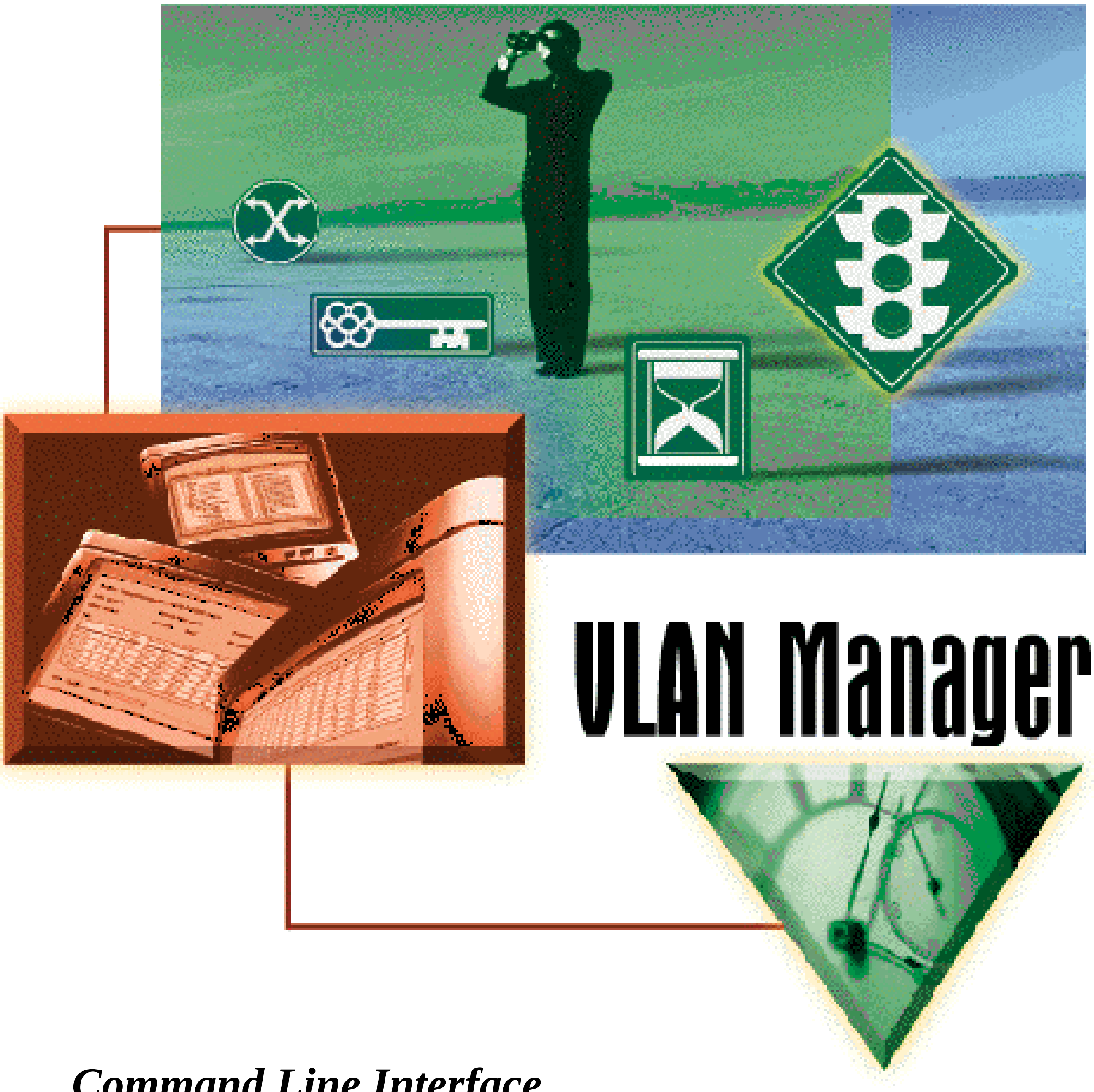
# SPECTRUM



## VLAN Manager

*Command Line Interface*

Cabletron Systems

# Notice

Cabletron Systems reserves the right to make changes in specifications and other information contained in this document without prior notice. The reader should in all cases consult Cabletron Systems to determine whether any such changes have been made.

The hardware, firmware, or software described in this manual is subject to change without notice.

IN NO EVENT SHALL CABLETRON SYSTEMS BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS MANUAL OR THE INFORMATION CONTAINED IN IT, EVEN IF CABLETRON SYSTEMS HAS BEEN ADVISED OF, KNOWN, OR SHOULD HAVE KNOWN, THE POSSIBILITY OF SUCH DAMAGES.

Cabletron Systems makes no representations or warranties to the effect that the Licensed Software is virus-free.

Printed in the United States of America.

Order Number:  9032748-03

Cabletron Systems, Inc.
P.O. Box 5005
Rochester, NH 03866-5005

# Restricted Rights Notice

(Applicable to licenses to the United States Government only.)

1. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

   Cabletron Systems, Inc., 35 Industrial Way, Rochester, New Hampshire 03866-0505.

2. (a) This computer software is submitted with restricted rights. It may not be used, reproduced, or disclosed by the Government except as provided in paragraph (b) of this Notice or as otherwise expressly stated in the contract.

   (b) This computer software may be:

   (1) Used or copied for use in or with the computer or computers for which it was acquired, including use at any Government installation to which such computer or computers may be transferred;

   (2) Used or copied for use in a backup computer if any computer for which it was acquired is inoperative;

   (3) Reproduced for safekeeping (archives) or backup purposes;

   (4) Modified, adapted, or combined with other computer software, provided that the modified, combined, or adapted portions of the derivative software incorporating restricted computer software are made subject to the same restricted rights;

   (5) Disclosed to and reproduced for use by support service contractors in accordance with subparagraphs (b) (1) through (4) of this clause, provided the Government makes such disclosure or reproduction subject to these restricted rights; and

   (6) Used or copied for use in or transferred to a replacement computer.

   (c) Notwithstanding the foregoing, if this computer software is published copyrighted computer software, it is licensed to the Government, without disclosure prohibitions, with the minimum rights set forth in paragraph (b) of this clause.

   (d) Any other rights or limitations regarding the use, duplication, or disclosure of this computer software are to be expressly stated in, or incorporated in, the contract.

   (e) This Notice shall be marked on any reproduction of this computer software, in whole or in part.

# Contents

## Chapter 1 Introduction

## Chapter 2 Commands

# Chapter 1

# Introduction

The VLAN Manager Command Line Interface (CLI) is a collection of independent Java™ applications that provide access to VLAN Server functions. These applications can be invoked from the command line or built on top of SPECTRUM's CLI Tool Kit. By providing an alternate interface to many VLAN Server functions, the VLAN Manager CLI makes it possible to create user-specific scripts and web-based interfaces for use by non-Administrative users, allowing them to change VLAN configurations and control access to specific network services without MIS/network administrator assistance.

One example of a VLAN Manager CLI implementation might be to create a web-based interface that allows an instructor in a classroom to change the network access privileges for students at certain times. The interface might consist of buttons on a browser screen that the instructor could click to isolate students in a classroom from the campus network during testing. To the instructor, the "Testing Button" is as intuitive as a light switch. It lets the instructor perform a very specific task that launches CLI applications that perform the more complex functions in the VLAN Server of changing the access for the classroom VLAN.

## Requirements

VLAN CLI installation includes the Java Runtime Environment. The windowing tool set requires the Java Foundation Classes, also known as "Swing."

## VLAN CLI Applications

The VLAN CLI Applications are launched using the run_cli.sh script within the VLAN/ tools directory in the VLAN Manager install area. Each of the applications along with their syntax is described in Chapter 2, *Commands*.

# Chapter 2

# Commands

*This chapter provides a reference to the VLAN CLI command syntax.*

The following commands can be used to launch the programs currently available:

| Command | Function |
|---|---|
| *Domain Related Commands* | |
| *CreateDomain* | Create a new Domain. |
| *DeleteDomain* | Delete a Domain. |
| *GetDomains* | Display the names of all the Domains in the VLAN Server |
| *GetDomainDetails* | Display switches, VLANs, and users in the specified Domain |
| *MacInMultipleDomains* | Control duplication of MACs in multiple domains. |
| *SetDomainPolling* | Modify polling parameters for a Domain. |
| *VLAN Related Commands* | |
| *CreateVlan* | Create a VLAN in a domain. |
| *DeleteVlan* | Delete a VLAN from a domain. |
| *GetVlanDetails* | Get VLAN information. |
| *MapUserToVlan* | Add a User to a VLAN. |
| *ModifyVlan* | Modify VLAN attributes. |
| *RemoveUserFromVlan* | Remove a User from a VLAN. |
| *SetDefaultVlan* | Change the default VLAN for a port. |
| *Switch Related Commands* | |
| *CreateSwitch* | Create a switch. |
| *DeleteSwitch* | Delete a switch. |
| *DHCPTagService* | Read, enable or disable DHCP Bindery Status |
| *GetSwitchDetails* | Get switch information. |

| Command | Function |
|---|---|
| *User Related Commands* | |
| *AddAlias* | Add an alias. |
| *AddUserToSwitch* | Add a User to a switch. |
| *DeleteAlias* | Delete an alias. |
| *DeleteUser* | Deletes a User from a domain. |
| *GetUserDetails* | Get user information. |
| *PersistUser* | Persist user on a switch. |
| *UnPersistUser* | Unpersist user on a switch. |
| *ReleaseAllCallsForUser* | Release all calls for a specific user. |
| *RemoveUserFromSwitch* | Remove a User from a switch. |
| *Security Related Commands* | |
| *AddHostToSecurity* | Add a hostname to VLAN Security. |
| *AddUserToSecurity* | Add a user id and defines access to VLAN Security. |
| *DeleteHostFromSecurity* | Deletes a host from the VLAN Manager Security |
| *DeleteHostFromSecurity* | Remove a user id from VLAN Security. |
| *Restriction Related Commands* | |
| *AddAliasRestriction* | Adds an alias restriction to a user. |
| *AddUserMobility* | Adds an entry to the Restricted Mobility Ports List. |
| *AddPortRestriction* | Denies user access to a specific switch and port. |
| *DeleteAliasRestriction* | Removes an alias restriction from a user. |
| *DeletePortRestriction* | Restores access from one user or all users to a specific switch and port. |
| *DeleteUserMobility* | Removes one or all entries from the Restricted Mobility Ports List. |

To launch any of the programs use the **`run_cli.sh`** script. The script is located in the <*vlan install area*>`/VLAN/tools` directory.

***Example:*** `run_cli.sh` <*command*> <*arguments*>

A usage message, listing applicable parameters, is displayed whenever a command that requires parameters is attempted without them.

With the exception of the GetSwitchAdj and GetSwitchLinks commands, the CLI programs can be used with both VLAN Manager, Version 1.7 and 1.8.

CLI programs use the `VLAN.properties` file in the `VLAN/tools` directory to identify the VLAN Server that will execute VLAN Manager CLI functions and to specify the port used to communicate with the VLAN Server. Use your favorite text editor to modify the `VLANServerName=`<*server hostname*> and the `VLANServerPort=`<*server port number*> parameters in the `VLAN.properties` file to define the VLAN Server hostname and port.

# Domain Related Commands

## CreateDomain

***Syntax:*** `CreateDomain` <*domain name*> `PollInt=`<*interval*> `Polling=`<*enb/dis*>

Creates a new domain with a specific polling interval and polling status.

where:

*domain name* - a domain being managed by the VLAN Manager.

*interval* - defines the time between polls expressed in seconds. The default is 300 seconds (5 minutes).

Polling - can be set to *enb* (enabled) or *dis* (disabled). The default is enabled.

***Example:*** `run_cli.sh CreateDomain building1`

Creates a domain named building1.

## DeleteDomain

***Syntax:*** `DeleteDomain` <*domain name*>

***Example:*** `run_cli.sh DeleteDomain building1`

Deletes a domain named building1.

## GetDomains

***Syntax:*** `run_cli.sh GetDomains`

No arguments are required. This command returns a list of domain names that are being managed by the VLAN Server.

***Example:*** `run_cli.sh GetDomains`

***Returned:*** Blue
Red
Green

## GetDomainDetails

***Syntax:*** `run_cli.sh GetDomainDetails` <*domain name*> `attr=`<*details*>

This command returns detailed information about a domain according to the requested details.

where:

*domain name* - a domain being managed by the VLAN Manager.

*details* - defines the specific detailed information being requested. Detail requests can be chained using commas (,) as a separator.

- `SWL` - returns a list of switch names in domain.
- `VL` - returns a list of VLANs in domain.
- `UL` - returns a list of Users in domain.
- `PL` - returns a list of Persisted Users in domain.

***Example:*** `run_cli.sh GetDomainDetails building1 attr=SWL`

***Returned:***

```
192.168.119.100
192.168.119.101
```

`run_cli.sh GetDomainDetails building1 attr=VL`

***Returned:***

```
Base
IPX_RIP/SAP_22197e5f_802.3
Blue
192.168.119.0
```

`run_cli.sh GetDomainDetails building1 attr=UL`

***Returned:***

```
User MAC: 0.0.0d.65.3e.17
User Switch Name: 192.168.119.100
Slot Num: 5
Port Num: 3
User AliasList:
IP 4.4.4.9
IP 5.5.5.9
-------------------------------
User MAC: 0.0.c.4a.34.12
User Switch Name: 192.168.119.101
Slot Num: 8
Port Num: 1
User AliasList:
IP 19.3.4.4
-------------------------------
User MAC: 0.0.c.19.3c.2
User Switch Name: 192.168.119.101
Slot Num: 8
Port Num: 3
User AliasList:
```

```
-------------------------------

run_cli.sh GetDomainDetails building1 attr=PL
```

***Returned:*** If 0.0d.65.3e.17 and 0.0.c.4a.34.12 were persisted users:

```
192.168.119.100 0.0.0d.65.3e.17
192.168.119.101 0.0.c.4a.34.12
```

## MacInMultipleDomains

***Syntax:*** `MacInMultipleDomains` <*domain name*> <*action*> <*list of MAC addresses*>

This command shows the MultiDomain status of MAC addresses and can be used to allows or disallow the specific MAC addresses to be in more than one domain.

where:

*domain name* - a domain being managed by the VLAN Manager.

*action* - ALLOW, DISALLOW, or READ

*list of MAC addresses* - are the MAC addresses that are the subject of the action. When no MAC Address are supplied, all the MAC addresses in the domain are set or read, according to the action.

***Example:***

```
run_cli.sh MacInMultipleDomains building1 ALLOW
8.0.20.76.4d.a9 0.A0.C9.62.14.b0
```

***Example:***

```
run_cli.sh MacInMultipleDomains building1 DISALLOW
8.0.20.b4.33.56 0.A0.C9.23.1f.9
```

***Example:***

```
run_cli.sh MacInMultipleDomains building1 READ
```

***Returned:***

```
8.0.20.b4.33.56 disallowed in multiple domains.
8.0.20.76.4d.a9 allowed in multiple domains.
0.A0.C9.62.14.b0 allowed in multiple domains.
0.A0.C9.23.1f.9 disallowed in multiple domains.
```

## SetDomainPolling

***Syntax:*** `SetDomainPolling` <*domain name*> `PollInt=`<*interval*> `Polling=`<*enb/dis*>

SetDomainPolling can be used to modify the Polling Interval and the Polling Status for a specific domain. The program displays messages to show whether or not the domain was updated successfully.

where:

*domain name* - a domain being managed by the VLAN Manager.

*interval* - defines the time between polls expressed in seconds. The default is 300 seconds (5 minutes).

Polling - can be set to *enb* (enabled) or *dis* (disabled). The default is enabled.

***Example:***

```
run_cli.sh SetDomainPolling building1 Polling=dis
```

Disables polling for the building1 domain.

```
run_cli.sh SetDomainPolling building1 PollInt=500
                  Polling=enb
```

Enables polling for the building1 domain and sets the poll interval to 500.

# VLAN Related Commands

## CreateVlan

***Syntax:***

```
CreateVlan <domain name> <vlan name> vlanid=<vlan id>
CreateVlan <domain name> <vlan name> tag=<value> plcy=<value>
            fld=<value> state=<value> vlanid=<vlan id>
```

Constructs a VLAN and defines its attributes using the values defined for `tag`, `plcy`, `fld`, and `state`.

where:

- *domain name* - the domain where the new VLAN will be created.
- *vlan name* - is the name assigned to the new VLAN.
- tag - any string to be used as a tag name.
- plcy - the new VLAN policy (Open or Secure)
- fld - flooding (On or Off)
- state - Enb = enabled or Dis = disabled
- *vlan id* - (optional) is an integer value that identifies this VLAN.

VLAN attributes that are not specifically defined are assigned the following default values:

- tag = same as *vlan name*
- plcy = Open
- fld = On
- state = Enabled

***Example:***

```
run_cli.sh CreateVlan building1 floor3
```

creates a VLAN named `floor3` within the `building1` domain

```
run_cli.sh CreateVlan building1 floor3 tag=MIS
            plcy=Secure fld=Off vlanid=4000
```

## DeleteVlan

***Syntax:*** `DeleteVlan` <*domain name*> <*vlan name*>

Removes a vlan.

where

- *domain name* - the domain where the VLAN will be deleted.
- *vlan name* - is the name of the VLAN being deleted.

***Example:***

```
run_cli.sh DeleteVlan building1 floor3
```

## GetVlanDetails

***Syntax:*** `GetVlanDetails` <*domain name*> <*vlan name*> `attr=`<*details*>

Returns detailed information about a VLAN for the requested VLAN attributes.

where:

- *domain name* - a domain being managed by the VLAN Manager.
- *vlan name* - is the name of the VLAN for which details are being requested.
- *details* - defines the specific attribute information being requested. Attributes can be chained using commas (,) as a separator. The attributes are:
  - DHCP - DHCP IP address
  - UL - returns a list of users in the VLAN
  - PLCY - returns the VLAN policy (Open/Secure)
  - FLD - returns the setting for flooding (On/Off)
  - STATE - returns the state of the VLAN (Enabled/Disabled)
  - VLANID - VLAN Identification number

***Example:***

```
run_cli.sh GetVlanDetails building1 floor2 attr=UL
```

***Returned:***

```
User MAC: 0.0.0d.65.3e.17
User Switch Name: 192.168.119.100
Slot Num: 5
Port Num: 3
User AliasList:
IP 4.4.4.9
IP 5.5.5.9
-------------------------------
User MAC: 0.0.c.4a.34.12
User Switch Name: 192.168.119.101
Slot Num: 8
Port Num: 1
User AliasList:
IP 19.3.4.4
-------------------------------
```

***Example:***

```
run_cli.sh GetVlanDetails building1 floor2 attr=PLCY
```

***Returned:*** Policy is: Open

***Example:***

```
run_cli.sh GetVlanDetails building1 floor2 attr=FLD
```

***Returned:*** Flooding is: On

***Example:***

```
run_cli.sh GetVlanDetails building1 floor2 attr=STATE
```

***Returned:*** State is: Enabled

## MapUserToVlan

***Syntax:***

`MapUserToVlan` <*domain name*> <*vlan name*> <*switch name*>
`MapUserToVlan` <*domain name*> <*vlan name*> <*user Mac addr*>

This command attempts to map a specific user, all users on a switch, or all users on a switch and port to a VLAN, according to the parameters provided.

where:

- *domain name* - a domain being managed by the VLAN Manager.
- *vlan name* - is the name of the VLAN to which users will be mapped.
- *switch name* - is the name of the switch where the users being mapped are attached
- *user Mac addr* - the MAC address of the user being mapped

***Example:***

```
run_cli.sh MapUserToVlan building1 floor3 0.23.4d.2a.12.4
```

Maps the user `0.23.4d.2a.12.4` to the `floor3` VLAN in the `building1` domain.

***Example:***

```
run_cli.sh MapUserToVlan building1 floor3 122.34.34.1
```

Maps all users on switch `122.34.34.1` to the `floor3` VLAN in the `building1` domain.

## ModifyVlan

***Syntax:*** `ModifyVlan` <*domain name*> <*vlan name*> `plcy=`<*open/secure*>
`fld=`<*on/off*> `state=`<*enb/dis*> `vlanid=`<*vlan id*>
`dhcpip=`<*dhcp value*>

Sets the attributes of an existing VLAN using the values defined for `plcy`, `fld`, and `state`.

where:

- *domain name* - the domain where the new VLAN will be created
- *vlan name* - is the name assigned to the new VLAN
- `state` - *enb* (enable) the VLAN - *dis* (disable) the VLAN
- `fld` - on - turn flooding on, off - turn flooding off
- `plcy` - secure - set policy to secure, open - set policy to open
- *`vlan id`* `- VLAN Identifier`
- *dhcp value* - DHCP IP address value.

***Example:***

```
run_cli.sh ModifyVlan building1 floor2 state=dis fld=on
```

Disable the `floor2` VLAN in the `building1` domain and turn on flooding.

```
run_cli.sh ModifyVlan building1 floor2 plcy=open fld=off
state=enb
```

Set the policy for the floor2 VLAN in the building1 domain to open, turn off flooding and enable the floor2 VLAN.

## RemoveUserFromVlan

***Syntax:*** `RemoveUserFromVlan` <*domain name*> <*vlan name*>
<*user Mac addr*>
`RemoveUserFromVlan` <*domain name*> <*vlan name*> <*switch name*>

RemoveUserFromVlan attempts to remove (unmap) a specific user, all users on a switch, or all users on a switch port from a VLAN.

where:

- *domain name* - a domain being managed by the VLAN Manager.
- *vlan name* - is the name of the VLAN from which users will be removed.
- *switch name* - is the name of the switch where the users being removed are attached
- *user Mac addr* - the MAC address of the user being removed

***Example:***

```
run_cli.sh RemoveUserFromVlan building1 floor1 0.32.1a.3d.21.2
```

Removes user `0.32.1a.3d.21.2` from the `floor1` VLAN in the `building1` domain.

***Example:***

```
run_cli.sh RemoveUserFromVlan building1 floor1 122.43.21.45
```

Removes all users on switch `122.43.21.45` from the `floor1` VLAN in the `building1` domain.

## SetDefaultVlan

***Syntax:*** `SetDefaultVlan` <*switch name or IP*> <*default vlan name*> <*port num*>

Assigns a name to be used as the default VLAN name. Users attached to this port will become members of the default VLAN, unless specifically mapped to another VLAN.

***Example:***

```
run_cli.sh SetDefaultVlan 122.32.4.32 floor1 3
```

# Switch Related Commands

## CreateSwitch

***Syntax:*** `CreateSwitch` <*domain name*> <*switch IP*> <*community names*>

Adds a switch to the specified domain.

where:

- *domain name* - the domain where the switch will be added.
- *switch IP* - is the address of the switch being added.
- *community names* - (optional) one or more Community Names used for access to the switch. the domain where the switch will be added. The default is *public*.

***Example:***

```
run_cli.sh CreateSwitch building1 122.34.32.12 pass1 pass2
```

Adds a switch, `122.34.32.12`, to the `building1` domain and defines two community names (`pass1` and `pass2)`.

## DeleteSwitch

***Syntax:*** `DeleteSwitch` <*switch name or IP*>

This command attempts to delete the specified switch

where:

*switch name or IP* - is the name or IP address of the switch being deleted.

***Example:***

```
run_cli.sh DeleteSwitch 122.104.23.54
```

Deletes the switch, `122.104.23.54`.

## DHCPTagService

***Syntax:*** `DHCPTagService` <*domain name*> <*action*> <*sw1 sw2... swn*>

This command can be used read the current setting of the DHCP (Dynamic Host Configuration Protocol) Bindery status, and enable or disable the DHCP client in all the switches in a domain or in selected switches.

where:

*domain name* - is a domain (managed by the VLAN Manager) containing the switches in the switch list.

*action* - defines how the command will be used. Valid actions are:

- READ - read the current status of the specified switches.
- ENB - enable DHCP Bindery Status
- DIS - disable DHCP Bindery Status

*sw1 sw2... swn* - (optional) is a list of switch IP addresses, separated by spaces. The action is applied to the entire domain when the switch list is omitted.

***Example:***

```
run_cli.sh DHCPTagService building1 READ
```

This example reads the DHCP Bindery status from all switches in the domain.

```
run_cli DHCPTagService building1 ENB 192.118.119.20
            192.118.119.22 192.118.119.23
```

This example enables the DHCP client in switches 192.118.119.20, 192.118.119.22, and 192.118.119.23.

# GetSwitchDetails

***Syntax:*** `GetSwitchDetails` *<switch name or IP>* `attr=`*<attributes>*

Retrieves detailed information about a switch according to the requested attributes.

where:

- *switch name or IP* - is the name or IP address of the switch for which details are being requested.
- *attributes* - defines the specific attribute information being requested. Attributes can be chained using commas (,) as a separator. The attributes are:
  - PORTS - Ports
  - COMNM - Community Name
  - DCMTO - DCM Timeout
  - DCMR - DCM Retry
  - POLINT - Polling interval
  - SYSUT - System Uptime
  - SYSDSCR - System Description
  - SWCND - Switch Condition
  - UL - User list

***Example:***

```
run_cli.sh GetSwitchDetails 192.168.117.100 attr=PORTS
```

***Returned:***

```
Port Name: Slot 2, Port 1
Port Num:  1
Port Mode: Other
Port Default VLAN: Base
Port Num Users: 0
-------------------------------
              .
              .
              .
-------------------------------
Port Name: Slot 2, Port 24
Port Num:  24
Port Mode: Access
Port Default VLAN: Base
Port Num Users: 0
-------------------------------
Port Name: INB Port
Port Num:  25
Port Mode: INB
Port Default VLAN:
Port Num Users: 0
-------------------------------
```

***Example:***

```
run_cli.sh GetSwitchDetails 192.168.117.100 attr=COMNM
```

***Returned:***

```
Community String is: public
```

***Example:***

```
run_cli.sh GetSwitchDetails 192.168.117.100 attr=DCMTO
```

***Returned:***

```
DCM Timeout is: 3
```

```
run_cli.sh GetSwitchDetails 192.168.117.100 attr=DCMR
```

***Returned:***

```
DCM Retry is: 10
```

***Example:***

```
run_cli.sh GetSwitchDetails 192.168.117.100 attr=POLINT
```

***Returned:***

```
Poll Interval: 300
```

***Example:***

```
run_cli.sh GetSwitchDetails 192.168.117.100 attr=SYSUT
```

***Returned:***

```
System Uptime: 5 days 20:14:19
```

***Example:***

```
run_cli.sh GetSwitchDetails 192.168.117.100 attr=SYSDSCR
```

***Returned:***

```
System Description: Cabletron 9E423-24 Rev 03.03.01 w/SecureFAST 01.08.05  05/20/98--12:12 ofc
```

***Example:***

```
run_cli.sh GetSwitchDetails 192.168.117.100 attr=SWCND
```

***Returned:***

```
Switch Condition: Normal
```

***Example:***

```
run_cli.sh GetSwitchDetails 192.168.117.100 attr=UL
```

***Returned:***

```
User MAC: 0.0.1d.7.7e.6d
User Switch Name: 192.168.117.100
Slot Num: 2
Port Num: 13
User AliasList:
-------------------------------
User MAC: 8.0.69.9.30.d4
User Switch Name: 192.168.117.100
Slot Num: 2
Port Num: 23
User AliasList:
IP 192.168.118.39
-------------------------------
```

# User Related Commands

## AddUserToSwitch

***Syntax:*** `AddUserToSwitch` *<switch name or IP>* **<***Slot***>** *<Port> <MACAddress>*

Creates a new User on the specified switch.

where:

*switch name or IP* - is the name or IP address of the switch where the user is being removed.

*Slot* - is the chassis slot number containing the switch where the user is being added.

*Port* - is the port number on the switch where the user is being added.

*user Mac addr* - the MAC address of the user is being added.

***Example:***

```
AddUserToSwitch 122.34.53.13 2 3 8.0.20.2.2.95
```

Adds the user whose MAC address is 8:0:20:2:2:95 to port 3 of board in slot 2 of the switch chassis 122.34.53.13.

## RemoveUserFromSwitch

***Syntax:*** `RemoveUserFromSwitch` *<switch name or IP>* **<***user Mac addr***>**

where:

*switch name or IP* - is the name or IP address of the switch where the user is being removed.

*user Mac addr* - the MAC address of the user is being removed.

***Example:***

```
run_cli.sh RemoveUserFromSwitch 122.33.41.3 0.3d.4a.12.54.2
```

This example removes user, `0.3d.4a.12.54.2` from switch, `122.33.41.3`.

# AddAlias

***Syntax:*** `AddAlias` *<switch name or IP>* *<user Mac addr>* *<alias type>* *<alias value>*

where:

*switch name or IP* - is the name or IP address of the switch where the user for whom the alias is being created is attached.

*user Mac addr* - the MAC address of the user for whom the alias is being created.

*alias type* - is the alias type being created. Valid entries are:

- IP
- IPX
- AppleTalk

*alias value* - is the alias address being created.

***Example:***

```
run_cli.sh AddAlias 122.42.12.32 2e.12.4d.2a.34.21 IP
122.42.13.23
```

This example adds the IP address of `122.42.13.23` as an alias for the user's MAC address, `2e.12.4d.2a.34.21.`

```
run_cli.sh AddAlias 122.42.12.32 2e.12.4d.2a.34.21 IPX
11.22.33.44.11.22.33.44.55.66
```

This example adds the IPX address of `11.22.33.44.11.22.33.44.55.66` as an alias for the for the user's MAC address, `2e.12.4d.2a.34.21.`

## DeleteAlias

***Syntax:*** `DeleteAlias` *<switch name or IP> <user Mac addr> <alias type> <alias value>*

Removes the association between a user’s MAC address and an alias.

where:

*switch name or IP* - is the name or IP address of the switch where the user, from whom the alias is being deleted, is attached.

*user Mac addr* - the MAC address of the user from whom the alias is being deleted.

*alias type* - is the alias type being deleted. Valid entries are:

- IP
- IPX
- AppleTalk

*alias value* - is the alias address being deleted

***Example:***

```
run_cli.sh DeleteAlias 122.42.12.32 IP 2e.12.4d.2a.34.21
         122.42.13.23
```

## DeleteUser

***Syntax:*** `DeleteUser` *<domain name> <user Mac addr>*

where:

*domain name* - the domain from which the user will be deleted.

*user Mac addr* - the MAC address of the user being deleted.

***Example:***

```
run_cli.sh DeleteUser building1 0:23:1d:45:2a:21
```

## GetUserDetails

***Syntax:*** `GetUserDetails` <*domain name*> <*MAC addr*> `attr=<`*`attribute(s)`*`>`

Displays selected attributes for a specific user.

where:

*domain name* - the domain containing the user.

*MAC addr* - the MAC address of the user whose details are being requested.

*attribute(s)* - available attribute(s) to display (currently only one):

**VLANS** - Vlan list

***Example:***

```
run_cli.sh GetUserDetails building1 floor2 attr=VLANS
```

***Example:***

```
run_cli.sh GetUserDetails building1 myPC attr=VLANS
```

## PersistUser

***Syntax:*** `PersistUser` <*switch name or IP*> <user *MAC addr*>

Persists the existing user on the specified switch.

where:

*switch name or IP* - the name or IP address of the switch where the user who is being persisted is attached.

*user MAC addr* - is the MAC address of the user that is being persisted.

***Example:***

```
run_cli.sh PersistUser 122.42.12.32 2e.12.4d.2a.34.21
```

This example persists the user with the MAC address of 2e.12.4d.2a.34.2, which is attached to the switch with the IP address of 122.42.12.32.

# UnPersistUser

***Syntax:*** `UnPersistUser` <*switch name or IP*> <user *MAC addr*>

Unpersists the existing user on the specified switch.

where:

*switch name or IP* - the name or IP address of the switch where the user who is being unpersisted is attached.

*user MAC addr* - is the MAC address of the user that is being unpersisted.

***Example:***

```
run_cli.sh UnPersistUser 122.42.12.32 2e.12.4d.2a.34.21
```

This example unpersists the user with the MAC address of 2e.12.4d.2a.34.2, which is attached to the switch with the IP address of 122.42.12.32.

# ReleaseAllCallsForUser

***Syntax:*** `ReleaseAllCallsForUser` <*domain name*> <*user MAC*>

Attempt to release all calls for a specified user. A message is displayed showing whether or not the call release is successful.

where

*domain name* - the domain containing the user.

*user MAC* - is the MAC address of the user whose calls are being released.

***Example:***

```
run_cli.sh ReleaseAllCallsForUser building1 32:1a:45:21:5b:12
```

Release all calls for the user 32:1a:45:21:5b:12 in the building1 domain.

# Security Related Commands

## AddHostToSecurity

***Syntax:*** `AddHostToSecurity` <*hostname*>

Adds a hostname to VLAN Security granting access to the VLAN Server from the specified host.

where:

*hostname* - is the hostname being added.

***Example:***

```
run_cli.sh AddHostToSecurity blue
```

Allows access to the VLAN Server from the hostname "blue."

## AddUserToSecurity

***Syntax:*** `AddUserToSecurity` <*user name*> <*access type*>

Creates a user model in VLAN Security and granting a specific level of access, to the VLAN Server from the specified user id.

where:

*user name* - the user id of the user being added to VLAN Security

*access type* - (optional) defines the access privilege of the user:

**RO** Read only access.

**RW** (default setting) Read and write access.

***Example:***

```
run_cli.sh AddUserToSecurity john
```

Adds user, john to VLAN Security, allowing RW access.

***Example:***

```
run_cli.sh AddUserToSecurity john RO
```

Adds User named john and restricts access to read only.

## DeleteHostFromSecurity

***Syntax:*** `DeleteHostFromSecurity` <*hostname*>

Deletes a host model from VLAN Security, thereby denying access to the VLAN Server from that system.

`where:`

*hostname* - is the hostname for the system being deleted from VLAN Manager Security

***Example:*** `run_cli.sh DeleteHostFromSecurity blue`

Denies access to the VLAN Server from the system with the hostname “blue.”

## DeleteUserFromSecurity

***Syntax:*** `DeleteUserFromSecurity` <*user name*>

Deletes a user model from VLAN Security, thereby denying access to the VLAN Server from that user id.

`where:`

*user name* is the user id of the user being deleted from VLAN Manager Security

***Example:*** `run_cli.sh DeleteUserFromSecurity john`

Denies access to the VLAN Server from the user id “john.”

# Restriction Related Commands

Six restriction commands are used to maintain three lists that manage user restrictions and port restrictions. Refer to the ***VLAN Manager User's Guide*** for more information about user and port restrictions.

- The **Restricted Mobility Ports List** provides the means for locking users into selected ports. The list is maintained on a user basis and shows a user's MAC address together with switches and ports where the user can be heard by the VLAN Server. If there are no entries in the list, then no restrictions are applied to user mobility– any user can be heard at any switch port. If a user's MAC address appears on the Restricted Mobility Ports List, then that user can only be heard on the associated switch and port.

- The **Restricted Aliases List** allows locking a user to a specific alias, which, in turn, ties the user to a specific switch and port. When a user is restricted to a particular alias for a supported protocol, then that user is automatically restricted to access at the port to which that alias is connected. Restricted Aliases entries override user mobility restrictions in the Restricted Mobility Ports List. When the aliases entries are deleted, access restrictions revert to any user mobility restrictions that may have been applied.

- The **Port Restrictions List** is maintained on a switch and port basis allowing you to restrict access to the switch and port from specific users. It shows user MAC addresses that can be heard on a particular switch and port. If there are no entries on the list, then no restrictions are applied. However, if there is at least one entry on the list, then access is permitted for those users and denied to all others.

NOTE

The term "*port*" refers to a logical port assignment that relates to a physical slot and port within the switch chassis. Refer to the chapter titled, **Managing Ports** in the ***VLAN Manager User's Guide*** for more information about logical port numbering.

## AddUserMobility

***Syntax:*** `AddUserMobility` *<domain name> <user Mac> <switch name> <port>*

This command adds the specified user MAC address to the **Restricted Mobility Ports List**, which means that the user can be heard at the specified switch and port. This command must be repeated for each switch and port where the user is allowed access.

where:

*domain name* - the domain containing the user.

*user MAC* - is the MAC address of the user whose mobility is being restricted.

*switch name* and *port* - defines the switch and port where the user is permitted to connect.

***Example:***

```
run_cli.sh AddUserMobility building1 0:0:1d:45:2a:21
         192.118.119.20 13
```

This example allows MAC address, 0:0:1d:45:2a:21 to be heard on port 13 on switch 192.118.119.20.

## DeleteUserMobility

***Syntax:*** `DeleteUserMobility` *<domain name> <user Mac>*
`DeleteUserMobility` *<domain name> <user Mac> <switch name> <port>*

Depending on the syntax used, this command removes an individual entry or clears all entries from the **Restricted Mobility Ports List**.

- Without a *switch name* and *port* specified, this command clears the Restricted Mobility Ports List, thereby removing all restrictions from User Mobility (any user can be heard at any switch and port).
- With a *switch name* and *port* specified, this command removes the entry specified by the *user MAC*, *switch name*, and *port*, thereby denying access to that switch and port from the specified user MAC address. However, if the entry being removed is the last entry in the list, this command will delete the last entry and remove all restrictions from User Mobility for this user (the user can be heard at any switch and port).

where:

*domain name* - the domain containing the user.

*user MAC* - is the MAC address of the user whose mobility is being restricted.

*switch name* and *port* - defines the switch and port where user access is being restricted.

***Example:***

```
run_cli.sh DeleteUserMobility building1 0:0:1d:45:2a:21
```

This example restores unrestricted user mobility to user 0:0:1d:45:2a:21.

***Example:***

```
run_cli.sh DeleteUserMobility building1 0:0:1d:45:2a:21
         192.118.119.20 13
```

This example removes the entry that specifies user 0:0:1d:45:2a:21 and port 13 on switch 192.118.119.20 from the Restricted Mobility Ports List. Assuming that this was not the last entry in the list, this command denies access to port 13 on switch 192.118.119.20 from this user.

## AddAliasRestriction

***Syntax:*** `AddAliasRestriction` *<switch name> <port> <user Mac> <alias type> <alias value>*

This command adds an alias restriction to the specified user MAC address. This command assigns the specified alias and alias type to the user, thereby restricting this user to access from the switch and port specified.

where:

*switch name* and *port* - defines the switch and port where alias restriction is being added.

*user MAC* - is the MAC address of the user where the alias restriction is being applied.

*alias type* - is the address format of the alias. Valid alias types are IP, IPX and NETBIOS.

*alias value* - is the alias of the user whose access is being denied.

***Example:***

```
run_cli.sh AddAliasRestriction 192.118.119.20 2
8.4.10.2.2.95 IP 192.118.22.22
```

This example assigns the IP alias 192.118.22.22 to user, 8.4.10.2.2.95 and allows access only from port 2 of switch 192.118.119.2.

## DeleteAliasRestriction

***Syntax:*** `DeleteAliasRestriction` *<switch name> <port> <user Mac> <alias type> <alias value>*

This command removes the specified alias restriction. If there are no User Mobility restrictions currently applied to the user, this command will restore access to the user. This command must be repeated for each alias restriction being removed.

where:

*switch name* and *port* - defines the switch and port where alias restriction is being removed.

*user MAC* - is the MAC address of the user whose access is being restored.

*alias type* - is the address format of the alias. Valid alias types are IP, IPX and NETBIOS.

*alias value* - is the alias of the user whose access is being restored.

***Example:***

```
run_cli.sh DeleteAliasRestriction 192.118.119.20 2
8.4.10.2.2.95 IP 192.118.22.22
```

This example removes the alias restriction, assigning IP alias 192.118.22.22 to the user 8.4.10.2.2.95 that restricted that user’s access to port 2 of switch 192.118.119.2.

## AddPortRestriction

***Syntax:*** `AddPortRestriction` *<domain name> <switch name> <port> <user Mac>*

This command adds a user to a switch’s Port Restriction List. When the Port Restrictions List contains at least one user, the users on the Port Restriction List are the only users that are permitted access to the specified switch port. This command must be repeated for each user being added to the Port Restriction List.

where:

*domain name* - the domain containing the user.

*switch name* and *port* - defines the switch and port where the port restriction is being added.

*user MAC* - is the MAC address of the user being allowed access.

***Example:***

```
run_cli.sh AddPortRestriction building1 192.118.119.20 2
        8:0:20:2:2:95
```

This example allows access from MAC, 8:0:20:2:2:95, to port 2 on switch 192.118.119.20.

## DeletePortRestriction

***Syntax:*** `DeletePortRestriction` *<domain name> <switch name> <port> <user Mac>*
`DeletePortRestriction` *<domain name> <switch name> <port>*

Depending on the syntax used, this command removes an individual entry or clears all entries from the Port Restriction List of a specific switch and port. This command must be repeated for each user when removing individual entries.

- Without a *user Mac* specified, this command clears the Port Restrictions List, thereby removing all restrictions from user access (any user can connect at this switch and port).
- With a *user Mac* specified, this command removes the entry specified by the *user MAC*, *switch name*, and *port*, thereby denying access from the specified user MAC address to that switch and port. However, if the entry being removed is the last entry in the list, this command will delete the last entry and remove all restrictions from the port (any user can connect at this switch and port).

where:

*domain name* - the domain containing the user.

*switch name* and *port* - defines the switch and port where user restrictions are being removed.

*user MAC* - is the MAC address of the user whose address is being removed.

***Example:***

```
run_cli.sh DeletePortRestriction building1 192.118.119.20 2
8.0.20.2.2.95
```

This example removes user 8.0.20.2.2.95 from the Port Restriction List for port 2 on switch 192.118.119.20. Assuming that this was not the last entry in the list, this command denies access to port 2 on switch 192.`118.119.20` from this user.

```
run_cli.sh DeletePortRestriction building1 192.118.119.20 2
```

This example clears the Port Restriction List for port 2 on switch 192.118.119.20 restoring access to any user.

# Index