



# CHAPTER 1

## Configuring and Managing Using the VXSM Command Line Interface

---

This chapter describes the role, purpose, and structure of the VXSM command line interface (CLI). The CLI provides a set of commands that enable the user to configure, manage, and troubleshoot the VXSM card.

### Role of the CLI

During initial Cisco MGX8880 voice gateway installation, troubleshooting, or operational situations requiring low-level control, the command line interface (CLI) provides the best means for accessing the switch (media gateway) or a service module installed in the media gateway chassis.

During normal operations, the tools typically used in conjunction with a Cisco MGX8850 media gateway are:

- Cisco WAN Manager—Used for managing connections to an MGX 8880 gateway.

Each PXM card or service module housed in an MGX 8850 or MGX 8880 chassis, such as the VXSM card, supports its own CLI.

The available command set for the VXSM card depends on the following:

- Privilege level of the user—For example, a low-level user cannot access certain high-level commands.
- State of the card—Active (a) state, standby (s) state, or init (i) state.
- Version of the software image that is loaded using the **setrev** command on the PMX card.

Each service module housed in a Cisco MGX8880 chassis has a set of commands specific to that card type, although many commands share similarities among card types.

### Command Line Prompt

The format of the CLI prompt is:

*name.slot number.card type.card state >*

where:

- *name*—Identifies the name of the node. (The name of the node is Unknown until a name is assigned using the **cnfname** command).

- *slot number*—Indicates the number of the slot in the chassis containing the card.
- *card type*—Identifies the Processor Switching Module (PXM) or the type of service module, such as the VXSM card.
- *card state*—Indicates the state of the card:
  - *i*—Initialization state The card is still loading application modules.
  - *a*—Active state. Card is either fully configured and ready to execute its function or is already performing its functions with live traffic.
  - *s*—Standby state. Typically, a card enters standby (s) state when it first powers on and boots or when you execute a command that places it in this state.

An example of a CLI prompt for a VXSM card follows:

```
M8850_NY.3.VXSM.a >
```

This prompt indicates that the:

- Node name is *M8850\_NY*
- Card occupies slot *3* of the MGX8850 chassis
- Card type is *VXSM*
- Card is in an active (*a*) state

## Command Notation

The notation used for VXSM commands and parameters in this document are listed below:

- Commands and their parameters are separated by a space.
- Variables appear in *italics*.
- Commands, keywords, and literal strings (such as “yes”) appear in **bold**.
- Required arguments appear within left and right arrows (“<>”).
- Optional parameters appear within square brackets (“[ ]”).
- A vertical bar ( | ) represents the logical OR function in command syntax.

## Position-Dependent and Keyword-Driven Parameters

A command can include parameters that are position-dependent or keyword-driven.

### Position-Dependent Parameters

For position-dependent parameters, you must enter parameters in the order they appear in the command syntax description or the online help system.

To create a logical port, for example, the position-dependent command syntax is:

```
addport <ifNum> <bay.line> <guaranteedRate> <maxrate> <slotID> <ifType> [vpi]
```

Also, for position-dependent parameters, you can use the number sign (#) to indicate that no change is to be made to the existing parameter corresponding to the position in which the # sign is entered.

For example, the symbolic notation for the **addh248assoc** command is shown below:

**addh248assoc** <GatewayLinkId> <MgcGroupIndex> <GatewayIpIndex> <TransportProtocol>  
<MgHeaderAddrType><dynamicTkptVersion><maxCommandMdgSize><maxReplyMsgSize>

When this command is actually entered at the system console, it takes the following form:

```
M8850_NY.3.VXSM.a > addh248assoc 1 10 12 # 16
```

In effect, the # sign in the example above causes the system to use the existing value of the <TransportProtocol> parameter. Hence, no change is made to the current value of the parameter in this position.

Note that any trailing # signs in a command can be eliminated altogether, since such a trailing symbol is of no consequence and is ignored by the system's command language interpreter.

For example, a command entered as **addh248assoc 1 10 12 # 16 # # #** is interpreted as:

```
addh248 1 10 12 # 16
```

### Keyword-Driven Parameters

For a keyword-driven parameter, a keyword must precede the variable. The keyword is preceded by a dash and followed by the parameter itself (for example, **-timeout** <secs>). The order in which you enter keyword-driven parameters does not matter—although any preceding or succeeding, position-dependent parameters must appear as shown in the command syntax description or the online help.

The function of the command in the example below is to delete more than one connection at a time. The mandatory, position-dependent connection identifier consists of a logical port (*ifNum*) and the VPI and VCI of the first connection to be deleted. Following the connection identifier, the line shows two optional, keyword-driven parameters. These keyword-driven parameters can be in any order as long as they appear after the position-dependent parameters.

```
delcons <ifNum> <vpi> <vci> [-num <num conns to del>] [-verbose <1 | 0 >]
```

