=<pool name>] (-P <pool name> | --
   status=<block status>) -A <startAddr/prefixLength>) | \
(-Tdc_server -n <name> -d <domain name>) | \
(-Tdhcp_server -n <name> -d <domain name>) | \
(-Tdns_server -n <name> -d <domain name>) | \
(-Tdns_view -n <name>) | \
(-Tinterface -n <interface name> -N <node ID>) | \
(-Tipv4_address -a <address>) | \
(-Tipv6_address -a <address>) | \
(-Tipv4_subnet -a <address>) | \
(-Tipv6_subnet -A <startAddress/prefixLength>) | \
(-Tmanaged_file -n <name>) | \
(-Tmyview -n<name>) | \
(-Tnetwork -a <start address>) | \
(-Tnode -N <node ID>) | \
(-Torg -n <organization>) | \
(-Tpool -n <pool name> [--parentpool=<pool name>]) | \
(-Tsubnet_org -n <name>) | \
(-Tuser -n <name>) | \
(-Tzone -n <name> [--view=<view>] [--ipv4reversezone])
```

Parameters

qip-cli -c **uda,set** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **uda**. Action is **set**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		

Long Option	Short Option	Allowable Values
--type	-T	Specify the infrastructure instance, depending on the infrastructure type. Allowable values are as follows: • AC device • AC subscriber • Block • DC server • DNS server • DHCP server • DNS view • Interface • IPv4 address • IPv6 address • IPv4 subnet • IPv6 subnet • Managed file • Myview • Network • Node • Organization • Pool • Subnet organization • User • Zone
--mac	-m	MAC Address of the device
--address	-a	Valid IPv4 or IPv6 address.

Long Option	Short Option	Allowable Values
--name	-n	Allowable values can be names of one of the following: • Subscriber • DC server • DNS server • DHCP server • DNS view • Interface • Managed file • Myview • Organization • Pool • Subnet organization • User • Zone
--pool	-P	Name of the pool

Long Option	Short Option	Allowable Values
--uda		**`--uda`** [[***`/component`***/]***`group`***/]***`name`***[=***`value`*** The following explains how UDAs are handled in the Web GUI compared to how they are handled when using them in a CLI. **Component** ***Optional.*** The component you specify must be defined under the infrastructure element for which this CLI is searching (for example, node, pool, or block). **Group** ***Optional***. The attribute group you specify must be defined within the infrastructure or the search fails. • When the GUI indicates **<NONE>** for a group, omitting a value for a group in the CLI means "not in a group". Example: **`--uda /node//state=NJ`**, or **`--uda state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute is not in a group. The first example specifies that the attribute is associated with a node. The second example does not specify what component the UDA is associated with. • When the GUI specifies **<ANY>**, using an asterisk (**`*`**) to specify the group in the CLI means that the attribute must be in a group, regardless of the group's name. Example: **`--uda */state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute must belong to a group, but the group is not specified. • When the GUI specifies **<IGNORE>**, for a group, using a dash (**`-`**) in the CLI means it does not matter if the UDA is in a group, or not. Example: **`--uda -/state=NJ`**. In this case, the attribute is "state" and the value is NJ. The attribute will be found regardless of whether it is in a group. **Name** This argument contains the name of the attribute, followed by an equals sign (=), followed by the value for which you are searching.
--parentpool	-L	Parent pool name

Long Option	Short Option	Allowable Values
--status	-S	Block status Either pool name or block status is required, but not both.
--subnet	-A	StartAddress/prefixLength
--domain	-d	Name of the domain
--nodeid	-N	Node's ID
--view	-V	Name of the view containing the zone
--ipv4reversezone		Specifies that the zone is an IPv4 reverse zone.

Command line input sample

```
qip-cli -c uda,set -Tdc_server -n device1 --domain d -u admin1 -p
   password -o VitalQIP --uda ud
```

udagroup, add

udagroup,add allows you to add UDA groups.

Synopsis

```
qip-cli -c udagroup,add -I<udaGroupFile \
[--format=xml|csv[,suboptions]] \
[--batch] [--rejectfile=<file>] [login options]
```

Parameters

qip-cli -c **udagroup,add** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **udagroup**. Action is **add**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Input Filename

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--rejectfile	-r	Output reject filename
--batch		Enables batch processing on multiple records of the same type in an input file.

Command line input sample

```
qip-cli -c udagroup,add -u admin1 -p password -o VitalQIP
   -I udagroupfile1.xml --batch
```

udagroup,associate

udagroup,associate allows you to manage a UDA group's association with different infrastructure types.

Synopsis

```
qip-cli -c udagroup,associate -T <infrastructureType> -n<grp>...
   [login options]
```

Parameters

qip-cli -c **udagroup,associate** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **udagroup**. Action is **associate**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--infrastructureType	-T	Specifies the type of infrastructure. Values can be **ac_device**, **ac_subscriber**, **block**, **dc_server**, **dhcp_server**, **dns_server**, **dns_view**, **interface**, **ipv4_address**, **ipv6_address**, **ipv4_subnet**, **ipv6_subnet**, **managed_file**, **myview**, **network**, **node**, **organization**, **pool**, **subnet_organization**, or **zone**.
--name	-n	UDA group's name.

Command line input sample

```
qip-cli -c udagroup,associate -T ac_device -n udagroup1 -u admin1
   -p password -o VitalQIP
```

udagroup, countassociations

udagroup,countassociations allows you to view the number of infrastructure types associated with the UDA group.

Synopsis

```
qip-cli -c udagroup,countassociations -n<grp>...  [login options]
```

Parameters

qip-cli -c **udagroup,countassociations** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **udagroup**.
		Action is **countassociations**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--name	-n	UDA's name. If working with multiple UDAs, the user may specify the group name.

Command line input sample

```
qip-cli -c udagroup,countassociations -n udagroup1 -u admin1
   -p password -o VitalQIP
```

udagroup, delete

udagroup,delete allows you to delete UDA groups.

Synopsis

```
qip-cli udagroup,delete -n<groupName> [login options]
```

Parameters

qip-cli -c **udagroup,delete** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **udagroup**.
		Action is **delete**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--groupName	-n	UDA group's name.

Command line input sample

```
qip-cli -c udagroup,delete -n udagroup1 -u admin1 -p password
   -o VitalQIP
```

udagroup, dissociate

udagroup,dissociate allows you to dissociate a UDA group from different infrastructure types.

Synopsis

```
qip-cli -c udagroup,dissociate -T <infrastructureType> -n<grp>...
[login options]
```

Parameters

qip-cli -c **udagroup,dissociate** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **udagroup**. Action is **dissociate**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--infrastructureType	-T	Specifies the type of infrastructure. Values can be **ac_device**, **ac_subscriber**, **block**, **dc_server**, **dhcp_server**, **dns_server**, **dns_view**, **interface**, **ipv4_address**, **ipv6_address**, **ipv4_subnet**, **ipv6_subnet**, **managed_file**, **myview**, **network**, **node**, **organization**, **pool**, **subnet_organization**, or **zone**.
--grp	-n	UDA group's name.

Command line input sample

```
qip-cli -c udagroup,dissociate -T ac_device -n udagroup1 -u admin1
   -p password -o VitalQIP
```

udagroup, findassociations

udagroup,findassociations allows you to view the infrastructure types associated with UDAs.

Synopsis

```
qip-cli -c udagroup,findassociations -n<grp>
[login options]
```

Parameters

qip-cli -c **udagroup,findassociations** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **udagroup**. Action is **findassociations**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--grp	-n	UDA's name. If working with multiple UDAs, the user may specify the group name.

Command line input sample

```
qip-cli -c udagroup,findassociations -n udagroup1 -u admin1
   -p password -o VitalQIP
```

udagroup, get

udagroup,get allows you to get UDA group definitions.

Synopsis

```
qip-cli -c udagroup,get -n<groupName> [XML and CSV output options]
[login options]
```

Parameters

qip-cli -c **udagroup,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **udagroup**. Action is **get**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--grp	-n	UDA group's name.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Command line input sample

```
qip-cli -c udagroup,get -n udagroup1 -u admin1 -p password
   -o VitalQIP
```

udagroup,modify

udagroup,modify allows you to modify UDA definitions.

Synopsis

```
qip-cli udagroup,modify -I<udaGroupFile> [XML and CSV output
   options]) | \((-n <group>  (--append <uda> | --remove <
   uda>)... ) \ [login options]
```

Parameters

qip-cli -c **udagroup,modify** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **udagroup**.
		Action is **modify**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Input Filename
--name	-n	UDA group's name.
--append		Add a UDA definition.
--remove		Remove an existing UDA definition.

Long Option	Short Option	Allowable Values
--format		Specifies whether the format of the output file is either xml or csv. If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`xml`** (default) **`csv`** If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.

Command line input sample

```
qip-cli -c udagroup,modify -I udagroupfile1.xml -u admin1
   -p password -o VitalQIP
```

udagroup,search

udagroup,search allows you to search UDA groups.

Synopsis

```
qip-cli udagroup,search -n<groupPattern> | -U<udaName> \
[XML and CSV output options] [login options]
```

Parameters

qip-cli -c **udagroup,search** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **udagroup**. Action is **search**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--name	-n	UDA group's name.

Long Option	Short Option	Allowable Values
--uda		**`--uda`** [[***`/component`***/]***`group`***/]***`name`***[=***`value`***

The following explains how UDAs are handled in the Web GUI compared to how they are handled when using them in a CLI.

Component

Optional. The component you specify must be defined under the infrastructure element for which this CLI is searching (for example, node, pool, or block).

Group

Optional. The attribute group you specify must be defined within the infrastructure or the search fails.

- When the GUI indicates **<NONE>** for a group, omitting a value for a group in the CLI means "not in a group". Example: **`--uda /node//state=NJ`**, or **`--uda state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute is not in a group. The first example specifies that the attribute is associated with a node. The second example does not specify what component the UDA is associated with.
- When the GUI specifies **<ANY>**, using an asterisk (**`*`**) to specify the group in the CLI means that the attribute must be in a group, regardless of the group's name. Example: **`--uda */state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute must belong to a group, but the group is not specified.
- When the GUI specifies **<IGNORE>**, for a group, using a dash (**`-`**) in the CLI means it does not matter if the UDA is in a group, or not. Example: **`--uda -/state=NJ`**. In this case, the attribute is "state" and the value is NJ. The attribute will be found regardless of whether it is in a group.

Name

This argument contains the name of the attribute, followed by an equals sign (=), followed by the value for which you are searching.

Command line input sample

```
qip-cli -c udagroup,search -n testuda -u admin1 -p password
   -o VitalQIP
```

zone,add

zone,add allows you to add a zone to a view within the the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c zone,add <login options> -I <DNS_VIEW_ZONE_REC>.xml
   [--batch] [--rejectfile <filename>.xml [-v] [-h|?]
```

Parameters

qip-cli -c **zone,add** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **zone**. Action is **add**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing zone records for the CLI.
--batch		Enables batch processing on multiple records of the same type in an input file.
--rejectfile	-r	File containing the records that failed processing. The file contains the error message details, and a list of any records not added.You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.

Command line example

```
qip-cli -c zone,add -u admin1 -p password -o VitalQIP
   -I zonerecs.xml --batch
```

You can use output from the **zone,get** or the **zone,search** command as a model for an input file for this command.

Global and view-specific zones

VitalQIP allows global zones to be shared into a view so a single DNS server can provide different levels of service according to the client's IP address. This type of zone always contains the same resource records as the same zone in the globally managed namespace.

Zones can be defined specifically to a DNS View. These zones are referred to as view-specific zones. **`zone,add`** only creates view-specific zones. View-specific zones provide clients with clients different DNS responses than they would receive in the global namespace. A view-specific zone can contain a different set of resource records than a globally managed zone of same name.

For additional information on global and view-specific zones, see the DNS Views chapter of the *VitalQIP Web Client User's Guide*.

zone,delete

zone,delete allows you to delete a zone contained within a DNS view.

Synopsis

```
qip-cli -c zone,delete <login options> -Z <Zone_name>
   -V <View_name> [-v] [-h|?]
```

Parameters

qip-cli -c **zone,delete** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **zone**.
		Action is **delete**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--zone	-Z	Name of the zone being deleted.
--view	-V	Name of the view that contains the zone.

Command line example

```
qip-cli -c zone,delete -u admin1 -p password -o VitalQIP
   --zone eastern --view humanres
```

Global and view-specific zones

VitalQIP allows global zones to be shared into a view so a single DNS server can provide different levels of service according to the client's IP address. This type of zone always contains the same resource records as the same zone in the globally managed namespace.

Zones can be defined specifically to a DNS View. These zones are referred to as view-specific zones. **Zone,delete** only applies to view-specific zones. View-specific zones provide clients with clients different DNS responses than they would receive in the global namespace. A view-specific zone can contain a different set of resource records than a globally managed zone of same name.

For additional information on global and view-specific zones, see the DNS Views chapter of the *VitalQIP Web Client User's Guide*.

zone,get

zone,get allows you to obtain an XML representation of the contents of a zone contained within a DNS view.

Synopsis

```
qip-cli -c zone,get <login options> (-d <domain>) | (-Z <zonename>
   -V <viewname> [--sharedintoview [--ipv4reversezone]])
   [-O <output_file>.xml]  [--outputxsl file1,file2] [-v] [-h|?]
```

Parameters

qip-cli -c **zone,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **zone**. Action is **get**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.
Specific Options		
--zone	-Z	Name of the zone being obtained.
--view	-V	Name of the view that contains the zone.
--domain	-d	Name of the domain that contains the view. You cannot use this option and the **--zone** and **--view** options in the same command. Using this option returns all zones in the domain.
--sharedintoview		Specifies that the zone is a global zone that has been shared into the specified zone.
--ipv4reversezone		Specifies that the zone is an IPv4 reverse zone.You must specify **--sharedintozone** to use this option.

Command line input sample

```
qip-cli -c zone,get -u admin1 -p passwd -o VitalQIP -Z EastCoast
   -V HumanRes
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<DNS_VIEW_ZONE_REC>
  <defaultTtl>86400</defaultTtl>
  <email>user@alcatel-lucent.com</email>
  <expireTime>604800</expireTime>
  <negativeCacheTtl>600</negativeCacheTtl>
  <refreshTime>21600</refreshTime>
  <retryTime>3600</retryTime>
  <zone>
    <ipv4ReverseZone>false</ipv4ReverseZone>
    <sharedIntoView>false</sharedIntoView>
    <viewName>HumanRes</viewName>
    <zoneName>EastCoast</zoneName>
  </zone>
</DNS_VIEW_ZONE_REC>
```

The output from this CLI, may contain the element `<dnsOptionList>`, which contains a list of DNS options.:

Each DNS option may take an argument. It may either be a single string, or an address match list. For the case of a single string, put it in the `<value>` element. For the case of an address match list, use the `<addrMatchValue>` element. An address match list uses the following syntax:

- A <zone> contains 0 or more entries. Any of the elements under <acl> may be repeated in any order. For example, an <acl> may contain several <ipv4_address> instances. Also, as is the nature of an access control list, the sequence is significant.
- One of the elements under <acl> is <other>. The values for <other> are the special keywords: **ANY**, **NONE**, **LOCALHOST** and **LOCALNETS**.
- To define an exclusion, prepend a "!" before the value. For example:**<ipv4_network>!192.168.100.0/24</ipv4_network>** and **<other>!LOCALHOST</other>**.

The names of all supported options are defined in *$QIPHOME/web/xml/dns_zone_options.xml* or can be obtained by running "info,dnsoptions" (p. 5-114)

zone,modify

zone,modify allows you to modify a zone in the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c zone,modify <login options> (-I <DNS_VIEW_ZONE_REC>
   [--replace <currentName>]) | (--sharedintoview
   -I <DNS_SHARED_ZONE_REC) [-v] [-h|?]
```

Parameters

qip-cli -c **zone,modify** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **zone**. Action is **modify**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing zone records as input for the CLI.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--replace		Zone's current name. You cannot specify this option and the **--shareintoview** option. **Note:** The --replace option is only necessary when a new name is specified by the input file.
--shareintoview		Share this zone into the current view. You cannot specify this option and the **--replace** option.

Command line input sample

```
qip-cli -c zone,modify -u admin1 -p admin1 -o Documentation
   -I zonemods.xml
```

You can use the output file from the **zone,get** cli as the template for the input file for this command. However, certain elements in the output are read-only and cannot be modified. You can use **qip-cli -c info,get -n DNS_SHARED_ZONE_REC** to see which elements are marked as read-only. Attempting to use **zone,modify** to set read-only values is silently ignored by **zone,modify**.

The zone's options can be inherited from the global zone by setting **<useGlobalOptions>true</useGlobalOptions>**. By setting this value to **false**, override values can be used. Attempting to provide a **dnsOptionList** when **useGlobalOptions** is true is silently ignored by **zone,modify**.

Global and view-specific zones

VitalQIP allows global zones to be shared into a view so a single DNS server can provide different levels of service according to the client's IP address. This type of zone always contains the same resource records as the same zone in the globally managed namespace.

Zones can be defined specifically to a DNS View. These zones are referred to as view-specific zones. View-specific zones provide clients with clients different DNS responses than they would receive in the global namespace. A view-specific zone can contain a different set of resource records than a globally managed zone of same name.

For additional information on global and view-specific zones, see the DNS Views chapter of the *VitalQIP Web Client User's Guide*.

zone,optionreport

zone,optionreport allows you to view the options associated with a zone.

Synopsis

```
qip-cli -c zone,optionreport <login options>
   [-O <output_file>.xml|.csv] [--outputxsl file1,file2]
   ([--format text|xml|csv[,columndelimiter=<char>]
   [,comment=<char>][,recorddelimiter=<char>]
   [,escapemode=doubled|backslash][,header]
   [,textqualifier=<char>][,utf8]]) -V Viewname [--sharedintoview]
   [-v] [-h|?]
```

Parameters

qip-cli -c **zone,optionreport** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **zone**. Action is **optionreport**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.

Long Option	Short Option	Allowable Values
--format		If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`text`** (default) **`csv`** xml If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`text`** (default) **`csv`** xml If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--zone	-Z	Name of the zone.
--view	-V	Name of the view that contains the zone.
--sharedintoview		Specifies that the zone is shared into the view (not view-specific).

Text

Command line input sample

```
qip-cli -c zone,optionreport -u admin1 -p passwd -o Documentation
   -Z EastCoast -V HumanRes --sharedintoview
```

Sample Output File

```
Option Name      Effective Value  View?  Zone?
===============  ===============  =====  =====
allow-notify     {!112.0.0.0;};          Y
check-integrity  yes;             Y
check-mx         warn;            Y
check-mx-cname   ignore;          Y
check-sibling    no;              Y
check-srv-cname  warn;            Y
check-wildcard   yes;             Y
forward          first;           Y
```

If a value is not defined, it is inherited from the view.

Four symbols indicate a given view/zone's option value:

> The value defined here has been overridden by a column to the right.

= Although this value is overridden by a column to the right, this column happens to contain the same value as that other column.

Y The effective value is defined by this view, zone, or server.

<blank> No option value is defined by this view, zone, or server.

CSV

Command line input sample

```
qip-cli -c zone,optionreport -u admin1 -p passwd -o Documentation
   -Z EastCoast -V HumanRes --format csv
```

Sample Output File

```
additional-from-auth,yes;,yes;
allow-query,{10.10.2.5;};,{10.10.2.5;};
min-refresh-time,600;,600;
```

XML

Command line input sample

```
qip-cli -c zone,optionreport -u admin1 -p passwd -o Documentation
   -Z EastCoast -V HumanRes --format xml
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<DnsViewOptionReport>
  <option>
    <effectiveValue>yes;</effectiveValue>
    <name>additional-from-auth</name>
    <viewValue>yes;</viewValue>
  </option>
  <option>
    <effectiveValue>{10.10.2.5;};</effectiveValue>
    <name>allow-query</name>
    <viewValue>{10.10.2.5;};</viewValue>
  </option>
  <option>
    <effectiveValue>600;</effectiveValue>
    <name>min-refresh-time</name>
    <viewValue>600;</viewValue>
  </option>
</DnsViewOptionReport>
```

zone,search

zone,search allows you to search for zones.

Synopsis

```
qip-cli -c zone,search <login options> [-O <output_file>.xml]
   [--outputxsl file1,file2] [-Z <namePattern>] -V <view>
   [--sharedintoview [--ipv4reversezone]]
   [--uda=[group/]name[=value]] [-v] [-h|?]
```

Parameters

qip-cli -c **zone,search** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **zone**. Action is **search**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.

Long Option	Short Option	Allowable Values
--uda	-U	**`--uda`** [[***`/component`***/]***`group`***/]***`name`***[=***`value`*** The following explains how UDAs are handled in the Web GUI compared to how they are handled when using them in a CLI. **Component** ***Optional.*** The component you specify must be defined under the infrastructure element for which this CLI is searching (for example, node, pool, or block). **Group** ***Optional***. The attribute group you specify must be defined within the infrastructure or the search fails. • When the GUI indicates **<NONE>** for a group, omitting a value for a group in the CLI means "not in a group". Example: **`--uda /node//state=NJ`**, or **`--uda state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute is not in a group. The first example specifies that the attribute is associated with a node. The second example does not specify what component the UDA is associated with. • When the GUI specifies **<ANY>**, using an asterisk (**`*`**) to specify the group in the CLI means that the attribute must be in a group, regardless of the group's name. Example: **`--uda */state=NJ`**. In this case, the attribute is "state" and the value is "NJ." The attribute must belong to a group, but the group is not specified. • When the GUI specifies **<IGNORE>**, for a group, using a dash (**`-`**) in the CLI means it does not matter if the UDA is in a group, or not. Example: **`--uda -/state=NJ`**. In this case, the attribute is "state" and the value is NJ. The attribute will be found regardless of whether it is in a group. **Name** This argument contains the name of the attribute, followed by an equals sign (=), followed by the value for which you are searching.
--version	-v	Displays the version of this CLI.

Long Option	Short Option	Allowable Values
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--zone	-Z	Name of the zone for which you are searching. You can use an asterisk (*) as a wildcard character to return a range of results. If you use the asterisk as the first or only character, you must put this value in either single or double quotes.
--view	-V	Name of the view that contains the zone.
--sharedintoview		A global zone shared into this view.
--ipv4reversezone		Zone is an IPv4 Reverse Zone. You can only use this option if you have specified the **`--shareintoview`** option.

Command line input sample

```
qip-cli -c zone,search -u admin1 -p passwd -o VitalQIP
   -Z EastCoast -V HumanRes
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<object-stream>
<DNS_VIEW_ZONE_REC>
  <defaultTtl>86400</defaultTtl>
  <email>user@alcatel-lucent.com</email>
  <expireTime>604800</expireTime>
  <negativeCacheTtl>600</negativeCacheTtl>
  <refreshTime>21600</refreshTime>
  <retryTime>3600</retryTime>
  <zone>
    <ipv4ReverseZone>false</ipv4ReverseZone>
    <sharedIntoView>false</sharedIntoView>
    <viewName>HumanRes</viewName>
    <zoneName>EastCoast</zoneName>
  </zone>
</DNS_VIEW_ZONE_REC>
</object-stream>
```

The output from this CLI, may contain the element `<dnsOptionList>`, which contains a list of DNS options.:

Each DNS option may take an argument. It may either be a single string, or an address match list. For the case of a single string, put it in the `<value>` element. For the case of an address match list, use the `<addrMatchValue>` element.

The names of all supported options are defined in *$QIPHOME/web/xml/dns_zone_options.xml* or can be obtained by running "info,dnsoptions" (p. 5-114)

zone,share

zone,share allows you to add a global zone to a view in the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c zone,share <login options> -I <inputfile>.xml [--batch]
   [--rejectfile <filename>.xml [-v] [-h|?]
```

Parameters

qip-cli -c **zone,share** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **zone**. Action is **share**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing zone records for the CLI.
--batch		Enables batch processing on multiple records of the same type in an input file.
--rejectfile	-r	File containing the records that failed processing. The file contains the error message details, and a list of any records not added.You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	

Command line example

```
qip-cli -c zone,share -u admin1 -p password -o VitalQIP
   -I zonerecs.xml --batch
```

You can use output from the **zone,get** or the **zone,search** command as a model for an input file for this command.

Global and view-specific zones

VitalQIP allows global zones to be shared into a view so a single DNS server can provide different levels of service according to the client's IP address. This type of zone always contains the same resource records as the same zone in the globally managed namespace.

Zones can be defined specifically to a DNS View. These zones are referred to as view-specific zones. **`zone,share`** shares an existing global zone. View-specific zones provide clients with clients different DNS responses than they would receive in the global namespace. A view-specific zone can contain a different set of resource records than a globally managed zone of same name.

For additional information on global and view-specific zones, see the DNS Views chapter of the *VitalQIP Web Client User's Guide*.

zone,unshare

zone,unshare allows you to remove a zone from a view in the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c zone,unshare <login options> -V <viewname>
   -Z <zonename> [--ipv4reversezone] [-v] [-h|?]
```

Parameters

qip-cli -c **zone,unshare** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **zone**.
		Action is **unshare**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--view	-V	Name of the view to be unshared.
--zone	-Z	Name of the zone in which the view resides
--ipv4reversezone		Specifies that the zone is an IPv4 reverse zone.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	

Command line example

```
qip-cli -c zone,unshare -u admin1 -p password -o VitalQIP
   -V <eastcoast> -Z <humanres>
```

You can use output from the **zone,get** or the **zone,search** command as a model for an input file for this command.

Global and view-specific zones

VitalQIP allows global zones to be shared into a view so a single DNS server can provide different levels of service according to the client's IP address. This type of zone always contains the same resource records as the same zone in the globally managed namespace.

Zones can be defined specifically to a DNS View. These zones are referred to as view-specific zones. zone,unshare removes a zone from a view. The global zone continues to exist.

View-specific zones provide clients with clients different DNS responses than they would receive in the global namespace. A view-specific zone can contain a different set of resource records than a globally managed zone of same name.

For additional information on global and view-specific zones, see the DNS Views chapter of the *VitalQIP Web Client User's Guide*.

zoneserver,add

zoneserver,add allows you to add a zone server, to the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c zoneserver,add <login options> -I <inputfile>.xml
   [--batch] [--rejectfile <filename>.xml [-v] [-h|?]
```

Parameters

qip-cli -c **zoneserver,add** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **zoneserver**. Action is **add**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing zone server, records for the CLI.
--batch		Enables batch processing on multiple records of the same type in an input file.
--rejectfile	-r	File containing the records that failed processing. The file contains the error message details, and a list of any records not added.You must specify a file name if you want to see this output. **Note:** The rejectfile name must be unique. If you specify an existing file name, the file is not overwritten and the CLI does not execute.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.

Command line example

```
qip-cli -c zoneserver,add -u admin1 -p password -o VitalQIP
   -I zoneserver.xml --batch
```

You can use output from the **zoneserver,get** or the **zoneserver,search** command as a model for an input file for this command.

zoneserver,delete

`zoneserver,delete` allows you to delete a zone server, contained within a DNS view.

Synopsis

```
qip-cli -c zoneserver,delete <login options> -n <zoneservername>
   -Z <zone_name> -V <View_name> [[--sharedintoview]
   [--ipv4reversezone]] [-v] [-h|?]
```

Parameters

qip-cli -c **`zoneserver,delete`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>.
		Object is **`zoneserver`**.
		Action is **`delete`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--name	-n	Name of the zone server.
--zone	-Z	Name of the zone that contains the zone server.
--view	-V	Name of the view that contains the zone.
--sharedintoview		Indicates that the zone server was shared into a view.
--ipv4reversezone		Indicates that the zone server is in an IPv4 reverse zone.

Command line example

```
qip-cli -c zoneserver,delete -u admin1 -p password -o VitalQIP
   --name hrserv1 --zone eastern --view humanres
```

zoneserver,get

zoneserver,get allows you to obtain an XML representation of the contents of a zone server contained within a DNS view.

Synopsis

```
qip-cli -c zoneserver,get <login options> -n <serverName>
   -Z <zonename> -V <viewname> [[--sharedintoview]
   [--ipv4reversezone]] [-O <output_file>.xml]
   [--outputxsl file1,file2] [-v] [-h|?]
```

Parameters

qip-cli -c **zoneserver,get** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **zoneserver**. Action is **get**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--name	-n	Name of the zone server.
--zone	-Z	Name of the zone that contains the zone server.
--view	-V	Name of the view that contains the zone.
--sharedintoview		Indicates that the zone server was shared into a view.
--ipv4reversezone		Indicates that the zone server is in an IPv4 reverse zone.

Command line input sample

```
qip-cli -c zoneserver,get -u admin -p passwd -o VitalQIP
   -Z EastCoast -V HumanRes -n corporate.global.com
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<DNS_ZONE_SERVER_REC>
  <name>Corporate.global.com</name>
  <zone>
    <ipv4ReverseZone>false</ipv4ReverseZone>
    <sharedIntoView>false</sharedIntoView>
    <viewName>HumanRes</viewName>
    <zoneName>EastCoast</zoneName>
  </zone>
</DNS_ZONE_SERVER_REC>
```

The output from this CLI, may contain the element `<dnsOptionList>`, which contains a list of DNS options.:

Each DNS option may take an argument. It may either be a single string, or an address match list. For the case of a single string, put it in the `<value>` element. For the case of an address match list, use the `<addrMatchValue>` element. An address match list uses the following syntax:

- A <zoneserver> contains 0 or more entries. Any of the elements under <acl> may be repeated in any order. For example, an <acl> may contain several <ipv4_address> instances. Also, as is the nature of an access control list, the sequence is significant.
- One of the elements under <acl> is <other>. The values for <other> are the special keywords: **ANY**, **NONE**, **LOCALHOST** and **LOCALNETS**.
- To define an exclusion, prepend a "!" before the value. For example:**<ipv4_network>!192.168.100.0/24</ipv4_network>** and **<other>!LOCALHOST</other>**.

The names of all supported options are defined in *$QIPHOME/web/xml/dns_zone_options.xml* or can be obtained by running "info,dnsoptions" (p. 5-114)

zoneserver,modify

zoneserver,modify allows you to modify a zone server in the VitalQIP® (QIP) database.

Synopsis

```
qip-cli -c zoneserver,modify <login options>
   [--replace <currentname>] -I <inputfile>.xml [-v] [-h|?]
```

Parameters

qip-cli -c **zoneserver,modify** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **zoneserver**. Action is **modify**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--inputfile	-I	Name of the file containing zone server records as input for the CLI.
--version	-v	Displays the version of this CLI.
--help	-h -?	Displays syntax and Options for this CLI.
Specific Options		
--replace		Zone server's current name. **Note:** The --replace option is only necessary when a new name is specified by the input file.

Command line input sample

```
qip-cli -c zoneserver,modify -u admin1 -p admin1 -o Documentation
   -I zoneservermods.xml
```

You can use the output file from the **zoneserver,get** cli as the template for the input file for this command.

zoneserver,optionreport

zoneserver,optionreport allows you to view the options associated with a zone server.

Synopsis

```
qip-cli -c zoneserver,optionreport <login options>
   [-O <output_file>.xml|.csv] [--outputxsl file1,file2]
   ([--format text|xml|csv[,columndelimiter=<char>]
   [,comment=<char>][,recorddelimiter=<char>]
   [,escapemode=doubled|backslash][,header]
   [,textqualifier=<char>][,utf8]]) -V <Viewname> -Z <Zonename>
   -n <zoneservername> [--sharedintoview [--ipv4reversezone]] [-v]
   [-h|?]
```

Parameters

qip-cli -c **zoneserver,optionreport** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **zoneserver**. Action is **optionreport**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.

Long Option	Short Option	Allowable Values
--format		If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`text`** (default) **`csv`** xml If you specify the appropriate suffix when you specify a filename, you do not need to use the **`--format`** option. You are only required to use this option if you do not specify a suffix on your filename, or you want to use a value other than the default provided for any suboptions. Allowable values are: **`text`** (default) **`csv`** xml If you specify **`csv`**, there are a variety of suboptions available. See "CSV input and output" (p. 5-8) for a list of each option and its definition. **Note:** If you change a default value when creating an output file, and you later use that file as input for another CLI, you must specify the same csv suboptions.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--zone	-Z	Name of the zone.
--view	-V	Name of the view that contains the zone.
--sharedintoview		Zone server is shared into this view. You cannot use this option and **`--ipv4reversezone`** option.

Text

Command line input sample

```
qip-cli -c zoneserver,optionreport -u admin1 -p passwd
   -o Documentation -Z EastCoast -V HumanRes
   -n corporate.global.com --format text
```

Sample Output File

```
Option Name           Effective Value  View?  Zone?  Server?
====================  ===============  =====  =====  =======
additional-from-auth  yes;             Y
allow-notify          {10.10.2.3;};                  Y
allow-query           {10.10.2.5;};    Y
check-integrity       yes;                           Y
min-refresh-time      600;             Y
notify                yes;                           Y
```

If a value is not defined, it is inherited from the zone, or if not specified there, from the view.

Four symbols indicate a given view/zone/server's option value:

> The value defined here has been overridden by a column to the right.

= Although this value is overridden by a column to the right, this column happens to contain the same value as that other column.

Y The effective value is defined by this view, zone, or server.space

<blank> No option value is defined by this view, zone, or server.

CSV

Command line input sample

```
qip-cli -c zoneserver,optionreport -u admin1 -p passwd
   -o Documentation -Z EastCoast -V HumanRes
   -n corporate.global.com --format csv
```

Sample Output File

```
additional-from-auth,yes;,yes;
allow-notify,{10.10.2.3;};,,,{10.10.2.3;};
allow-query,{10.10.2.5;};,{10.10.2.5;};
check-integrity,yes;,,,yes;
min-refresh-time,600;,600;
notify,yes;,,,yes;
```

XML

Command line input sample

```
qip-cli -c zoneserver,optionreport -u admin1 -p passwd
   -o Documentation -Z EastCoast -V HumanRes
   -n corporate.global.com --format xml
```

Sample Output File


```
<?xml version="1.0" encoding="UTF-8" ?>
<DnsViewOptionReport>
  <option>
    <effectiveValue>yes;</effectiveValue>
    <name>additional-from-auth</name>
    <viewValue>yes;</viewValue>
  </option>
  <option>
    <effectiveValue>{10.10.2.3;};</effectiveValue>
    <name>allow-notify</name>
    <serverValue>{10.10.2.3;};</serverValue>
  </option>
  <option>
    <effectiveValue>{10.10.2.5;};</effectiveValue>
    <name>allow-query</name>
    <viewValue>{10.10.2.5;};</viewValue>
  </option>
  <option>
    <effectiveValue>yes;</effectiveValue>
    <name>check-integrity</name>
    <serverValue>yes;</serverValue>
  </option>
  <option>
    <effectiveValue>600;</effectiveValue>
    <name>min-refresh-time</name>
    <viewValue>600;</viewValue>
  </option>
  <option>
    <effectiveValue>yes;</effectiveValue>
    <name>notify</name>
    <serverValue>yes;</serverValue>
  </option>
</DnsViewOptionReport>
```

zoneserver,search

`zoneserver,search` allows you to search for zone servers.

Synopsis

```
qip-cli -c zoneserver,search <login options>
   [-O <output_file>.xml] [--outputxsl file1,file2]
   [--sharedintoview [--ipv4reversezone]]
   [--uda=[group/]name[=value]] [-v] [-h|?]
```

Parameters

qip-cli -c **`zoneserver,search`** recognizes the following parameters:

Long Option	Short Option	Allowable Values
--command	-c	Syntax is <object>,<action>. Object is **`zoneserver`**. Action is **`search`**.
Login Options		
--username	-u	Administrator name.
--password	-p	Administrator password.
--organization	-o	Working organization.
Base Options		
--outputfile	-O	Name of the file containing output from the CLI.
--outputxsl		Name of the file that specifies the XML format of the output file. Not applicable if the output file is a csv file.
--version	-v	Displays the version of this CLI.
--help	-h	Displays syntax and Options for this CLI.
	-?	
Specific Options		
--zone	-Z	Name of the zone for which you are searching. You can use an asterisk (*) as a wildcard character to return a range of results. If you use the asterisk as the first or only character, you must put this value in either single or double quotes.
--view	-V	Name of the view that contains the zone server.
--sharedintoview		Share a global zone server into this view. You cannot use both this option and **`--ipv4reversezone`** option.
--ipv4reversezone		Zone is an IPv4 Reverse Zone. You cannot use both this option and **`--shareintoview`** option.

Command line input sample

```
qip-cli -c zoneserver,search -u admin1 -p passwd -o VitalQIP
   -Z EastCoast -V HumanRes
```

Sample Output File

```
<?xml version="1.0" encoding="UTF-8" ?>
<object-stream>
<DNS_ZONE_SERVER_REC>
  <name>corporate.global</name>
  <zone>
    <ipv4ReverseZone>false</ipv4ReverseZone>
    <sharedIntoView>false</sharedIntoView>
    <viewName>HumanRes</viewName>
    <zoneName>EastCoast</zoneName>
  </zone>
</DNS_ZONE_SERVER_REC>
</object-stream>
```

The output from this CLI, may contain the element `<dnsOptionList>`, which contains a list of DNS options.:

Each DNS option may take an argument. It may either be a single string, or an address match list. For the case of a single string, put it in the `<value>` element. For the case of an address match list, use the `<addrMatchValue>` element.

The names of all supported options are defined in *$QIPHOME/web/xml/dns_zone_options.xml* or can be obtained by running "info,dnsoptions" (p. 5-114)

Index

A

A record
host name, 3-4

A records, 1-274

About CLIs, 5-2

active lease file, 1-152
DHCP, 1-152

active leases, 1-2, 1-8, 1-45
converting, 1-165, 1-170

database, 1-48

DHCP, 1-2, 1-8

retrieving, 1-8

address fields, 1-200
searching, 1-200

Address Management and Address Allocation, 5-12

address ranges, 1-63
exporting, 4-75

importing, 4-3

Network Profile, 1-63

addressrange,add CLI, 5-37

addressrange,delete CLI, 5-39

addressrange,get CLI, 5-40

addressrange,modify CLI, 5-43

administrative roles, 1-65
adding or modifying, 1-207

deleting, 1-42

managed list report, 1-194

report, 1-180

reports, 1-180, 1-194

administrator
managed ranges report, 1-190

Administrator Audit Report, 1-179

administrator passwords, 1-31

administrators
changing the password, 1-31

copying and deleting, 1-12

domain folders, 1-83

domain folders policy option, 1-83

list from the database, 1-190

managed ranges, 1-190

alias name, 1-33

aliases, 1-33

amseedblock,add CLI, 5-45

amseedblock,delete CLI, 5-47

amseedblock,get CLI, 5-48

applications, 1-38
list from the database, 1-66

Audit Manager, 1-36, 1-288, 2-2, 2-4

audit table
clear, 1-34

audit tables
clearing, 1-34

Automatic Reclaim Report, 1-237

B

BIND 4.x, 4-78

BIND 8.x, xx, 3-5

block,addtopool CLI, 5-50

block,allocate CLI, 5-52

block,deletefrompool CLI, 5-55

block,expand CLI, 5-56

block,free CLI, 5-57

block,freepending CLI, 5-58

block,get CLI, 5-60

block,merge CLI, 5-63

block,modify CLI, 5-64

block,move CLI, 5-66

block,renumber CLI, 5-67, 5-68

block,return CLI, 5-70

block,search CLI, 5-71

block,split CLI, 5-78

Bootp server, 1-30
list from the database, 1-67

Bootp servers
list from the database, 1-67

Bootptab configuration files
generating, 1-30

Bootptab file, 1-30, 3-21
generating, 1-30

Bootptab files, 1-30
exporting, 3-19

generating, 1-30

C

CalculateLAMSize function, 1-289

CalculateQIPSize function, 1-289

CheckDatabaseLogin function, 1-289

CheckDBProcesses function, 1-289

CheckSybaseDatabase function, 1-289

CheckSybaseDevice function, 1-289

CheckUserExists function, 1-290

CheckVersionFromData function, 1-289

CheckVersionFromDatabase function, 1-289

CheckVersionFromScript function, 1-289
clear audit data, 1-34
ClearAdmin function, 1-290
ClearData function, 1-290
client class
vendor and user classes, 1-68, 1-209
configuration and data files, 1-50
generating, 1-52
SOA record, 1-52
configuration files
DNS, 1-52
contact information, 1-69
adding or modifying, 1-211, 1-223, 1-240
exporting, 4-77
importing, 4-7
contacts
contact ID, 1-39, 1-69, 1-243
exporting, 4-77
importing, 4-7
contract information, 1-211, 1-243
CreateAccess function, 1-290
CreateSybaseDatabase function, 1-290
CreateSybaseDevice function, 1-290

D

database, 1-34, 1-36
DHCP server, 1-8, 1-48, 1-158
DNS files, 3-2
dropping tables and indexes, 1-288
exporting, 3-2
importing, xx, 2-4, 3-2
initializing, 1-36
searching, 1-200
utility, 1-288
database tables
clearing, 1-34
DECNet
addresses, 1-73, 1-284
dual protocol, 1-232
DECNet address
releasing, 1-282, 1-284
DECNet addresses, 1-73, 1-106, 1-284, 1-287
searching for, 1-201
DHCP
active lease file, 1-152
active leases, 1-8, 1-48, 1-152
adding and deleting MAC addresses, 1-148
dumping active lease files, 1-45
dumping the active lease file, 1-45
generating configuration files, 1-46, 1-158
host name IP address synchronization, 1-48
MAC pools, 1-148
scope, 1-197
scopes, 1-197
subnet list, 1-76
templates, 1-130, 1-132, 1-209, 1-277
upgrading from 3.1 to 4.0, 1-165, 1-170
DHCP Policy Template, 1-129, 1-209, 1-231, 1-244
DHCP scopes, 1-158, 1-197
DHCP server subnets, 1-76
DHCP servers, 1-170
database, 1-8, 1-48, 1-158
exporting, 4-88
importing, 3-2, 4-2, 4-15, 4-43
list from the database, 1-77
MAC address pools, 1-148
MAC pools, 1-148
Microsoft, 1-158
reports, 1-183
DHCP templates
client class, 1-68, 1-209
creating, 1-277
DNS
configuration and data files, 1-52
domain extensions, 4-97
exporting files, 3-2
exporting resource records, 4-78
exporting zone options, 4-80
generating configuration and data files, 1-50
importing, 3-2, 3-18
importing resource records, 3-17, 4-11
importing zone options, 4-15
non-managed servers, 1-99, 1-220
resource records, 3-8, 3-17, 4-11
reverse zones, 1-116
Update Service, 1-56
updating the SOA, 1-52
zone extensions, 1-141, 1-256
zone options, 1-250, 4-15
zones, 1-250
DNS files
exporting, 3-2
importing, 3-2
DNS servers, 1-80
exporting, 4-88

importing, 3-2, 3-13, 3-18, 4-2, 4-15, 4-42
list from the database, 1-80
non-managed, 1-220
non-managed profile, 1-99
primary, 1-116, 1-263
resource records, 1-274
secondary, 1-119, 1-263
DNS Update Service, 1-56
domain
adding, modifying, or deleting, 1-54
domain extensions, 1-274
folders, 1-54
grouping into folders, 1-54
Domain Controller, 1-266
domain extensions, 4-97
domain folders, 1-54, 1-83
Domain Profile, 1-144, 1-259, 4-97
domains, 1-82
domain folders, 1-54, 1-83, 1-216
Domain Profile, 1-144, 1-259, 4-97
exporting, xx, 4-78, 4-80
folders, 1-83, 1-216
importing, 3-2, 3-13, 4-2, 4-72
resource records, 1-202, 1-213, 3-10
user-defined fields, 1-202
zone extensions, 1-256
DropIndex function, 1-290
DropSP function, 1-290
DropTable function, 1-291
DropTrigger function, 1-291

E

EditTextFile function, 1-291
encryption, 1-178
enteraddrrange, 4-3
enteralias, 3-2, 3-17, 3-18, 4-5
entercontact, 4-7
enterdnsobj, 4-9
enterdnsrr, 3-2, 3-17, 3-18, 4-11
enterdomain, 3-2, 3-13, 3-18, 4-2, 4-15
enterlocalobj, 4-22
enterlocation, 4-24
entermanufacturer, 4-26
enternetwork, 3-2, 3-26
enterorganization, 4-30
enterospf, 4-32
enterreversezone, 4-34
enterserver, 3-2, 4-2, 4-15
entersimpleobj, 3-2
entersubnet, 3-2
entersubnetorg “ “D2HBentersubneto49, 4-55
enterudf, 4-59
enteruser, 4-61
enterusergrp, 4-63
enterzoneext, 4-64
EstimateRequiredSpace function, 1-291
exportaddrrange, 4-75
exportcontact, 4-77
exportdnsrr, 4-78
exportdomain, 4-80
exporting
Audit Manager data, 2-2
Bootptab files, 3-19
contacts, 4-77
DNS files, 3-2
DNS zone options, 4-80
domains, xx, 4-78, 4-80
locations, 4-82
managed address ranges, 4-75
manufacturers, 4-83
networks, 4-84
objects, 3-19, 4-89
organizations, 4-85, 4-93
OSPF areas, 4-86
OSPFs, 4-86
resource records, 4-78
reverse zones, 4-87
servers, 4-88
subnet organizations, 4-93
Subnet Profile, 4-91
subnets, 4-91
user groups, 4-96
user-defined fields, 4-94
users, 4-95
VitalQIP data, 2-2
zone extensions, 4-97
exporting Bootptab files, 3-19
exporting contacts, 4-77
exportlocation, 4-82
exportmanufacturer, 4-83
exportnetwork, 4-84
exportorganization, 4-85
exportospf, 4-86
exportreversezone, 4-87
exportserver, 4-88
exportsimpleobj, 4-89
exportsimpleobj, 4-89
exportsubnet, 4-91
exportsubnetorg, 4-93
exportudf, 4-94
exportuser, 4-95
exportusergrp, 4-96
exportzoneext, 4-97

F

folders, 1-54, 1-83

free subnets, 1-84, 1-187
report, 1-187

G GAP objects, 1-123, 1-239
Generating Bootptab configuration files, 1-30
generating Bootptab files, 1-30
GetAndSetSybaseDBOption function, 1-291
GetDatabaseSize function, 1-291
Global Policies, 1-43
deleting, 1-43
objects, 4-23
user-defined fields, 1-43

H HINFO records, 1-201, 3-4
importing, 4-9
host name, 1-33, 1-49
A record, 3-4
host names
synchronization with IP addresses, 1-48
hub ports, 1-87, 1-287
releasing, 1-285

I importing, 3-1
Bootptab files, 3-26
contact information, 4-7
contacts, 4-7
database information, 2-4
DHCP servers, 4-43
DNS files, 3-2, 3-18
DNS servers, 3-2, 3-13, 3-18, 4-2, 4-42
domains, 3-2, 3-13, 3-18, 4-2, 4-15, 4-72
locations, 4-24
MAC addresses, 4-2
managed address ranges, 4-3
manufacturers, 4-26
networks, 3-2, 3-26
object alias information, 3-2, 3-17, 3-18, 4-5
object information in local format, 4-22
objects, 3-2, 3-8, 3-22, 4-5, 4-66
organizations, 1-267, 4-30, 4-55
OSPF areas, 4-32
OSPFs, 4-2, 4-32
reource records for objects, 4-9
resource records, 3-2, 3-8, 3-17, 4-11
resource records for objects, 3-2, 3-17, 3-18, 4-11
reverse zones, 4-34
rules file, 1-49
ser-defined fields, 4-59
servers, 3-2, 3-13, 4-2, 4-15, 4-42, 4-88
subnet domains, 4-72
subnet organizations, 4-55
Subnet Profile, 4-52
subnets, 3-2
user groups, 4-63
users, 4-61
using the rules file, 3-8
zone extensions, 4-64
zone options, 4-15
info,get CLI, 5-114, 5-115
Inquiry Report, 1-188
IP addresses, 1-8, 1-286
aliases, 3-17
and host names, 1-33
deleting objects, 1-38
DHCP scope, 1-158, 1-197
free, 1-89, 1-286
Planned Use addresses, 1-2
reclaiming, 1-171
reserved, 1-32, 1-38
retreiving available, 1-286
subnet addresses, 1-89, 1-120
subnets, 1-89, 1-120
synchronization with host names, 1-48
unallocated, 1-38, 1-286
unallocating, 1-38
unlocking, 1-287
unused, 1-32, 1-171
users, 1-140
IP addresses and aliases, 1-33
IP addresses and host names, 1-33

J job,cancel CLI, 5-119
job,delete CLI, 5-120
job,get CLI, 5-121
job,search CLI, 5-124

L list,search CLI, 5-128
Local Host configuration file generating, 1-151
Local Host Server configuration files, 1-151
Local Host server, 1-151
Local host server, 1-90
local host servers
list from the database, 1-90
local objects
importing, 4-22
locations
adding and modifying, 1-217, 1-223, 1-240
adding or modifying, 1-217

exporting, 4-82
importing, 4-24
list from the database, 1-91
location ID, 1-248

M MAC address, 4-2
adding or modifying, 1-148
querying the MAC pool, 1-148
MAC address pools, 1-148
importing, 4-2
MAC addresses
adding and deleting from MAC pools, 1-148
MAC pools, 1-272
maintainer,add CLI, 5-98, 5-112, 5-131
maintainer,delete CLI, 5-99, 5-133, 5-146, 5-147, 5-182, 5-290, 5-309
maintainer,get CLI, 5-100, 5-134, 5-148, 5-183, 5-292, 5-310
maintainer,modify CLI, 5-105, 5-137, 5-150, 5-185, 5-294, 5-312
maintainer,search CLI, 5-103, 5-106, 5-109, 5-139, 5-153, 5-296, 5-300, 5-304, 5-306, 5-313, 5-317
managed list
add, 1-26
managed ranges
report, 1-190
manufacturers
exporting, 4-83
importing, 4-26
list from the database, 1-93
model lists, 1-94
Microsoft DHCP, 1-158

N network address ranges, 1-63
networks
exporting, 4-84
free subnets, 1-187
importing, 3-2
importing managed address ranges, 4-3
list from the database, 1-97
shared networks, 1-128
NIS
generating files, 1-150
NIS servers, 1-98, 1-150
generating files, 1-150
importing, 4-43
list by organization, 1-98
list from the database, 1-98
node,add CLI, 5-157
node,delete CLI, 5-160
node,get CLI, 5-162
node,link CLI, 5-166
node,modify CLI, 5-167
node,move CLI, 5-169
node,search CLI, 5-171
node,unlink CLI, 5-179
non-managed DNS servers, 1-220

O object
object class, 1-110
object aliases
importing, 3-2, 3-17, 4-5
Object History Report, 1-180
Object Profile, 1-103, 1-197, 1-223
objects
adding and modifying, 1-223
adding or modifying, 1-223, 3-19
aliases, 1-33
by manufacturer, 1-93
by user, 1-140
deleting, 1-38, 1-281
exporting, 3-19, 4-89
free, 1-110
GAP objects, 1-123, 1-239
host name, 1-33, 1-49
importing, 3-2, 3-8, 3-22
importing local, 4-22
list from the database, 1-101
moves, 1-154, 1-156
moving, 1-154
object classes, 1-93, 1-94, 1-123, 1-239, 3-8
object names, 1-110, 1-158
Object Profile, 1-103, 1-197
object range, 1-38
policies, 4-23
reclaiming, 1-171
report, 1-188, 1-191
reports, 1-181, 1-188, 1-191
resource records, 1-78, 1-213, 1-276, 3-8
scheduled moves, 1-125
searching, 1-200
statistics, 1-171
status, 1-32, 1-49
subnet address, 1-120
tombstoned, 1-79
unused, 1-32
user-defined fields, 1-246
Oracle, 1-36, 1-40, 1-288, 2-2, 4-2, 4-72
OracleReCompile function, 1-291
OracleUpdateStatistics function, 1-292
organizations, 1-3, 1-6, 1-67

Bootp servers by organization, 1-67
domain folders, 1-83
exporting, 4-85, 4-93
generating DNS files, 1-56
importing, 1-267, 4-30, 4-55
list by user name, 1-111
NIS servers by organization, 1-98
reverse zones, 1-117
subnet organizations, 1-39, 1-85, 1-121, 1-238, 1-267, 4-55, 4-93
user-defined fields, 1-135, 1-246
OSPF, 1-113, 1-233
exporting, 4-86
importing, 4-2, 4-32
list from the database, 1-112, 1-113
objects by OSPF, 1-191
OSPF Profile, 1-113, 1-233
OSPF areas, 4-86
OSPF Profile, 1-113, 1-233

P

passwords, 1-3, 1-5
administrators, 1-31
changing, 1-31
encrypting, 1-35
encryption, 1-35
qip.pcy file, 1-3
qipman, 1-35
Planned Use addresses, 1-2, 1-287
policies
adding or modifying, 1-234
deleting, 1-43
deleting records, 1-281
policies file, 1-3, 1-5, 2-4
policy, 1-115
adding or modifying, 1-234
administrators, 1-83
deleting, 1-40, 1-43
hostname, 4-23, 4-48
qip.pcy file, 1-5
reclaim policy, 1-172
tombstone, 1-281
policy information, 1-115
adding or modifying, 1-234
pool,add CLI, 5-186
pool,delete CLI, 5-188
pool,get CLI, 5-189
pool,merge CLI, 5-193
pool,modify CLI, 5-194
pool,search CLI, 5-196

Q

qip.pcy, 1-3, 1-5, 2-4
qip.pcy file
passwords, 1-3
qip-active, 1-8
qip-admin, 1-12
add managed list, 1-26
qip-altersubnet, 1-29
qip-bootpgen, 1-30
qip-bootptabcsv, 3-20
qipbulkload, 3-2, 4-5, 4-66
qip-changepassword, 1-31
qip-check, 1-32
qip-checkobjname, 1-33
qip-clear, 1-34
qip-crypt, 1-35
qip-dbinit, 1-36
qip-del, 1-38
qip-delacltemplate, 1-41
qip-deladminrole, 1-42
qip-delpolicy, 1-43
qip-dhcpdebuglog, 1-44
qip-dhcpdump, 1-45
qip-dhcpgen, 1-46
qip-dhcpsync, 1-48
qip-dnscsv, 3-2
qip-dnsgen, 1-50
qip-dnsupdate, 1-52
qip-export, 2-2
qip-folder, 1-54
qip-genddnsconfs, 1-56
qip-getacltemplate, 1-57
qip-getactiveobjectlst, 1-59
qip-getaddrrangelst, 1-63
qip-getadminrole, 1-65
qip-getapplst, 1-66
qip-getbtpsvrlst, 1-67
qip-getclientclass, 1-68
qip-getcontactlst, 1-39, 1-69, 1-211, 1-243
qip-getdebuglevel, 1-72
qip-getdecnetaddr, 1-73, 1-284
qip-getdhcpscopes, 1-74
qip-getdhcpsubnetlst, 1-76
qip-getdhcpsvrlst, 1-77
qip-getdnsrr, 1-78
qip-getdnssvrlst, 1-80
qip-getdomnlst, 1-82
qip-getfolderlst, 1-83
qip-getfreesubnetlst, 1-84
qip-gethublst, 1-86
qip-gethubport, 1-87
qip-gethubslotlst, 1-88
qip-getipaddr, 1-89
qip-getlocalsvrlst, 1-90
qip-getloclst, 1-91, 1-243
qip-getmaclst, 1-93
qip-getmacmodellst, 1-94
qip-getmacpools, 1-96
qip-getnetlst, 1-97

qip-getnissvrlst, 1-98
qip-getnmdnsserver, 1-99
qip-getobjectlst, 1-101
qip-getobjectprof, 1-103
qip-getobjname, 1-110
qip-getorganization, 1-111
qip-getospflst, 1-112
qip-getospfprof, 1-113
qip-getpolicy, 1-115
qip-getprimdnssvrlst, 1-116
qip-getrevzonelst, 1-117
qip-getrtrlst, 1-118
qip-getsecdnssvrlst, 1-119
qip-getsnaddr, 1-120
qip-getsnorglst, 1-121
qip-getsnorgprof, 1-122
qip-getsubnetlst, 1-124
qip-getsubnetprof, 1-126
qip-gettemplate, 1-130
qip-gettemplst, 1-132
qip-gettimesvrlst, 1-133
qip-getudfnamelst, 1-137
qip-getuser, 1-138
qip-getuseraddrlst, 1-140
qip-getzoneext, 1-141
qip-getzoneprof, 1-144
qip-globalmacpool, 1-148
qip-hndbgen, 1-150
qip-hostgen, 1-151
qip-import, 2-4
qip-leasefilegen, 1-152
qiploadsndomn, 4-72
qip-mcancel, 1-154
qip-move, 1-156
qip-msextract, 1-158
qip-namingpolicy, 1-159
qip-objectclass, 1-162
qip-postallocation, 1-165
qip-qdhcplease, 1-170
qip-reclaim, 1-171
qip-report, 1-174
qip-rot13, 1-178
qip-rptadminaudit, 1-179
qip-rptadminrole, 1-180
qip-rptaudithistory, 1-181
qip-rptdhcp, 1-183
qip-rptfreesubnet, 1-187
qip-rptinquire, 1-188
qip-rptmanaged, 1-190
qip-rptobjectlst, 1-191
qip-rptrole, 1-194
qip-rptzonerr, 1-195
qip-scope, 1-197
qip-search, 1-200
qip-searchacltemplates, 1-205
qip-setadminrole, 1-207
qip-setclientclass, 1-68, 1-209
qip-setcontact, 1-211
qip-setdnsrr, 1-79, 1-213
qip-setdomainfolder, 1-216
qip-setlocation, 1-217
qip-setmacpools, 1-219
qip-setnmdnsserver, 1-220
qip-setobject, 1-223, 3-19
qip-setospfprof, 1-233
qip-setpolicy, 1-234
qip-setreclaimschedule, 1-236
qip-setsnorgprof, 1-238
qip-setsubnet, 1-240
qip-setudf, 1-246
qip-setuser, 1-248
qip-setzonednsoptions, 1-250
qip-setzoneext, 1-256
qip-setzoneprof, 1-259
qip-sitegen, 1-266
qip-siteimport, 1-267
qip-splitmergeenum, 1-269
qip-splitrevzone, 1-271
qip-subnetmacpool, 1-272
qip-syncexternal, 1-274
qip-template, 1-277
qip-tombstonepurge, 1-281
qip-ungetdecnetaddr, 1-282, 1-284
qip-ungethubport, 1-285
qip-ungetipaddr, 1-286
qip-unlock, 1-287
qip-util, 1-288
qsi-import, 4-66, 4-72

R

RebuildIndexes function, 1-292
reclaim
  IP addresses, 1-171
registry,get CLI, 5-202
registry,modify CLI, 5-205
RemoveInvalidChars function, 1-292
RemoveSpaceFields function, 1-292
reource records
  domains, 1-202, 3-10
  importing, 3-2, 4-11
reousrce records
  reports, 1-195
report
  object list, 1-191
Reports
  administrator audit, 1-179
  Automatic Reclaim, 1-237
reports, 1-174
  Adminisitrative Role, 1-180
  administrative roles, 1-180, 1-194

administrative roles and managed lists, 1-194
Administrator Audit, 1-179
administrator audit, 1-179
administrators' managed lists, 1-190, 1-194
DHCP Server Profile, 1-183
DHCP server profile, 1-183
free subnet, 1-187
free subnets, 1-187
Inquiry Report, 1-188
managed range, 1-190
Object History, 1-181
object history, 1-180, 1-181
object inquiry, 1-188
object list by owner, 1-191
objects, 1-188, 1-191
qip-report, 1-8, 1-174
resource records, 1-195
scheduled reclaims, 1-171
zone resouce records, 1-195
reserved addresses, 1-32
resource record
CNAME, 1-108, 1-195, 1-231, 3-4, 3-8, 4-9, 4-12
SRV, 1-202
resource records, 1-78, 1-188, 1-195, 1-200, 1-202, 1-274, 3-2, 4-11, 4-78
adding and deleting, 1-213
deleting, 1-281
DNS, 3-8, 3-17, 4-11
domain, 1-78
exporting, 4-78
HINFO, 1-201, 3-4, 4-9
importing, 3-2, 3-17, 3-18, 4-9, 4-11
objects, 1-78
reports, 1-195
reverse zones, 1-78
rules file, 1-49
searching, 1-188, 1-200, 1-202
reverse zone
resource records, 1-213
zone extension, 1-256
Reverse Zone Profile, 1-144, 1-259
reverse zones, 1-116
DNS resource records, 1-78, 1-213
DNS servers, 1-116
exporting, 4-87
importing, 4-34
list from the database, 1-117
profile, 1-144, 1-259
PTR record, 3-4
Reverse Zone Profile, 1-144, 1-259
splitting, 1-271
user-defined fields, 1-40, 1-136, 1-247, 1-254
zone extensions, 1-141, 1-251, 1-256
routers, 1-46, 1-108, 1-118
default routers, 1-108, 1-129, 1-231, 1-244
rule,add CLI, 5-89, 5-207
rule,copy CLI, 5-91, 5-209
rule,delete CLI, 5-92, 5-210
rule,get CLI, 5-93, 5-211
rule,modify CLI, 5-95, 5-214
rule,search CLI, 5-96, 5-216
rules file, 3-8

S

scheduled moves, 1-125
canceling, 1-154
cancelling, 1-154
scheduled reclaims, 1-171
searching
for contacts, 1-69
for locations, 1-91
for objects, 1-188
qip-search, 1-200
searching VitalQIP, 1-200
SearchReplace function, 1-292
secondary servers
refresh time, 1-253
serarching
searching the search_token list, 3-11
serial number
in SOA record, 1-52
of object, 1-106, 1-228
servers
DHCP, 1-170
DNS servers, 1-116, 3-13
exporting, 4-88
importing, 3-2, 3-13, 4-2, 4-15, 4-42, 4-88
local host servers, 1-90
NIS, 1-98, 1-150
non-managed DNS, 1-99
Oracle database servers, 1-36
primary DNS servers, 1-116, 1-119
time servers, 1-133
upgrading, 1-170
SetSybaseConfigure function, 1-292
site files, 1-267
sites
creating, 1-266
subnet organizations, 1-266
SOA record, 1-52, 3-13, 3-15
SOA records
updating, 1-52
subnet

subnet addresses, 1-29, 1-49, 1-77, 1-120, 1-192, 1-240

Subnet Organization Profile, 1-122

subnet organizations
deleting, 1-39
exporting, 4-93
importing, 4-55
list from the database, 1-121
profile, 1-122, 1-238
profiles, 1-122, 1-238
Windows 2000, 1-266, 1-267
Windows 2000 sites, 1-267

Subnet Profile, 1-126, 1-240
exporting, 4-91
importing, 4-52

subnet profile
user-defined fields, 1-135, 1-247

subnet,add CLI, 5-219

subnet,delete CLI, 5-222

subnet,get CLI, 5-223

subnet,merge CLI, 5-227

subnet,modify CLI, 5-228

subnet,renumber CLI, 5-230

subnet,search CLI, 5-232

subnet,split CLI, 5-239

subnets
adding or modifying, 1-240
associated domains, 1-82
DHCP servers, 1-76
exporting, 4-91
free subnets, 1-84, 1-187
free subnets report, 1-187
importing, 3-2
importing domains, 4-72
join, 1-29
MAC pools, 1-272
moves, 1-154, 1-156
organization profiles, 1-238
profile, 1-126
profiles, 1-240
reclaiming, 1-171
router list, 1-118
routers, 1-46, 1-118
scheduled moves, 1-125
split, 1-29
splitting and joining, 1-29
Subnet Profile, 1-126, 1-240
Windows 2000, 1-267
wiring hubs, 1-86

Sybase, 1-36, 1-288, 2-2, 2-4, 4-2, 4-68, 4-72

SybaseUpdateStatistics function, 1-292

synchronizing IP addresses with host names, 1-48

T

template,add CLI, 5-241

template,copy CLI, 5-243

template,delete CLI, 5-244

template,get CLI, 5-245

template,modify CLI, 5-248

template,search CLI, 5-250

templates
DHCP Policy Templates, 1-129, 1-209, 1-231, 1-244

time servers, 1-108, 1-129, 1-133, 1-231, 1-244
list from the database, 1-133

tombstoned objects, 1-79

U

user classes, 1-68, 1-209, 1-247

user groups
exporting, 4-96
importing, 4-63

User Profile, 1-246, 1-247

user-defined fields, 1-188, 1-254
class, 1-39, 1-246
deleting, 1-40
domains, 1-202
exporting, 4-94
importing, 4-59
names, 1-134, 1-137
policies, 1-43
policy, 1-39
reverse zones, 1-40
setting the value, 1-246
Subnet Profile, 1-135, 1-247
User Profile, 1-247
values, 1-134

users, 1-68, 1-138, 1-209, 1-247
deleting, 1-39
exporting, 4-95
importing, 4-61
IP addresses, 1-140
searching for, 1-189
User Profile, 1-248

V

vercheck, 1-293

VitalQIP DHCP Report, 1-183

W

Windows 2000 sites
creating, 1-266

wiring hub slots, 1-88

wiring hubs
slot list, 1-88

Wiring_HUB, 1-88, 1-93, 1-94, 1-102, 1-105, 1-108, 1-110, 1-192, 1-227, 1-230, 3-12, 4-49, 4-56, 4-69

Wiring_HUB slots, 1-88, 1-230

Z

zone extension
DNS server, 1-256

zone extensions, 1-141
DNS servers, 1-141
exporting, 4-97
importing, 4-64
setting, 1-256
zone options
DNS, 1-250, 4-15, 4-80
exporting, 4-80
importing, 4-15
Windows 2000, 1-253
zone resource records, 1-195
zones
setting the profiles, 1-259
setting zone options, 1-250