Chapter 35

Management Stacking

Introduction

This chapter describes the Stacking feature, including how it functions and how to configure it.

Benefits

Stacking affords the following advantages when managing a group of switches:

- Because stack members are connected by open standard Ethernet or uplink switch ports, the switches can be at the same physical location or across geographical areas.
- Management interfaces are conserved because each stack is managed from a single IP address or terminal connection.
- Because a stack has one configuration file that is simple to maintain for all member switches, it efficiently manages individual switches. Stacks are easy to reconfigure in tune with changing network needs.
- Stacks offer an alternative to managing a group of switches by using a CLI or GUI on each switch, which is often tedious and time-consuming.

What is Stacking?

Definition

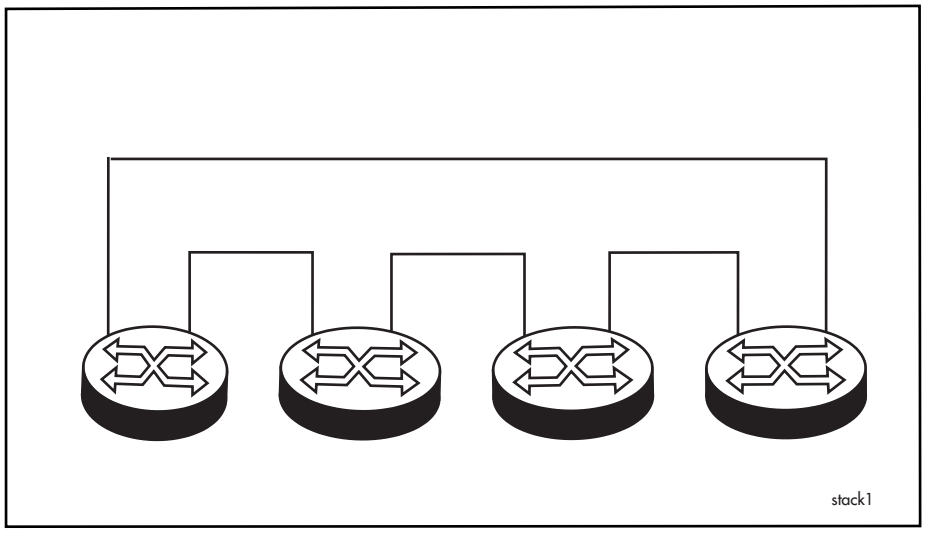
Stacking is a way to synchronise information across multiple switches and manage them as one logical device. Stacking uses a proprietary protocol to manage a group of separate switches as one.

When several switches perform similar functions, you can manage them as one. For ease and simplicity, a stack can be managed from any stack member.

Topologies

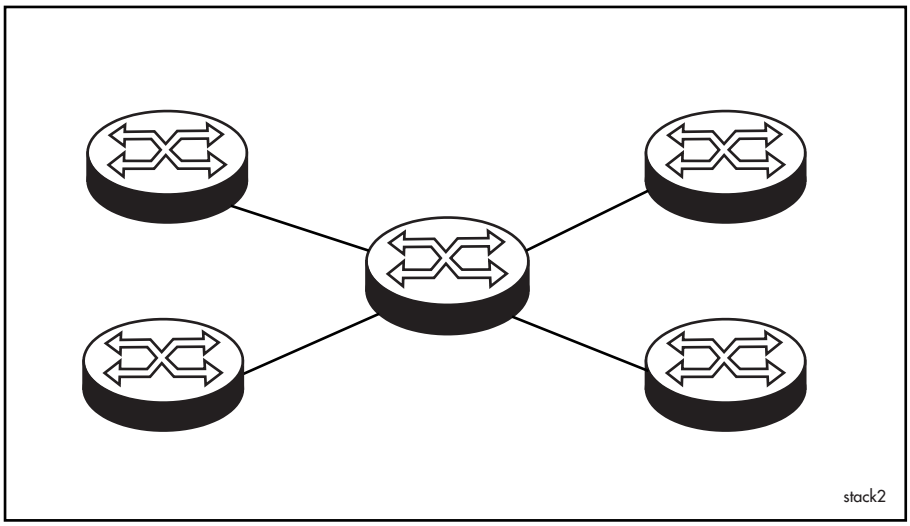
A *stack* consists of a maximum of nine switches connected by switch ports in the same Stacking VLAN. Stack members must be on the same LAN; however, they can be in different physical locations. No extra hardware is required because stack members use open standards interfaces. This allows flexible topologies; typical ones are ring and star.

Figure 35-1: Ring topology



Ring architecture allows path redundancy; the stack can be managed even if a link in it becomes unavailable. If a ring is used, loop protection must be configured on stacking ports, for example, by configuring RSTP (Rapid Spanning Tree Protocol).

Figure 35-2: Star topology



Star architecture lends itself to stacking being deployed on an existing Layer 2 network that does not require redundant paths for management.

Compatibility

Management stacking on Software Versions 2.x.x and 3.x.x is not compatible, which means you cannot combine switches running different software in one stack.

However, you can stack together any combination of Rapier, AT-8600, AT-8700XL, AT-8800, AT-8948, x900-48FE, and AT-9900 Series switches (these run Software Version 2.x.x).

How Stacking Works

Shared information

The Stacking feature centralises management by distributing and maintaining system-wide information about stack members. It also:

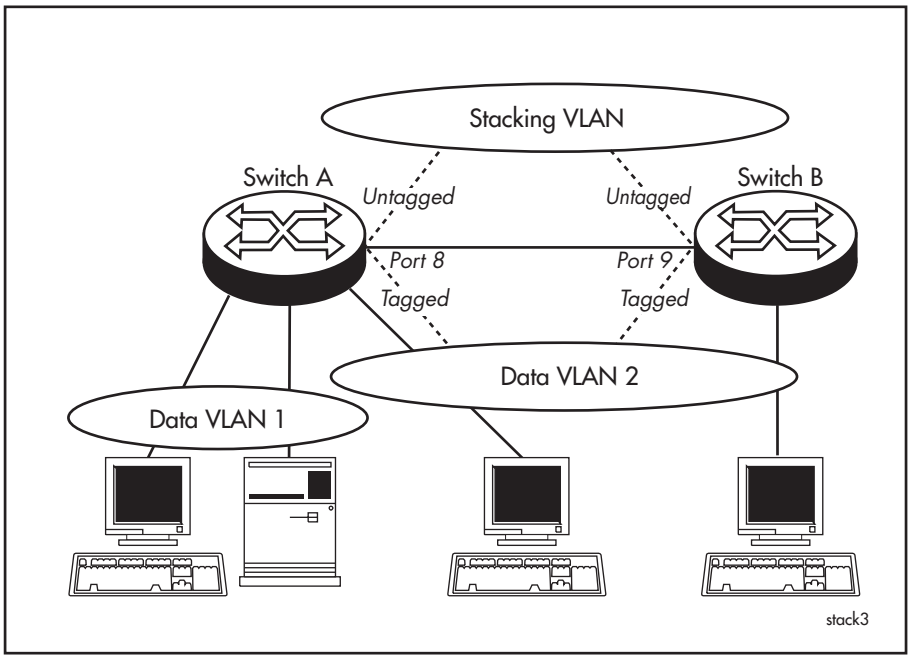
- synchronises and propagates information about individual stack members
- propagates CLI and GUI commands
- manages responses and acknowledgements
- synchronises the stack configuration file

Switches have individual *host IDs*, which you set, so that they know which device they are in a stack. Stacks have unique *stack IDs*, which you also set, so that switches know to which stack they belong. Stack IDs are essential when you have multiple stacks.

Stacked switches communicate with each other over a *Stacking interface*, which is a user-defined virtual interface, such as a VLAN. Ports in the Stacking VLAN should be added as tagged VLAN ports to data VLANs. This ensures that the Stacking VLAN carries user data.

In the following figure, data from Data VLAN 2 shares Ports 8 and 9 because they are untagged ports on the Stacking VLAN and tagged ports on the data VLAN.

Figure 35-3: VLAN ports for stacking



See "VLAN Tagging" on page 8-16 of Chapter 8, Switching for more information.

Stacking process

The following table summarises what happens after switches are configured as a stack and the stack is operational.

Stage	Description
1	When a stack is created, the user defines a virtual interface so that stack members can send each other stacking messages. When stacked switches begin transmitting over the interface, the members discover each other and synchronise to a single time. They learn information, such as individual host IDs, network configuration, and set-up. When information changes, it is updated on all members so that all members have their own up-to-date shared information.
2	Stack members learn the name of the stack configuration file. The dynamic configurations of all the stack members merge to create a stack configuration script. This script is used as the boot script for the entire stack. The most recent version of the stack configuration file is synchronised across the stack.
3	The user enters the **create config** command to add the local configuration for a new member to the stack configuration file. The stack configuration is written in a file on the switch where the user enters this command. Even when a switch is added to a stack, users can independently manage it with Host-Directed Commands (HDC). The stack synchronises to the member with the lowest host ID. When a user sets the time through SNMP, NTP, or the CLI, the stack synchronises to that time and periodically checks the time thereafter.
4	When stack members are added or removed, the user enters the **set config** command to tag the stack configuration file as one that must be synchronised across the stack. When a stack member modifies the stack configuration file, the change goes to all members.
5	When a user enters a command that produces output, such as a **show** command, stack members return output to the member that first sent the command.This member then compiles the output. When it is the same, the member merges it and displays it. When the output differs, the member displays it as separate output for each member.

Local commands

Most commands are propagated throughout the stack **except** for the following, which are local on individual stack members:

- add stack interface
- clear
- clear flash
- copy
- create config
- delete stack interface
- disable stack
- enable stack
- help
- set system hostid (other **set system** commands are stack-wide)
- set time
- show config dynamic
- show ffile
- show ffile check
- telnet

Local commands cannot be host directed, so you must type the command into the relevant switch's CLI.

Configuring a Stack

Planning

- Stacked devices must be within the same L2 forwarding domain.
- Stack members must have the same software version level. New versions should be loaded individually on members and without stacking enabled.
- Regardless of whether a stack is managed from an asynchronous port or an IP address, the stack member from which you manage it must have at least one port in a VLAN that has an IP address. Some protocols, for example RIP and OSPF, require each stack member to have a different IP address.
- The same Stacking VLAN must be created on every switch in the stack. This VLAN should not be used for other data. Stacking ports should be untagged members of the Stacking VLAN.
- To share data VLANs across a stack, ports in the Stacking VLAN should be tagged members of the data VLANs.
- If you need redundancy, use a ring topology. However, do not connect the ring until after you configure and enable RSTP (Rapid Spanning Tree Protocol) on each switch in the stack.
- When configuring SNMP on an existing stack, specify a list of host IDs before each SNMP command.
- If firewall protection is required, connect a separate firewall device as the external gateway for the stack.
- A switch can be a member of only one stack at a time.
- It is good practice to label the outside cases with individual stack IDs and host IDs after you have added switches to a stack.

Creating a stack

Before configuring features for the whole stack, use the following steps to configure stacking on **each** switch meant for the stack. This procedure assumes the switch is not pre-configured.

1. **Set the system host ID on the switch.**

   The host ID uniquely identifies this stack member in the stack. Give each switch a different host ID by using the command:

   ```
   set system hostid=1..32
   ```

2. **Create a VLAN.**

   Create a VLAN to use for stacking. The same VLAN must be on every switch in the stack. Use the command:

   ```
   create vlan=vlan-name vid=vid
   ```

   where:

   - *vlan-name is* a unique name from 1 to 32 characters.
   - *vid* is a unique number from 2 to 4094.

   See "Virtual Local Area Networks (VLANs)" on page 8-14 of Chapter 8, Switching for details about VLANs, including procedures for creating them.

3. **Add ports to the VLAN.**

   Add ports to the Stacking VLAN on the switch by using the command:

   ```
   add vlan=vid port=port-list
   ```

   where *port-list is* a:

   - port number or
   - hyphen-separated range of port numbers or
   - comma-separated list of port numbers and/or ranges.

   Port numbers in the Stacking VLAN are different for each switch; the port number format is *hostid.0.port*, for example, 1.0.24.

   If desired, check the VLAN configuration by using the **show vlan** command.

4. **Enable the Rapid Spanning Tree Protocol (RSTP) for rings.**

   If the stack is in a ring formation, enable RSTP on the Stacking VLAN to prevent traffic loops.

   Use the commands:

   ```
   create stp=stp-name
   enable stp=stp-name
   add stp=stp-name vlan=vid
   set stp=stp-name mode=rapid
   ```

   See "Configuring STP" on page 9-8 of Chapter 9, Spanning Trees for details about these commands.

5. **Add the VLAN as a Stacking interface.**

   Add an interface so that stack members can communicate with each other. Use the command:

   ```
   add stack interface=interface
   ```

6. **Set authentication for the stack.**

   If desired, set security for the stack member. The authentication type and password must be the same for all stack members. A stack member ignores messages from switches with a different authentication type and password. Use the command:

   ```
   set stack authentication={md5|plaintext} password=password
   ```

7. **Give the stack a unique identifier.**

   When there is more than one stack on the same network, give them different identifiers. Use the command:

   ```
   set stack stackid=1..16
   ```

   If you later want to change the stack ID for an *individual* switch, disable stacking on it, set the stack ID, and then enable stacking on it again.

   Important As you configure each subsequent switch, physically connect it to the port that you configured in Step 3.

8. **Enable stacking on the switch.**

   This command is stored in the stack configuration file and is activated if the switch is restarted. Use the command:

   ```
   enable stack
   ```

   Stack-wide commands are propagated after this point.

9. **Verify discovery.**

   Wait briefly for stack members to discover each other and use the following command to check that all host IDs are present:

   ```
   show stack
   ```

10. **Optionally set the correct time.**

    Switches automatically synchronise to the time on one device; however, this may not be the correct time. From any stack member, verify the time and set it as necessary by using the commands:

    ```
    show time
    set time
    ```

11. **Create and use a stack configuration file for the switch.**

    Create a stack configuration file and update the local configuration file with it. The stack shares the stacking script, which is synchronised across the stack. When a switch modifies the stack configuration file, the change goes to all members.

    Use the commands:

    ```
    create config=stack-filename.cfg
    set config=stack-filename.cfg
    ```

    See Chapter 5, Managing Configuration Files and Software Versions for details about these commands.

    If desired, check the configuration by using the **show config dynamic** command.

12. **After the first switch, set the others so that they use the stack configuration file.**

    For subsequent switches, execute the stack configuration file and enable stacking with the command:

    ```
    hostid: restart switch
    ```

Next Steps

You are now ready to configure the stack using other commands in the Software Reference. With the exception of the local commands previously noted in this chapter, commands are propagated to all members of the stack.

Tips

After stacking is enabled, remember to specify ports in the *hostid.0.port* format.

When you create VLANs for user data, add ports in the Stacking VLAN as tagged ports. This ensures that ports for stacking messages also carry user data.

To configure services on one stack member, a range, or a list, use the Host-Directed Command (HDC) feature whereby you send commands to specific members by typing the host ID, a colon, and then the command.

To later change the stack configuration, connect to any stack member and change the stack configuration file.

Adding a switch to a stack

Before you begin:

If a port is not available on an **existing** stack member, add one to the Stacking VLAN where the new switch is to be connected. Use the command:

```
add vlan=vid port=port
```

where *port* is a port number. Port numbers for stacking have the format *hostid.0.port*, for example 1.0.24.

If desired, check the VLAN configuration by using the **show vlan** command.

1. **Turn on the new switch.**

   Power up the new switch without network connections and without pre-configuring it. To ensure a clear starting configuration, use the commands:

   ```
   set config=none
   restart switch
   ```

2. **Set the system host ID on the switch.**

   The host ID uniquely identifies this stack member in the stack. You must be directly connected to the new switch and use the command:

   ```
   set system hostid=1..32
   ```

   If desired, view the host ID by using the **show system** command.

3. **Create a VLAN on the new switch.**

   Use the same VLAN as on other stack members. Use the command:

   ```
   create vlan=vlan-name vid=vid
   ```

   where:

   - *vlan-name is* a unique name from 1 to 32 characters
   - *vid* is a unique number from 2 to 4094.

   See "Virtual Local Area Networks (VLANs)" on page 8-14 of Chapter 8, Switching for details about VLANs, including procedures for creating them.

4. **Add ports to the VLAN.**

   Add ports to the Stacking VLAN on the switch by using the command:

   ```
   add vlan=vid port=port-list
   ```

   where *port-list* is a:

   - port number or
   - hyphen-separated range of port numbers or
   - comma-separated list of port numbers and/or ranges

   Port numbers in the Stacking VLAN are different for each switch; the port number format is *hostid.0.port*, for example, 1.0.24.

   If desired, check the VLAN configuration by using the **show vlan** command.

5. **Enable the Rapid Spanning Tree Protocol (RSTP) for rings.**

   If the stack is in a ring formation, enable RSTP on the Stacking VLAN to prevent traffic loops.

   Use the commands:

   ```
   create stp=stp-name
   enable stp=stp-name
   add stp=stp-name vlan=vid
   set stp=stp-name mode=rapid
   ```

   See "Configuring STP" on page 9-8 of Chapter 9, Spanning Trees for details about these commands.

6. **Add the VLAN as a Stacking interface.**

   Add an interface so that stack members can communicate with each other. Use the command:

   ```
   add stack interface=interface
   ```

7. **Set authentication for the stack.**

   The authentication type and password must be the same for all stack members. If security is set for the stack, set it for the new switch by using the command:

   ```
   set stack authentication={md5|plaintext} password=password
   ```

8. **Set the stack ID on the new switch.**

   Give the new switch the unique ID for the stack where it is a member. Use the command:

   ```
   set stack stackid=1..16
   ```

   If you later want to change the stack ID for an *individual* switch, disable stacking on it, set the stack ID, and then enable stacking on it again.

9. **Physically connect the new switch using Stacking ports previously configured.**

10. **Enable stacking on the switch.**

    This command is stored in the stack configuration file and is activated if the switch is restarted. Use the command:

    ```
    enable stack
    ```

11. **Verify discovery.**

    Wait briefly for stack members to discover each other and use the following command to check that all host IDs are present:

    ```
    show stack
    ```

12. **Create and use a stack configuration file for the switch.**

    Create a stack configuration file and update the local configuration file with it. The stack shares the stacking script, which is synchronised across the stack. When a switch modifies the stack configuration file, the change goes to all members. The stack configuration file name must be a maximum of 8 characters with a 3-character extension.

    Use the commands:

    ```
    create config=stack-filename.cfg
    set config=stack-filename.cfg
    ```

See Chapter 5, Managing Configuration Files and Software Versions for details about these commands.

If desired, check the configuration by using the **show config dynamic** command.

13. **Start the new switch as a stack member.**

    Execute the stack configuration file and enable stacking with the command:

    ```
    hostid: restart switch
    ```

Replacing a member of a stack

The replacement must be the same model of switch as the one you are replacing.

1. **Turn off and unplug the switch you want to replace.**

   The switch is removed from the neighbour lists of the stack members.

2. **Turn on the replacement switch.**

   Power up the new switch without network connections and without pre-configuring it. To ensure a clear starting configuration, use the commands:

   ```
   set config=none
   restart switch
   ```

3. **Optionally set the time on the replacement.**

   If the host ID of the replacement will be the lowest in the stack, the stack will synchronise to the time of the replacement, which may be undesirable. Verify the time on the replacement and set it as necessary by using the commands:

   ```
   show time
   set time
   ```

4. **Set the system host ID on the replacement.**

   The host ID uniquely identifies this stack member in the stack. The host ID must be the same as the switch it is replacing. Use the command:

   ```
   set system hostid=1..32
   ```

5. **Create a VLAN.**

   Use the same VLAN as on other switches in the stack. Use the command:

   ```
   create vlan=vlan-name vid=vid
   ```

   See "Virtual Local Area Networks (VLANs)" on page 8-14 of Chapter 8, Switching for details about VLANs, including procedures for creating them.

6. **Add ports to the VLAN.**

   Add ports to the Stacking VLAN on the switch by using the command:

   ```
   add vlan=vid port=port-list
   ```

   where *port-list* is a:

   - port number or
   - hyphen-separated range of port numbers or
   - comma-separated list of port numbers and/or ranges

Port numbers in the Stacking VLAN are different for each switch; the port number format is *hostid.0.port*, for example, 1.0.24.

If desired, check the VLAN configuration by using the **show vlan** command.

7. Enable the Rapid Spanning Tree Protocol (RSTP) for rings.

If the stack is in a ring formation, enable RSTP on the Stacking VLAN to prevent traffic loops

Use the commands:

```
create stp=stp-name
enable stp=stp-name
add stp=stp-name vlan=vid
set stp=stp-name mode=rapid
```

See "Configuring STP" on page 9-8 of Chapter 9, Spanning Trees for details about these commands.

8. Add the VLAN as a Stacking interface.

Add an interface so that stack members can communicate with each other. Use the command:

```
add stack interface=interface
```

9. Set authentication for the stack.

The authentication type and password must be the same for all stack members. If security is set for the stack, set it for the new switch by using the command:

```
set stack authentication={md5|plaintext} password=password
```

10. Set the stack ID on the replacement switch.

Give the new switch the unique ID for the stack where it is a member. Use the command:

```
set stack stackid=1..16
```

If you later want to change the stack ID for an *individual* switch, disable stacking on it, set the stack ID, and then enable stacking on it again.

11. Set the correct stack configuration file.

Ensure the most current stack configuration file is loaded on the replacement when it joins the stack. Use the commands:

```
set config=stack-filename.cfg
```

If the file name already exists on the switch, remove the old file with the command:

```
delete file=stack-filename.cfg
```

12. Enable stacking on the replacement switch.

The stack configuration file is sent to the replacement switch and it joins the stack when you use the command:

```
enable stack
```

Because this command is stored in the stack configuration file, it is activated if you restart the switch.

Important Connect the replacement switch to the stack using the same ports as the original setup.

13. Verify discovery.

Wait briefly for stack members to discover each other, and use the following command to check that all host IDs are present:

```
show stack
```

14. Replace serial numbers in the stack configuration file.

From any stack member, open the stack configuration file by using the command:

```
edit stack-filename.cfg
```

Look for the System Configuration section in the stack configuration file, shown in the following example:

```
#
SYSTEM configuration
#
1: set system hostid=1 serialnumber=49901563
2: set system hostid=2 serialnumber=50429430
3: set system hostid=3 serialnumber=41383493
#
```

Type the serial number of the replacement switch over the number of the switch you removed.

15. Propagate the edited stack configuration file throughout the stack.

Use the command:

```
set config=stack-filename.cfg
```

16. Start the replacement switch as a stack member.

For the replacement switch to use the edited stack configuration file, use the command:

```
hostid: restart switch
```

Permanently removing a member from a stack

If you want to manage the switch remotely after removing it from the stack, ensure it has an IP address on an interface to which you can connect.

1. Disable stacking on the desired stack member.

Connect directly to an IP address or terminal for the stack member, and use the command:

```
disable stack
```

2. Run the non-stack switch as standalone.

If desired, reset port number and format so that the removed switch uses the simple port numbering format. Set the switch to an individual configuration file, such as the one it previously used or boot.cfg, and ensure it does not contain a host ID. Use the commands:

```
set config=individual-filename.cfg
restart switch
```

To add the switch to a stack again, you must directly connect to it.

3. **Create a new stack configuration file for the stack.**

   Because you removed a stack member, you must re-synchronise the stack configuration file across the stack. The stack configuration file name must be a maximum of 8 characters with a 3-character extension.

   Connect to any stack member and use the commands:

   ```
   create config=stack-filename.cfg
   set config=stack-filename.cfg
   ```

   See Chapter 5, Managing Configuration Files and Software Versions for details about these commands.

   If desired, check the configuration by using the **show config dynamic** command.

Configuration Example

This section describes a basic example of stacking commands to set up two directly connected switches, Switch A and Switch B. It reflects general procedures titled "Creating a stack" on page 35-6.

Figure 35-4: Switch A

```
#Set the Hostid
set system hostid=1 serialnumber=40178884

#Save the running configuration
create config=switchA.cfg
set config=switchA.cfg

#Restart the switch to ensure correct port numbering
restart reboot

#Create a VLAN to be used for stacking
create vlan=stack vid=1000

#Add a port to the stacking VLAN
add vlan=stack port=1.0.25

#Set your stacking VLAN as a stacking interface
add stack interface=vlan1000

#Enable stacking on the switch
enable stack

#Create a stack configuration file
create config=stackconf.cfg

#Set the switch to the configuration file you have created
set config=stackconf.cfg
```

Figure 35-5: Switch B

```
#Set the Hostid
set system hostid=2 serialnumber=50435286

#Save the running configuration
create config=switchB.cfg
set config=switchB.cfg

#Restart the switch to ensure correct port numbering
restart reboot

#Create a VLAN to be used for stacking
create vlan=stack VID=1000

#Add a port to the stacking VLAN
add vlan=stack port=2.0.25

#Set the stacking VLAN as a stacking interface
add stack interface=vlan1000

#Enable stacking on the switch
enable stack

#Create a stack configuration file
create config=stackconf.cfg

#Set the switch to the configuration file you have created
set config=stackconf.cfg

#After both switches are connected using pre-defined stacking
ports, verify that all host IDs are present
show stack
```

Command Reference

This section describes the commands available on the switch for the stacking feature.

The shortest valid command is denoted by capital letters in the Syntax section. See "Conventions" on page xxxviii of About this Software Reference for additional conventions used to describe command syntax. See Appendix A, Messages for a complete list of messages and meanings.

Host-Directed Commands (HDC)

Syntax *hostid: non-local command*

where *hostid* is a:

- unique number from 1 to 32
- range of unique numbers from 1 to 32
- comma-separated list

Description The HDC capability sends a command to specific stack members instead of letting it propagate through the entire stack. HDC provides expanded flexibility when configuring large and complex Layer 3 networks and switches are in stacks.

Example To add an IP route to the stack member with host ID 3, use the command:

```
3: add ip route=0.0.0.0 int=vlan100 next=192.168.0.1
   int=vlan1000
```

Only the stack member with host ID 3 accepts the command and adds the IP route. When you enter the **create config** command, the HDC directive is added to the configuration script for the stack.

To enable IP on hosts 1, 3, and 5, use the following command. Other stack members will report that the command is not applicable to them.

```
1,3,5: ena ip
```

To create a Test VLAN (VID 10) on stack members 1 to 5, use the following command. Other stack members will report that the command is not applicable to them.

```
1-5: cre vlan=test vid=10
```

add stack interface

Syntax `ADD STAck INTerface=`*`interface`*

where *interface* is a character string formed by concatenating the interface type, vlan, with the VID (for example, vlan1000)

Description This command adds a predefined interface to a local switch. This interface sends and receives Stacking messages in stacks. It can also carry user data, depending on how other VLANs are configured after the stack is operational.

The interface must already exist. To see information about the current VLAN, use the **show vlan** command in the *Switching* chapter.

Example To add the *vlan1000* interface to this stack and enable communication within the stack, use the command:

```
add sta int=vlan1000
```

Related Commands
delete stack interface
enable stack
set stack stackid
show stack

delete stack interface

Syntax `DELete STAck INTerface=`*`interface`*

where *interface* is a character string formed by concatenating the interface type, vlan, with the VID (for example, vlan1000)

Description This command is local and deletes an interface that previously had been added to this stack member.

Example To delete the *vlan1000* interface from a stack, use the command:

```
del sta int=vlan1000
```

Related Commands
add stack interface
disable stack
enable stack debug
show stack

disable stack

Syntax `DISable STAck`

Description This command is local and cannot be run from a script. It disables stacking for the switch where the command is entered. In addition to a direct connection, you can log onto the switch in a telnet or TTY session.

When a switch has been disabled, it is not visible to the stack. You must manage it through another connection, such as a telnet or TTY session, or enable stacking again through the telnet or TTY session. If you disable stacking, you do not need to reboot.

Example To disable stacking on a switch, use the command:

```
dis sta
```

Related Commands
delete stack interface
enable stack
set stack stackid
show stack

disable stack debug

Syntax

```
DISable STAck DEBug[={ALL|STAte|PACket|INFO}]
   [DETail={SUMmary|FULL|RAW}]
```

Description

This command disables stacking debugging on individual stack members, not the entire stack. To disable the debugging feature, you must be logged onto the stack member in a telnet or TTY session. The following table explains the parameters.

Parameter	Description	
DEBug	Specifies debug options to disable. Default: **all**	
	ALL	Includes Info, State, and Packet (Summary).
	STAte	Stacking status changes.
	PACket	A stacking protocol packet has been sent or received.
	INfo	General information about stacking.
DETail	Specifies details for the **disable stack debug=packet** command only. Default: **summary**	
	SUMmary	Summary information in the packet header only.
	FULL	Summary and the full data sections.
	RAW	Full summary with the entire packet in natural hexadecimal format.

Example

To disable all packet debugging, use the command:

```
dis sta deb=pac
```

Related Commands

delete stack interface
disable stack
enable stack
disable stack debug
set stack stackid
show stack

enable stack

Syntax

```
ENAble STAck
```

Description

This command is local and allows stacking on a switch; it is stored in the stack configuration file. Before enabling this feature, you must have already set a host ID on the switch by using the **set system hostid** command. You must also be logged onto the stack member in a telnet or TTY session.

Example

To enable stacking on a switch, use the command:

```
ena sta
```

Related Commands

add stack interface
disable stack
set stack authentication
set stack stackid
show stack

enable stack debug

Syntax

```
ENAble STAck DEBug[={ALL|STAte|PACket|INFO}]
   [DETail={SUMmary|FULL|RAW}]
```

Description

This command enables stacking debugging for individual stack members, not the entire stack, and displays information for the specific switch. When a debug option is enabled, major system stacking errors are labelled *Stk_Error*. An error log shows where the error occurred.

To enable this feature, you must be logged onto the stack member in a telnet or TTY session. The following table explains the parameters.

Parameter	Description	
DEBug	Specifies debug options to enable. Default: **all**	
	ALL	Includes Info, State, and Packet (Summary).
	STAte	Shows stacking status changes.
	PACket	Displays information about stacking packets received and sent.
	INfo	Gives general information about stacking.
DETail	For the **enable stack debug=packet** command only. Default: **summary**	
	SUMmary	Summary information in the packet header only.
	FULL	Summary and the full data sections.
	RAW	Full summary with the entire packet in natural hexadecimal format.

Example

To enable packet debugging with summary only, use the command:

```
ena sta deb=pack det=sum
```

Related Commands

disable stack
disable stack debug
enable stack
show stack

set stack authentication

Syntax

```
SET STAck AUthentication={NONE|PLAintext|MD5}
   [PASSword=password]
```

Description

This command sets authentication on a local switch meant for a stack. Stacking packets are then authenticated by the method and password that you set. Authentication type and password must be the same for all stack members.

Authentication is optional and if you want it, you must set it before you enable stacking on the switch. Otherwise, disable stacking on the switch, set authentication, and enable stacking again. The following table explains the parameters.

Parameter	Description	
AUthentication	Sets the switch so that communication between stack members is authenticated with the password you define with the command. Packets without the password are discarded when received. Switches in the same stack must have the same password. Default: **none**	
	NONE	Disables authentication.
	PLAintext	Sets a cleartext password to authenticate stacking packets.
	MD5	Sets an encrypted password to authenticate stacking packets.
PASSword	Unique string 1 to 8 characters long that authenticates stacking packets. The *password* contains any printable characters and is case sensitive.	

Example

To set the stack so the unencrypted password “amigo” authenticates stacking packets, use the command:

```
set sta au=pla pass=amigo
```

Related Commands

add stack interface
disable stack
enable stack
set stack stackid
show stack

set stack stackid

Syntax `SET STAck STACKid=1..16`

Description This command sets a switch so that it belongs to a specific stack. Use it when changing the stack ID for the whole stack (all members) or when moving a member from one stack to another. Disable stacking on members that you want to change. Update the stack configuration file after you make changes by using the **create config** command.

When **stackid** is not for an existing stack or the default stack, then a new stack is created and given the number you specify. In this case, you get a message about the new stack. The default is 1.

Example To set the stack ID to 2, use the command:

```
set sta stack=2
```

Related Commands

add stack interface
disable stack
enable stack
enable stack debug
set stack authentication
show stack

set system hostid

Syntax `SET SYStem HOSTid={0..32} SERialnumber=`*`serial-number`*

Description This command sets a host ID for the switch as well as a new port numbering format. This command is local whereas other **set system** commands are propagated throughout a stack when the stacking feature is enabled. The switch must be restarted/rebooted to implement changes made by this command.

A stack can consist of a maximum of nine switches, and each switch must have a host ID before you can enable stacking on it. Host IDs uniquely identify each switch in its stack. The **hostid** parameter lets you specify a unique number from 1 to 32 for each stack member. This parameter also specifies the extended port numbering format for switch ports, which is hostid.board.port. The board ID is 0 (zero). Setting the host ID to 0 removes it and specifies simple port numbering (port format) for the switch ports. The default is 0.

The **serialnumber** parameter is the serial number for the switch and is usually optional; however, it is required when writing a stack configuration. It can be verified with the **show system serialnumber** command on page 4-53 of Chapter 4, Configuring and Monitoring the System. The default is the serial number for the switch.

Example To set 25 as the host ID and 50435286 as the serial number for a switch that you want to include in a stack, use the command:

```
set sys host=25 ser=50435286
```

Related Commands
add stack interface
enable stack

show stack

Syntax `SHow STAck [COUnters]`

Description This command displays general information as well as debugging information about this switch and other members of the stack (Figure 35-6, Table 35-1).

The **counters** parameter displays various counter values associated with stacking (Figure 35-7 on page 35-25, Table 35-2 on page 35-25).

The following example shows a stack with four members, all with identical output.

Figure 35-6: Example output from the **show stack** command

```
Stack Host 1-4
Stacking information
------------------------------------------------------------
Module status ................. Enabled
Stack id ...................... 1
Authentication ................ MD5
Password ...................... friend
Direct connected host ......... 2
Stacking interfaces ........... vlan1000

Stack Members:
HostId MacAddress           State
-----------------------------------
     1 00-00-cd-07-8a-00     up
     2 00-00-cd-02-e4-e0     up
     3 00-00-cd-00-ea-f9     up
     4 00-00-cd-02-e4-b0     up
-----------------------------------
```

Table 35-1: Parameters in output of the **show stack** command

Parameter	Meaning
Stack Host 1-2	Stack members to which the output applies.
	Stacking Information
Module Status	Whether Stacking is operative on the switch.
Stack ID	When more than one stack is on the same network, the stack ID to which this member belongs.
Authentication	The type of authentication, if any.
Password	Unique string of characters that authenticates packets.
Direct connected host	Host ID of the stack member to which you are directly connected through a telnet or TTY session.
Stacking Interfaces	The stacking interface for this stack member.
	Stack Members
Host ID	The unique ID for this stack member.
MAC Address	The physical address of this stack member on the network.
State	Whether all stack members are participating in the stack.

Figure 35-7: Example output from the **show stack counters** command

```
Manager >
Stack Host 1:

Stack Counters
-------------------------------------------------------------
General stacking counters:
  stkDebugErrors ........ 2

Stacking packet counters:
  rxStkPktTotal ....... 4871   txStkPktTotal ....... 2646
  rxStkPktDiscard ........ 0   txStkPktFail ........... 0

Stacking database counters:
  stkDbSdrCount .......... 7
-------------------------------------------------------------
```

Table 35-2: Parameters in output of the **show stack counters** command

Parameter	Meaning
StkDebugErrors	Major stacking system errors that could affect stacking operations.
rxStkPktTotal	Total stacking packets received.
rxStkPktDiscard	Number of stacking packets received and discarded.
txStkPktTotal	Total stacking packets transmitted.
txStkPktFail	Number of stacking packets that were transmitted but failed.
StkDbSdrCount	Total stacking records.

Examples

To display stacking counters, use the command:

```
sh sta cou
```

Related Commands

add stack interface
delete stack interface
disable stack
disable stack debug
enable stack
enable stack debug
set stack stackid