Technical Guide

Virtual Chassis Stacking (VCStack)

Feature Overview and Configuration Guide

Introduction

Virtual Chassis Stacking—VCStack™ —is the name given to two or more Allied Telesis switches that are configured to operate as a single switch. From a configuration and management point of view, it is as though the switches are just one device with a seamless transition from the ports of one stack member to the ports of the next.

VCStack provides a highly available system where network resources are spread out across stacked units, reducing the impact if one of the units fails. Aggregating switch ports on different units across the stack provides excellent network resiliency.

When configuring a VCStack, there are no limitations on how the ports of one stack member can interact with the ports of another stack member—they can belong to VLANs together, they can belong to port aggregations together, they can mirror to each other, and port-range configuration can span multiple stack members. The stack member ports truly operate as though they all belong to one virtual switch.

The same applies with Layer 2 and Layer 3 switching (both unicast and multicast). The stack operates as a single device and is not perceived by end users, or the traffic itself, to be any more than a single network node.

There are some limitations to the seamlessness of virtual chassis stacking; for example, the file systems on the individual stack members remain discrete.

This guide explains the physical creation of a VCStack, the configuration required on stack members, and how to monitor the operation of the VCStack. It also provides an understanding of how the stack behaves when a stack member stops responding.

AlliedWare Plus™
OPERATING SYSTEM

Products and software version that apply to this guide

To see whether your product supports VCStack, see the following documents:

- the product's Datasheet
- the product's Command Reference

These documents are available from the above links on our website at alliedtelesis.com.

Feature support may change in later software versions. For the latest information, see the above documents.

Which products support VCStack?

The table below lists product series that support VCStack along with the maximum number of units that can be stacked together within each series.

PRODUCT SERIES	MAX UNITS	NOTES
GS970EMX	4	GS970EMX/10 not supported.
GS980MX	4	
XS900MX	2	
x240	2	Stacking is not enabled by default.
x330	6	x330-10GTX not supported.
x530/x530L	8	4 units using 1G stacking.
x550	4	
x930	8	
x950	8	
SBx908GEN2	4	Stacking is not enabled by default.
SBx8100	2	

Stacking with legacy switches

For comprehensive stacking information on our legacy switches, refer to the section titled "Stacking on legacy switches" located near the end of this document.

Content

Benefits of virtual chassis stacking

VCStack provides a highly available solution for uninterrupted network access, simplified network management options, and a high degree of flexibility.

This section is an overview of the benefits of virtual chassis stacking.

High availability

Link aggregation across stack members

Aggregated links from access switches to the VCStack can terminate on different stack members. If a link in the aggregation is removed or fails, there is little network disruption.

Rolling reboot

A major benefit of the VCStack solution is that it provides unit resiliency - if one unit in the stack goes down, the other members of the stack continue to forward data. It is desirable for this continuity of service to persist even when the stack is being rebooted. Rolling reboot maintains continuous service by rebooting the stack in a rolling sequence, so that there is at least one unit actively forwarding data at all times during the stack reboot sequence.

Resiliency link

The dedicated stacking link is backed up by a further resiliency link. If the stacking link fails, communications between the stack members is maintained to enable graceful reconfiguration. This enables a smooth reconfiguration process and prevents the issues associated with 'split brain'. When a stack splits, it is called ‘split brain’. ‘Split brain’ can occur when a VCStack splits into two or more isolated switches or stacks when stacking management VLAN traffic fails. If the stacks are not connected via a resiliency link, each isolated switch or stack will act as the VCStack master, and will retain the stack virtual-mac address which could cause network disruption.

The resiliency-link feature with disabled-master-monitoring provides a backup mechanism for VCStack to prevent dual active masters if the stack separates.

Fast failover

In a VCStack environment, one of the stack members acts as the master switch and provides decision-making capabilities for the virtual chassis.

All other VCStack members operate in an active standby mode. They actively learn routing and forwarding information for the network, ensuring that in the event of a master failure, another member can seamlessly take control of the virtual chassis with minimal network downtime.

Synchronization of hardware forwarding tables, VLAN state tables, and port state tables is maintained across stack members for the following protocols:

- Spanning tree
- EPSR
- 802.1x
- ARP
- IPv6 Neighbor Discovery
- RIP
- RIPng
- OSPFv2
- OSPFv3
- BGP
- BGP4+
- IGMP
- MLD
- PIM
- PIMv6

Coupled with link aggregation across stack members, to ensure rapid recovery of Layer 2 forwarding, this full synchronization of IPv4 and IPv6 unicast and multicast forwarding tables ensures almost no packet loss upon failover.

In addition, non-stop forwarding techniques such as Graceful restart and protocol packet replication ensure that the software routing processes quickly return to an operational state, ready to handle updates from neighbors after a failover.

Furthermore, stack members share state information for other features to facilitate a seamless transition in case of failover.

This applies to the following features:

- AMF
- DHCP
- DHCP Snooping
- LLDP
- VRRP
- Power over Ethernet

Long distance stacking

Most of the Allied Telesis range of stackable devices support long-distance stacking, so that stack members can be kilometers apart. This provides the location resiliency required for effective disaster recovery. Long distance stacking provides a genuine distributed virtual network core. The complete distributed virtual chassis provides a solution with no single point of failure, and a single management entity.

Virtual MAC

When a VCStack is central to network design, the virtual chassis uses a virtual MAC address for communication with other devices. As this single virtual MAC address is used for the complete VCStack, there is no change of MAC address if a new stack member is required to become master. Therefore, there is no need for other devices in the network to learn a new MAC address into their MAC or ARP tables. You must enable virtual MAC to operate VCStack. See "Virtual MAC" on page 74.

Simplified network management

Configuration synchronization

If the startup-config on the master switch is updated, the new startup-config is automatically saved to flash memory on all stack members. Similarly, the new start-up-config is automatically copied to any unit that subsequently joins the stack. This ensures seamless master failover and zero-touch stack unit replacement.

Provisioning for pre-configuration of network devices

Provisioning allows you to pre-configure the switch ports of devices that are not currently physically present. You can configure the ports of an additional VCStack member before it is actually added to the stack. On the physical addition of the unit, the configuration is automatically applied, minimizing network disruption. A VCStack will also retain interface configuration for a device that is removed, facilitating effortless replacement of units.

License distribution

When a feature license is installed on the master switch, it will be copied across to the other stack members.

Management as a single unit

Managing a VCStack is simple, as the stack acts as a single virtual chassis, and can be managed as though it were a single unit.

Occasionally however, you might want to login to a specific member of the stack. Remote login allows you to log into the CLI of another stack member. Consequently, the management interface into the VCStack provides the best of both worlds - the VCStack can be managed as a single unit, or as individual units, depending on which is more convenient for the task at hand.

Extensive statistics

Extensive statistics available from a VCStack virtual chassis provide a wealth of information about data throughput on a per-port, per-resource, or traffic type basis.

The forms of virtual chassis stacking

There are three forms that virtual chassis stacking can take. The three operate internally in essentially the same way, and have essentially the same user management interface. However, each has a specific physical form.

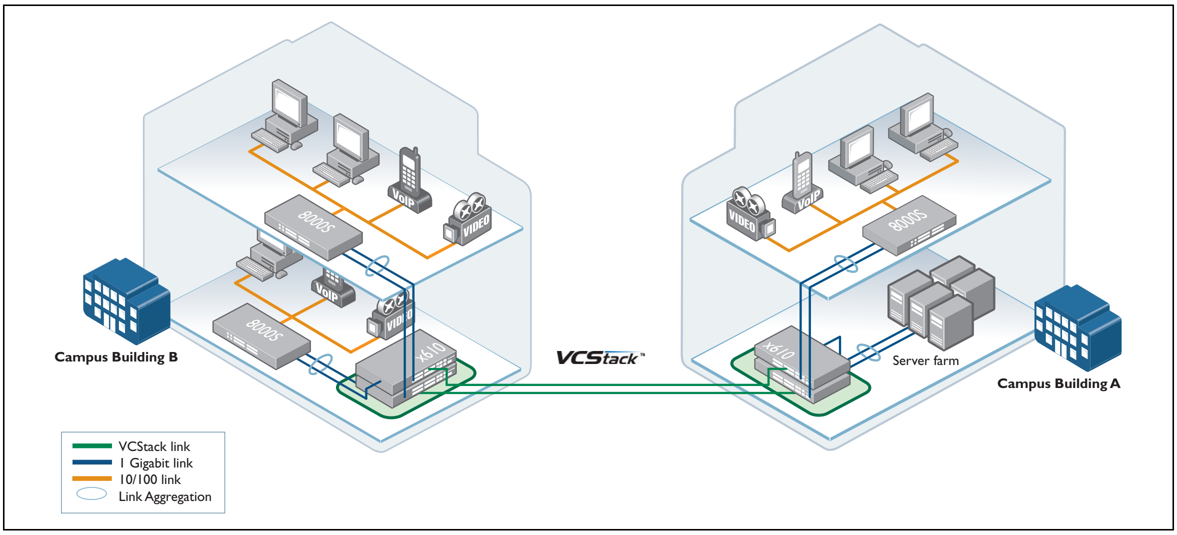
1. Standard Virtual Chassis Stacking – **VCStack** (sometimes referred to as **local** stacking)

   The basic form of virtual chassis stacking consists of a set of pizza box-style switches or mini-chassis stacked together in close physical proximity (typically less than a metre apart). VCStack provides a highly available system where network resources are spread out cross stacked units, reducing the impact if one of the units fails. Aggregating switch ports on different units across the stack provides excellent network resiliency.

2. Long Distance Virtual Chassis Stacking – **LD-VCStack**

   Long distance stacking allows a VCStack to be created over longer distances, perfect for a distributed network environment. The increased distance provided by fiber stacking connectivity means that members of the virtual chassis do not have to be collocated, but can be kilometres apart. All of the benefits and powerful features of VCStack remain exactly the same – Allied Telesis long distance stacking provides a genuine distributed virtual network core.

   There is no restriction on the distance over which long distance stacking can operate. The only limitation on the distance between members of a long distance stack is the distance over which the SFP+ modules can operate.



The network diagram above shows a campus where the VCStack network core is distributed across two separate buildings. By having two stack members in each location, the benefits of using link aggregation between the core and edge switches remain. The complete distributed virtual chassis provides a solution with no single point of failure, and a single management entity.

3. Virtual Chassis Stacking Plus – **VCStack Plus**

   The very high level of resiliency provided by stacking together two SBx8100 chassis is termed VCStack Plus. In this form of stacking, there is component redundancy within each chassis, as well as resiliency between the chassis.

How the stack communicates

The stack communicates through the stacking cables. You can also configure a resiliency link between stack members, which is used when a failover event occurs on the stack. See the section "Configuring the resiliency feature" on page 69 for more information.

One switch controls all switch management on a stack. Which stack member does this is discussed in "The roles of each switch in a stack" on page 10. The software version, startup configuration, and running configuration are exactly the same on each member of a stack. For information on how the stack synchronizes the files, see the section "Software and configuration file synchronisation" on page 63.

The internal communication between stack members is carried out using IP packets sent over the stacking links. This stack management traffic is tagged with a specific VLAN ID and uses IP addresses in a specified subnet.

The default is:

- VLAN 4094
- Subnet 192.168.255.0/28

The management traffic is queued to egress queue 7 on the stack link. The section "Configuring the stack" on page 59 discusses the minor restrictions necessary when configuring VLANs, IP subnets, and QoS on the stack.

The roles of each switch in a stack

Each stack switch serves in one of two roles: either as a **stack member** or as the **stack master** (usually the active master). The stack members are controlled by the stack master. All configuration, including updating the switch software, is set on the stack members by the stack master.

The **stack master** performs a number of tasks:

- It controls all switch management activity.
- It synchronizes boot release and configuration files with stack members.
- It processes all routing protocol packets, and transfers any requisite table updates to the stack members.

A stack member does **not** perform stack master tasks.

The stack master also handles communication on behalf of the stack:

- When you Telnet or SSH to the stack, you connect to a process running on the stack master.
- When you connect to the asyn port on a stack member, this automatically creates an SSH session to the master. This connection will behave as if you were connected to the asyn port on the master.
- The stack master handles SNMP interactions. It gathers MIB statistics from the stack members, and delivers these statistics in response to SNMP Get requests.

You can still execute some specific commands and manage files on an individual switch in the stack. See the section "Monitoring and troubleshooting" on page 78 for more information.

When a VCStack is operating normally, the stack master acts as the **active master**. However, in certain failover situations, when a stack member cannot contact the **active master**, it may act as either a **disabled master** or **fallback master**. See the section "Configuring the resiliency feature" on page 69 for more information about the differences between these stack master roles.

Selecting the active master

The stack members negotiate among themselves as to which switch will become the active master. The election of the active master is based on two criteria:

- each stack member's priority setting
- each stack member's MAC address

For each of these criteria, a lower number indicates a higher priority. The active master is the switch with the **lowest** priority value. If multiple switches share the lowest priority value, then the switch with the lowest MAC address becomes the active master.

Manually changing the active master

You can choose a specific switch to be active master by changing its priority value. By default, a switch has a stack priority value of 128. The command to change a switch’s priority value for stacking is:

```
awplus(config)# stack <stack-ID> priority <0-255>
```

The stack ID is a unique identifier that is assigned to each switch in the stack. See the section "Identifying switches in a stack" on page 12 for more information.

- If a switch is already part of a stack, you can still set the priority value on each switch in the stack through the active master. However, if you set the priority value on a stack member lower than the active master’s priority value, the active master does not immediately relinquish its role as master. The stack member with the lowest priority will take over as active master only if the current active master leaves the stack (this includes any reboot by the stack master).
- If a switch has not yet joined a stack, you can still use the **stack priority** command to change the priority value before connecting it to the stack. However, even if the new stack member has a lower priority value than the active master, it will not take over as active master unless the current active master is removed or rebooted.

Identifying switches in a stack

Each switch in a stack has an ID number, which can be an integer between 1 and 8. The default on each switch is a stack ID of 1.

The ID number of a stack member is an important identifier. All commands that are port or switch specific need to use the stack ID to identify which stack member the commands apply to.

The stack ID is also used to manage specific stack members. When you Telnet or SSH to the stack, your login session terminates on the active master. Similarly, if you connect to the console port of any stack member, your console login session is sent through to the active master. The active master switch is the automatic point of contact for any management session on the stack. If you want to examine something that physically resides on one of the other stack members, such as files in a stack member's flash, or voltage levels on a stack member's power supply, then you can do this by specifying the stack ID of the stack member in commands.

The stack IDs are used frequently in the stack configuration scripts. For example, any command in the configuration script that refers to a physical port will include a stack ID in the port number. The section "Port numbering" on page 59 explains the port numbering scheme used in stacks.

For these reasons, the stack IDs on each switch within a stack are unique, and a switch’s stack ID normally does not change without user intervention.

The only exception to this is when a new switch is connected to a stack and the switch has the same stack ID number as another stack member–the new switch's ID will be automatically renumbered.

Note: There is no connection between stack ID and active master status. The active master switch does not have to be the switch with a stack ID of 1.

Using stack IDs

Most AlliedWare Plus switches display the stack ID number on an 8-segment LED on the front panel. Alternatively, to see the stack ID on a switch before it is connected to a stack, use the command:

```
awplus# show stack
```

The output will show the current stack ID and any pending change to the stack ID.

```
awplus#show stack
Virtual Chassis Stacking summary information

ID  Pending ID  MAC address         Priority  Status   Role
1   -           0000.cd27.c4bf      128       Ready    Active master

Operational Status                  Standalone unit
Stack MAC address                   0000.cd27.c4bf
```

You can also use this command to see the stack numbers on a two-switch stack. To match the physical switch with the right stack ID number, look for the **active master** LED on the front panel. Then use the **show stack** command to show all members of the stack.

You can match the LED status to the **show** command output:

```
awplus#show stack
Virtual Chassis Stacking summary information

ID  Pending ID  MAC address        Priority  Status  Role
1   -           0000.cd27.c4bf     128       Ready   Active master
2   -           0000.cd28.0801     128       Syncing Backup Member

Operational Status                 Normal operation
Stack MAC address                  0000.cd27.c4bf
```

The **show stack** command is available on all stacking switches running the AlliedWare Plus operating system. For details of how this is implemented on the SBx8100, under VCStack Plus, please see "Monitoring and troubleshooting" on page 78

Physically locating a device in a stack

You can use the **find me** command to physically locate a specific device from a group of similar devices. Activating this command causes a selected number of port LEDs to alternately flash green then amber (if that device has amber LEDs) at a rate of 1 Hz.

By default the LEDs will flash for 60 seconds.

```
awplus# findme [interface <port-list>|member <stack-ID>][timeout <duration>]
```

Assigning stack IDs

Stack IDs can be assigned in a number of ways. We recommend only assigning stack IDs when you set up the stack, as a change to stack ID numbers is not automatically reflected in configuration scripts.

You can assign stack IDs to switches before they are part of a stack:

- "Manual assignment on a switch before stacking" on page 14

To assign stack IDs once the stack is created, you can use the following methods:

- "Automatic assignment as a switch joins the stack" on page 15
- "Manual renumbering of a switch after stacking" on page 15
- "Cascade renumbering of the stack" on page 15
- "Renumbering the whole stack using the XEM Select button" on page 100

Manual assignment on a switch before stacking

You can manually assign the stack IDs on switches before you stack them together.

If your switch has never had a stack ID assigned to it, it will have the stack ID of 1. You can assign it a new stack ID with the command:

```
awplus(config)# stack 1 renumber <1-8>
```

If the switch has previously been in a stack and was assigned a stack ID, it keeps that stack ID with it, even if it is removed from the stack. The stack ID has become an inherent property of the switch. If you wish to renumber it you will need to specify the current stack ID.

For example to renumber the stack ID from 3 to 5, use the command:

```
awplus(config)# stack 3 renumber 5
```

The stack ID renumbering does not take effect until the switch is rebooted, so you will receive the following warning as shown in the next figure:

```
awplus(config)#stack 1 renumber 2
Warning: the new ID will not become effective until the stack-member reboots.
Warning: the boot configuration may now be invalid.
```

Until the reboot occurs, the new stack ID value is noted as the 'pending' stack ID, as shown in the next figure:

```
awplus#show stack
Virtual Chassis Stacking summary information

ID   Pending ID  MAC address        Priority  Status   Role
1    2           0000.cd27.c4bf     128       Ready    Active master

Operational Status                  Standalone unit
Stack MAC address                   0000.cd27.c4bf
```

Automatic assignment as a switch joins the stack

When a stack is established, the stack will automatically assign a new stack ID to a switch if it has the same stack ID as another member. This means that you can create a stack without previously changing the stack ID numbers from the default of 1. The stack master will be assigned stack ID 1, and the other switches will be automatically assigned other IDs.

Manual renumbering of a switch after stacking

If you want to manually renumber the switches when they are already part of a stack, use the command:

```
awplus(config)# stack <1-8> renumber <1-8>
```

You can issue this command from the active master to renumber any switch in the stack. To avoid duplicate IDs, a warning message appears if you assign to a stack member the same ID that is currently assigned to another stack member. However, you can continue to renumber the stack member IDs and fix potential ID duplications. Once you have removed any duplicate IDs, you can reboot the switches with ID changes to implement your changes.

If you do not remove the duplications, then when the stack reboots, the switch with the highest stack priority (the lowest priority value) is allocated this ID, and the competing switch is automatically assigned another ID.

Cascade renumbering of the stack

For larger stacks, it is useful to have all the switches numbered in sequence, based on the order of their stack connections. The command that can achieve this is:

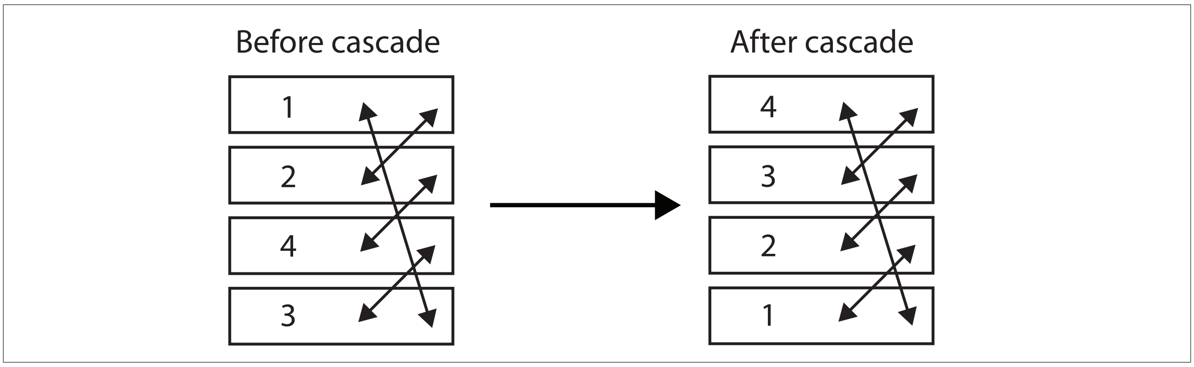
```
awplus(config)# stack <1-8> renumber cascade [<1-8>]
```

For example, you can enter the command:

```
awplus(config)# stack 3 renumber cascade 1
```

This starts the stack numbering on the switch that currently has Stack ID 3 and gives that switch Stack ID 1. The other switches in the stack are renumbered sequentially based on their connection to the switch that now has the stack ID 1.

The following diagram shows the numbering change from this command:



If the second stack ID parameter is not supplied in the command, then the numbering begins from 1. When the numbering process hits the maximum existing stack ID value, it assigns the value 1 to the next switch in the stack.

We recommend using this command to ensure the switches are sequentially numbered, if you did not manually assign the switches with stack IDs before connecting them to the stack. Use the command straight after the stack is first connected.

Caution with stack ID renumbering

Stack IDs are critical in identifying each physical switch in the port numbering and configuration commands. However, the stack's configuration script is not altered when stack IDs are renumbered. This means that you may lose the connection between configuration commands and the physical switches, and commands could end up being applied to a different switch.

For example, if switch A in the stack currently has stack ID 2, then any commands in the configuration script that refer to stack ID 2 are applied to switch A. If the stack IDs are renumbered, so that stack ID 2 is now allocated to switch B, then the commands in the configuration script that refer to stack ID 2 will now be applied to switch B. If switch A is given a stack ID which is not in the configuration script, than it will not have any configuration applied. We recommend that you only renumber the stack when it is initially configured, or during a time of major stack reconfiguration.

Connecting switches into a stack

Different x-Series switches use different cables and connectors to connect to a stack. The types of cables and connections available are dependent on the type of x-Series or CentreCom switches you are stacking.

The stacking options are:

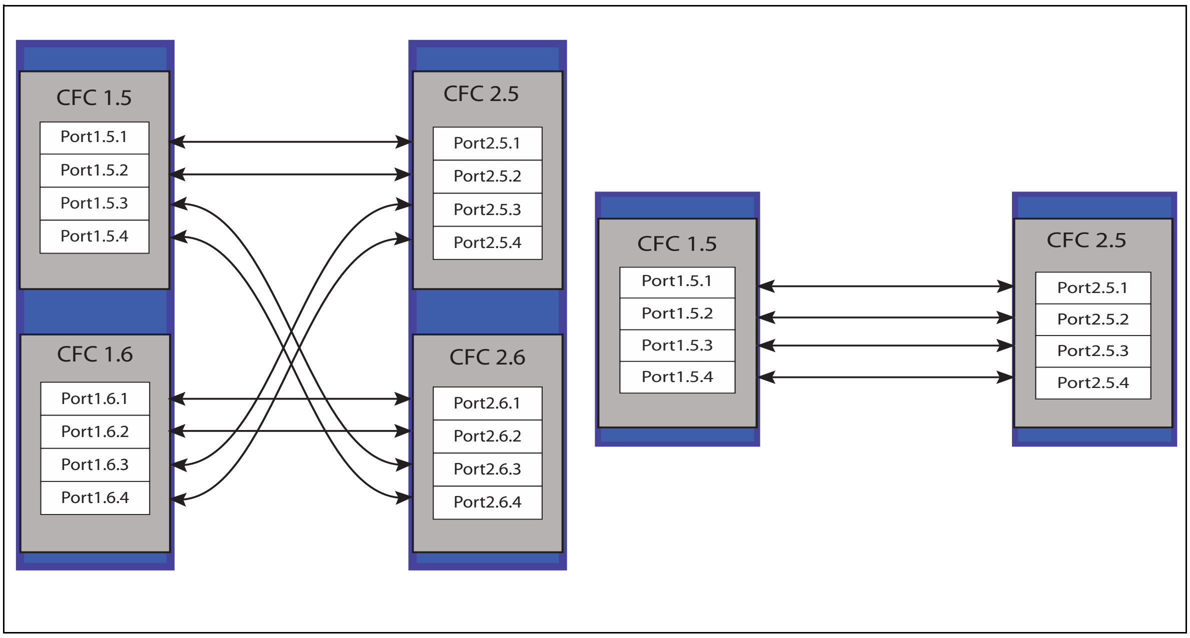
- "Stacking on SwitchBlade x8100 switches" on page 18
- "Stacking on SwitchBlade x908 GEN2 switches" on page 19
- "Stacking on the x950 Series switches" on page 22
- "Front-port and back-port stacking on x930 Series switches" on page 24
- "Stacking on the x330, GS980MX, GS970EMX, x530, and x530L Series switches" on page 26
- "Stacking on the x240 Series switches" on page 28
- "Stacking over 1G ports on the x930, x530, and GS980MX Series switches" on page 29
- "Mixed-mode VCStacking between x530 and x530L Series switches" on page 30
- "Front-port stacking on CentreCOM XS900MX Series switches" on page 31
- "Front-port stacking on CentreCOM FS980M Series switches" on page 32
- "General stacking restrictions by product" on page 44

Stacking on SwitchBlade x8100 switches

This section describes how to connect switches into a stack. Use it along with "VCStack Plus – stacking on the SBx8100" on page 39.

On the SBx8100, you can stack up to three units (78 ports) using HDMI stacking ports or stack up to 24 units using VCStack Plus. VCStack Plus allows you to make software upgrades for multiple switches with a single command, plus update all configurations in a single management session.

- If both SBx8100s have two CFC960 controller cards installed, then the cabling is as shown on the left.
- If both SBx8100s only have a single CFC960 controller card installed, then the cabling is as shown on the right:



Note: Any number of cables can be used to interconnect the CFCs, but reducing the number of cables will mean: reduced bandwidth between VCStack Plus members and reduction in resiliency capabilities.

For more information, see "Monitoring and troubleshooting" on page 78.

Stacking on SwitchBlade x908 GEN2 switches

This section describes how to connect switches into a stack. Use it along with "Connecting switches into a stack" on page 17, which is part of "Steps to set up Virtual Chassis Stacking" on page 33.

On the SwitchBlade x908 GEN2, you can stack up to four switches together[1], either locally or long-distance, using:

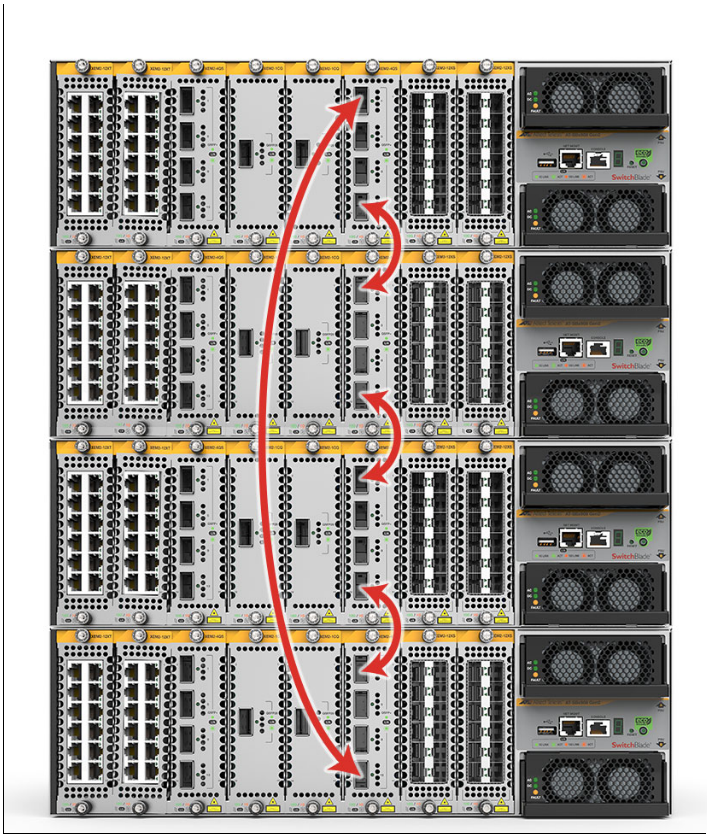
- up to eight 10G ports per switch, using the range of 10G XEM2 modules. This means linking each unit together with up to four cables. You can combine ports from different 10G XEM2 modules; the modules do not have to be the same.
- up to four 40G ports per switch, using the XEM2-4QS. This means linking each unit together with up to two cables.
- two 100G ports per switch, using the XEM2-1CQ. This means linking each unit together with one cable.

You do not have to have all your stacking ports on the same XEM. However, all stack ports must be of the same speed.

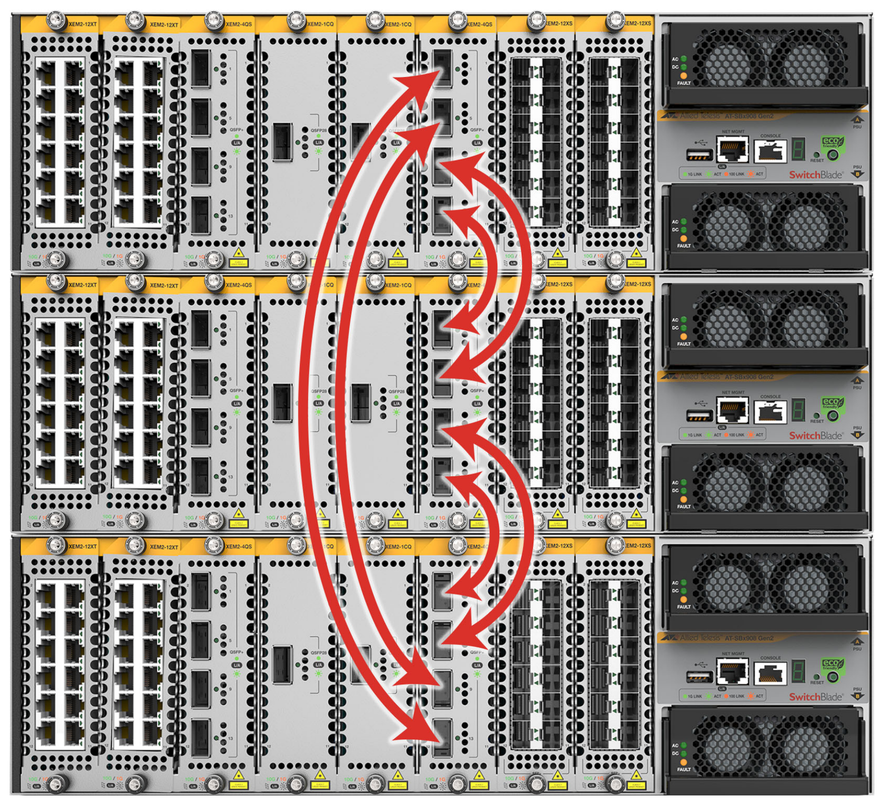
To see how to set up a stack, see "Steps to set up Virtual Chassis Stacking" on page 33.

The following diagram shows how to cable four switches into a stack using one 40G link between each stack member. The stacking cables must form a ring, as shown in the diagram below:

[1] Software version 5.4.7-2.x only supports 2-unit stacking over 40G ports



The following diagram shows how to cable three switches into a stack using two 40G links between each stack member. The stacking cables must form a ring, as shown in the diagram below:



If you use multiple cables to connect the stack members together, please be aware that:

- You need to change all the ports to stacking ports, using the **stackport** command.
- The switch automatically aggregates the stacking ports together. You do not have to create link aggregation groups.
- All stack ports must be of the same speed. You cannot build a stack using some 40G ports and some 10G ports. However, you can combine ports from the different 10G XEM2 modules.
- You must have the same number of stack ports on each link. For example, you cannot have two cables connecting switches 1 and 2 and three cables connecting switches 2 and 3.
- You must connect the switches in a ring, with each switch connected to two stack neighbors (except in a two-unit stack, which can only have one neighbor).

Stacking on the x950 Series switches

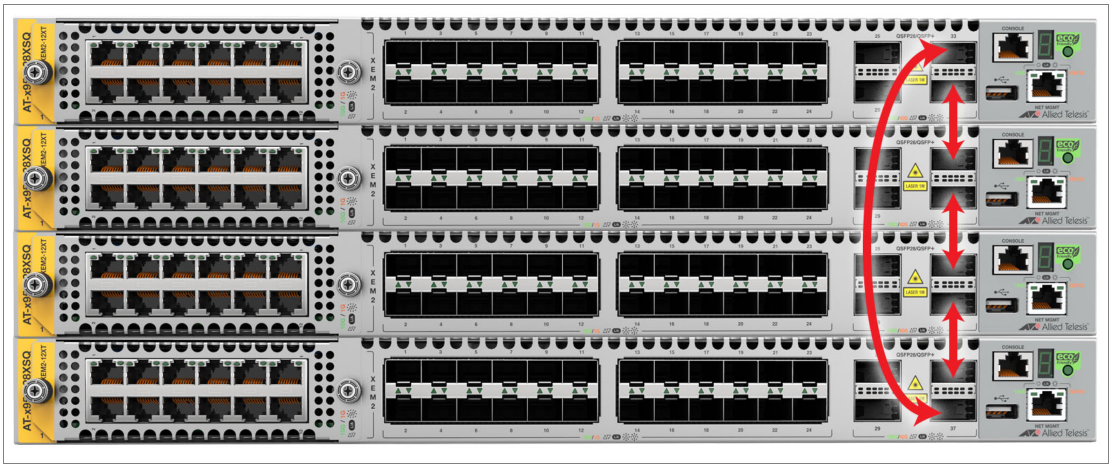
This section describes how to connect switches into a stack. Use it along with "Connecting other switches" on page 36, which is part of "Steps to set up Virtual Chassis Stacking" on page 33.

On the x950 Series switches, you can stack up to eight switches together either locally or long-distance, using:

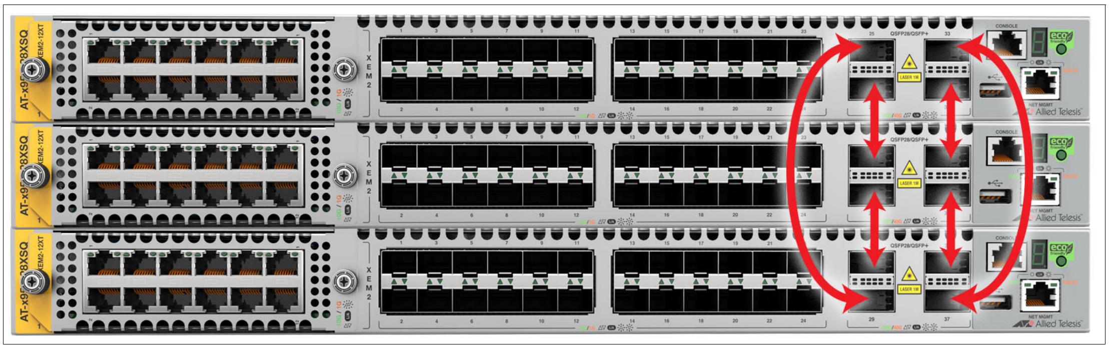
- up to eight 10G ports per switch. This means linking each unit together with up to four cables.
- up to four 40G ports per switch. This means linking each unit together with up to two cables.
- two 100G ports per switch. This means linking each unit together with one cable.

All stack ports must be of the same speed.

To see how to set up a stack, see "Steps to set up Virtual Chassis Stacking" on page 33. The following diagram shows how to cable four switches into a stack using one 40G link between each stack member. The stacking cables must form a ring, as shown in the diagram below:



The following figure shows how to cable three switches into a stack using two 40G links between each stack member.



If you use multiple cables to connect the stack members together, please be aware that:

- You need to change all the ports to stacking ports, using the **stackport** command.
- The switch automatically aggregates the stacking ports together. You do not have to create link aggregation groups.
- All stack ports must be of the same speed. You cannot build a stack using some 40G ports and some 10G ports.
- You must have the same number of stack ports on each link. For example, you cannot have two cables connecting switches 1 and 2 and three cables connecting switches 2 and 3.
- You must connect the switches in a ring, with each switch connected to two stack neighbors (except in a two-unit stack, which can only have one neighbor).

Front-port and back-port stacking on x930 Series switches

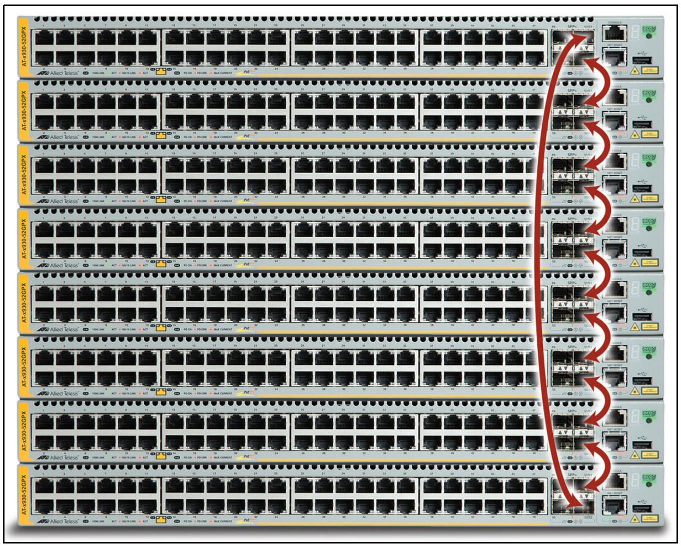
This section describes how to cable a stack together. Use it together with "Connecting other switches" on page 36 in “Steps to set up Virtual Chassis Stacking”.

The x930 Series allow up to 8 units to be stacked using either front or back port stacking. You cannot combine front and back port stacking.

Front-port stacking

Front-port stacking on the x930 Series provides 40Gbps connectivity.

The x930 Series have 4 x 10GbE SFP+ ports, two of which may be used for stacking instead of network connectivity. The stacking cables must form a ring, as shown in the diagram below. The stacking cables must form a ring, as shown in the diagram below:



If the switch does not have a StackQS module installed, then the front-panel ports S1 and S2 will be set up as the stacking ports by default when VCStacking is enabled by the **stack enable** command. If a StackQS module is installed, and you still want to use the front ports for stacking, then you need to explicitly set the front ports as the stacking ports by using the command **stack enable builtin-ports**.

If the S1 and S2 ports are not configured as stacking ports, then they operate as normal network ports. The numbers assigned to these ports are simply the sequential numbers beyond those of the other ports on the switch, i.e. ports 1.0.27 and 1.0.28 on a 28-port switch and ports 1.0.51 and 1.0.52 on a 52-port switch.

Stacking cables

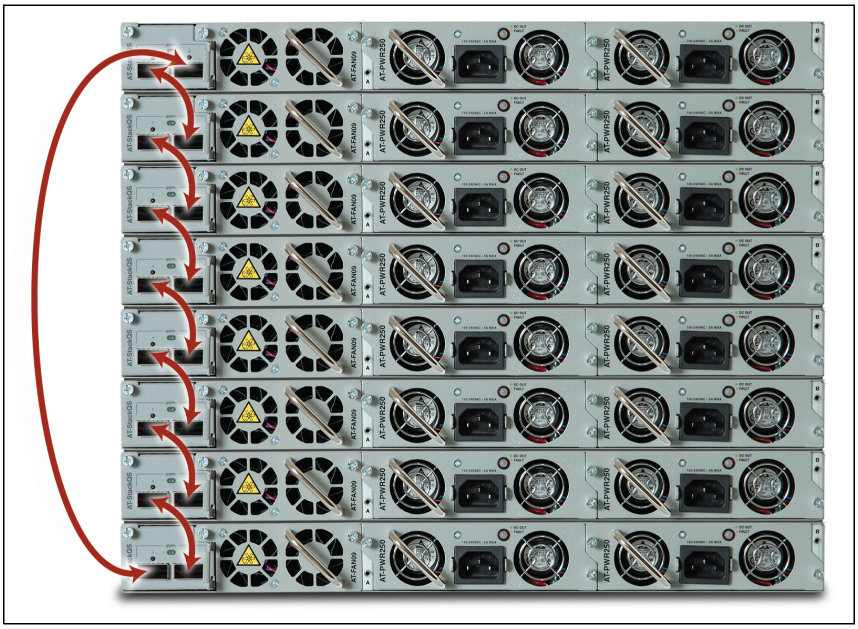
You can use the following cables for x930 Series front-port stacking:

- AT-SP10TW1—1 meter SFP+ direct attach cable
- AT-SP10TW3—3 meter SFP+ direct attach cable
- AT-SP10TW7—7 meter SFP+ direct attach cable

In addition, you can connect Allied Telesis SFP+ modules using cables of whatever length the SFP+ modules can support.

Rear-port stacking

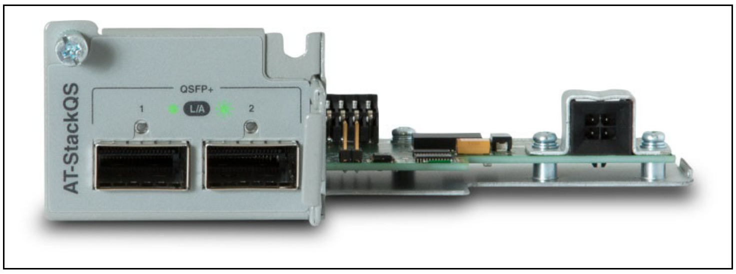
Rear-port stacking on the x930 Series provides 160Gbps connectivity using the StackQS module and direct attached cables. The stacking cables must form a ring, in the same manner as when using front-port stacking. The stacking cables must form a ring, as shown in the diagram below:



Stacking modules and cables

You can use the following module and cables for x930 Series rear-port stacking:

- AT-StackQS: 2 x QSFP+ stacking module
- AT-QSFP1CU: 1 meter QSFP+ direct attach stacking cable
- AT-QSFPSR Transceiver: 150 meter, 12-strand OM4 fiber optic cable
- AT-QSFPLR4 Transceiver (has a duplex LC connector): 2 meter to 10 km single mode fiber (SMF).



By default, if a StackQS module is installed in the switch, then its ports will be chosen as the stacking ports if stacking is enabled:

- If the configuration has been changed so that the front-panel ports are set up as the stacking ports, the role of stacking ports can be changed back to the StackQS ports by using the command **stack enable expansion-ports**.
- If the ports on the AT-StackQS card are not configured as stacking ports, then they operate as 40Gbps network switch ports. The ports are numbered port1.1.1 and port1.1.5.

Stacking on the x330, GS980MX, GS970EMX, x530, and x530L Series switches

This section describes how to cable a stack together. Use it together with "Connecting other switches" on page 36 in “Steps to set up Virtual Chassis Stacking”

It applies to the x330 (excluding x330-10GTX), GS980MX, GS970EMX (excluding GS970EMX/10), x530, and x530L Series switches.

These switches come with two stacking ports which are the last two SFP+ slots on the switches and are labeled **S1/19** and **S2/20** on the 20-port switches, **S1/27** and **S2/28** on the 28-port switches, and **S1/51** and **S2/52** on the 52-port switches.

The ports have two functions.

- You may use them as stacking ports when VCStack is enabled by the **stack enable** command (the default state).
- Or on the other hand, by disabling the VCStack feature, using the **no stack enable** command, you may also use them with regular SFP or SFP+ transceivers as additional networking ports.

Although these ports are labeled S1/19, S1/20, S1/27, S2/28, S1/51 and S2/52 on the front panel, on the CLI they are referred to as port1.0.19, port1.0.20, port1.0.27, port1.0.28, port1.0.51 and port1.0.52.

Front port stacking

If stacking is enabled, all ports except the last two SFP+ slots are network ports by default. However, any network ports can be selected as stackports, but they must first be prepared as follows:

On each switch:

- configure each desired stacking port using the stackport command.
- save the configuration file
- restart

Then you can cable the switches together.

Reconfiguring the default stackports

If the default stackports are not being used for VCStack, they can be set to network ports by using the command **no stackport**. Then save the configuration and reboot.

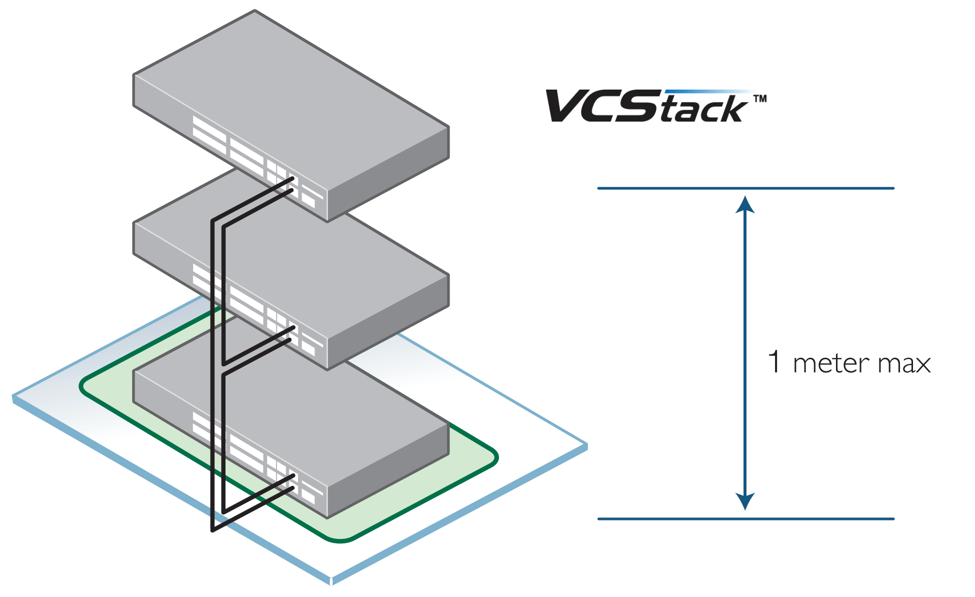
Local stacking using direct attach cables

When creating a VCStack of units that are all located within **a meter** of each other, use local stacking components.

GS980MX, GS970EMX, x330, x530 and x530L Series

- For x330, GS980MX, GS970EMX, x530, and x530L switches running AlliedWare Plus software version 5.4.9 or later, any Allied Telesis Twinax cable will work.

Figure 1: Three direct attach cables form a three-switch VCStack



Long-distance stacking using SFP+ modules

Use long-distance stacking components to create a stack where switches are more than one meter apart.

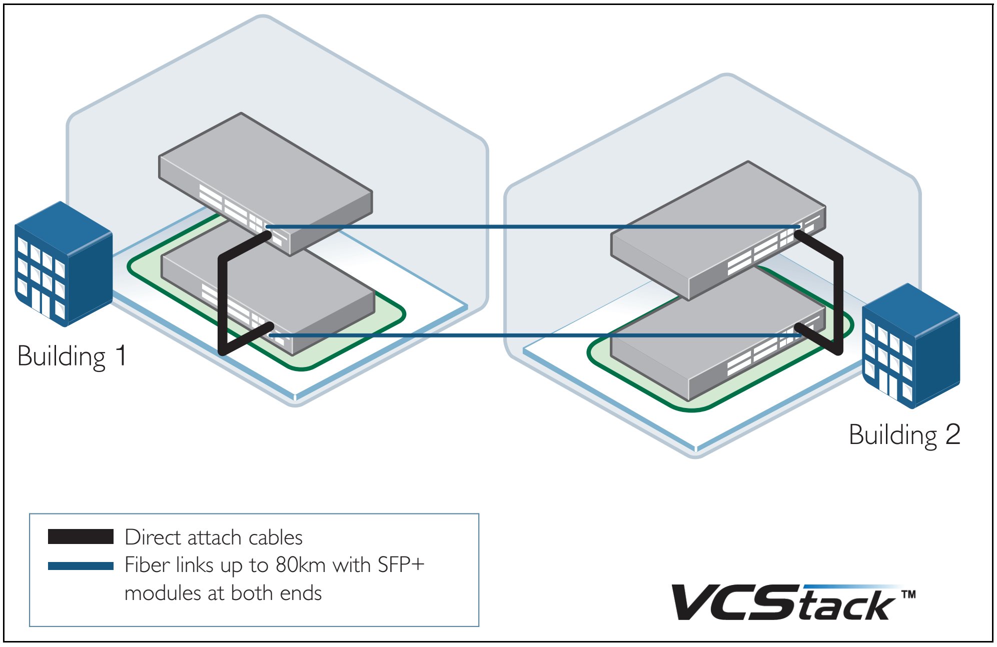
Requirements

GS980MX, x530 and x530L Series

- On switches running AlliedWare Plus 5.4.9 or later, you can use any Allied Telesis SFP+ module.
- Fiber cable.

If Figure 2 on page 28 was configured using long-distance stacking components, then six SFP+ modules and three fiber cables would be required. Note that a mix of local and long-distance stacking components can be used. This is useful if two groups of devices are separated by some distance. Figure 2 below shows this scenario.

Figure 2: Stacking using SFP+ modules



Stacking on the x240 Series switches

This section describes how to cable a stack together. Use it together with "SBx908 GEN2, x550 Series, x240 Series, and XS900MX Series switches" on page 34 in “Steps to set up Virtual Chassis Stacking”.

From software version 5.5.4-1.1 onwards, VCStack on 1G copper ports and SFP ports, is supported on the x240 Series.

You can form a stack with up to two members. Stacking is not enabled by default, therefore you must first enable stacking on the device, then specify which ports are the stacking ports.

Stacking cables

Use the following cables to stack x240 Series switches:

- AT-SP10TW1-1 meter SFP direct attach cable
- AT-SP10TW3 - 3 meter SFP+ direct attach cable

Stacking over 1G ports on the x930, x530, and GS980MX Series switches

This section describes how to cable a stack together. Use it together with "Connecting other switches" on page 36 in “Steps to set up Virtual Chassis Stacking”.

From software version 5.5.1-1.1 onwards, VCStack on 1G copper ports and SFP ports, is supported on the x530 Series (includes x530, x530L, x530DP) and GS980MX Series.

From software version 5.5.1-2.1 onwards, VCStack on 1G copper ports and SFP ports, is supported on the x930 Series.

You can form a stack with up to four members using VCStack on 1G copper and SFP ports, and up to eight 1G ports can be used for stacking between VCStack members.

Note: For any two directly linked stack members, all stackports must originate from a single packet processor and go to a single packet processor. For 52 port models, this means all 1G stackports to another member must be in the port range 1-24 or 25-52 (excluding 41-48 on models with 5G ports in that range).

Mixed-mode VCStacking between x530 and x530L Series switches

This section describes how to cable a stack together. Use it together with "Connecting other switches" on page 36 in "Steps to set up Virtual Chassis Stacking".

Stacking x530s and x530Ls together is supported from software version 5.4.9-0.1. This is known as **mixed-mode** VCStacking.

To use mixed-mode VCStacking, you must have a mixed-mode license installed on each device in the stack. To see the available licenses, check your device's datasheet, which is available at alliedtelesis.com. To purchase the license, contact your Allied Telesis representative. You will need to supply the device serial number.

Note: When in mixed-mode, the table limits (routing limits) on ALL units in the VCStack will be aligned to the base license of the x530L.

Configuring mixed-mode VCStacking

To configure mixed-mode VCStacking, on each unit in the stack you must:

Step 1: Install the mixed-mode stacking license

Once you have downloaded your license, you can transfer it onto the device's flash storage by any preferred method. For example, you can use the **copy** command to copy the file from a USB device to your flash storage.

Step 2. Enter the mixed-mode VCStacking command:

```
awplus(config)# stack mixed-mode
```

Step 3. Save the configuration

Step 4. Re-boot

Adding a unit from the other x530 family

If you want to add a unit from the other family to an existing VCStack, follow steps 1-4 above on both the stack and the unit to be added. The stack cables can be connected prior to rebooting the units.

A note on removing mixed-mode VCStacking

If the license gets removed but the command remains in the configuration file, the different units will NOT be able to form a stack together.

To turn off mixed-mode VCStacking:

Step 1: Enter the command:

```
awplus(config)# no stack mixed-mode
```

Step 2: Save the configuration file

Step 3: Re-boot

Once the command is removed from the configuration file, the license should also be removed but it is not mandatory to do so.

Front-port stacking on CentreCOM XS900MX Series switches

This section describes how to cable a stack together. Use it together with "SBx908 GEN2, x550 Series, x240 Series, and XS900MX Series switches" on page 34 in “Steps to set up Virtual Chassis Stacking”.

The CentreCOM XS900MX Series consists of the following switches: XS916MXS and XS916MXT.

For software version **5.4.6**, the CentreCOM XS900MX Series switches enable you to stack two units together through ports S1 and S2 using SP10TW1-1 meter SFP+ direct attach cables. Connect S1 on one switch to S2 on the other.

From software version **5.4.7** onwards, the CentreCOM XS900MX Series switches allow two units to be stacked using any of their ports. As the units house a mix of copper and fiber ports, this provides flexibility in which ports are used for stacking, and which remain available for network connectivity.

If stacking is enabled, all ports except ports 15 and 16 are network ports by default, and ports 15 and 16 are stacking ports by default. If stacking is disabled, all ports are network ports.

When changing a port to or from a stacking port, you must first configure the desired ports as stacking ports on each switch, using the **stackport** command. Then save the configuration file on each switch, restart both switches, and cable the switches together.

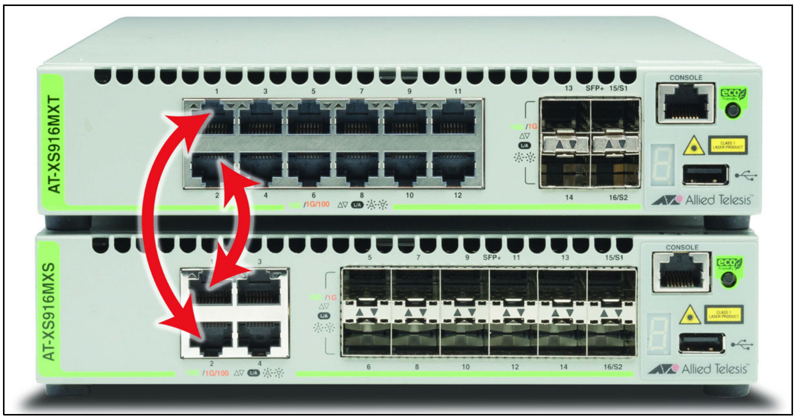
Stacking cables

Use one of the following cables to stack CentreCOM XS900MX Series switches.

To connect:

- SFP+ ports, use SP10TW1-1 meter SFP+ direct attach cables.
- RJ-45 ports, Cat 6a or above cables.

The stacking cables must form a ring, for example as shown in the diagram below, which shows stacking through RJ-45 ports 1 and 2:



Front-port stacking on CentreCOM FS980M Series switches

This section describes how to cable a stack together. Use it together with "Connecting other switches" on page 36 in “Steps to set up Virtual Chassis Stacking”.

VCStacking is supported on the 28 and 52 port models (FS980M/28, FS980M/28PS, FS980M/52, FS980M/52PS).

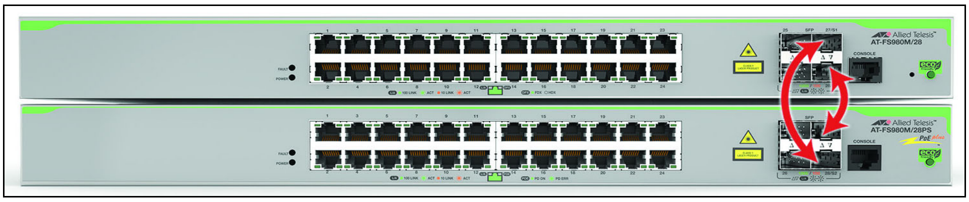
From software release **5.4.7** onwards, the CentreCOM FS980M Series switches allow up to 4 units to be stacked using front-port stacking, providing 4Gbps bandwidth. The CentreCOM FS980M Series have four 1GbE SFP ports, two of which may be used for stacking instead of network connectivity.

Stacking cables

Use the following cable to stack CentreCOM FS980M Series switches:

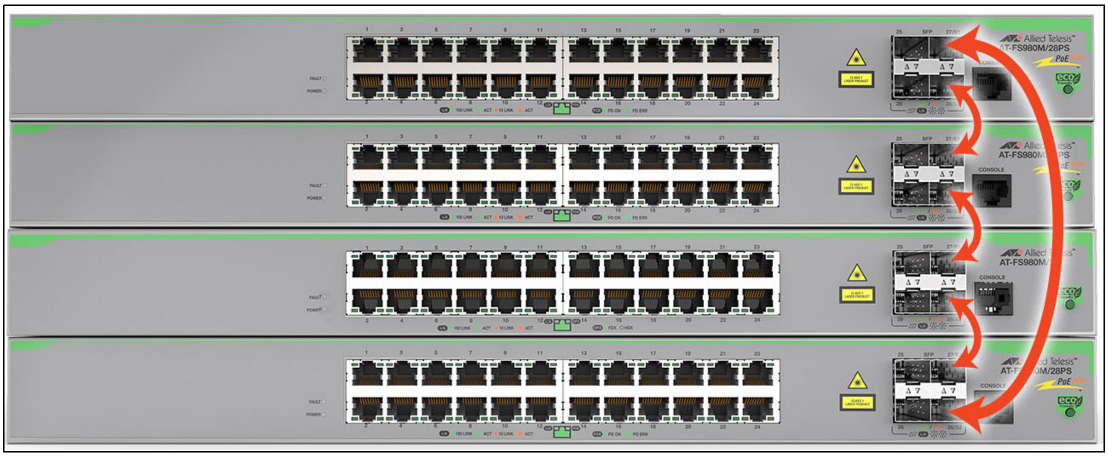
- AT-SP10TW1-1 meter SFP direct attach cable

The stacking cables must form a ring. For example, in a stack of two FS980M/28 switches, stacking port1.0.28 needs to be connected to port2.0.27 and port1.0.27 needs to be connected to port2.0.28, as shown in the diagram below:



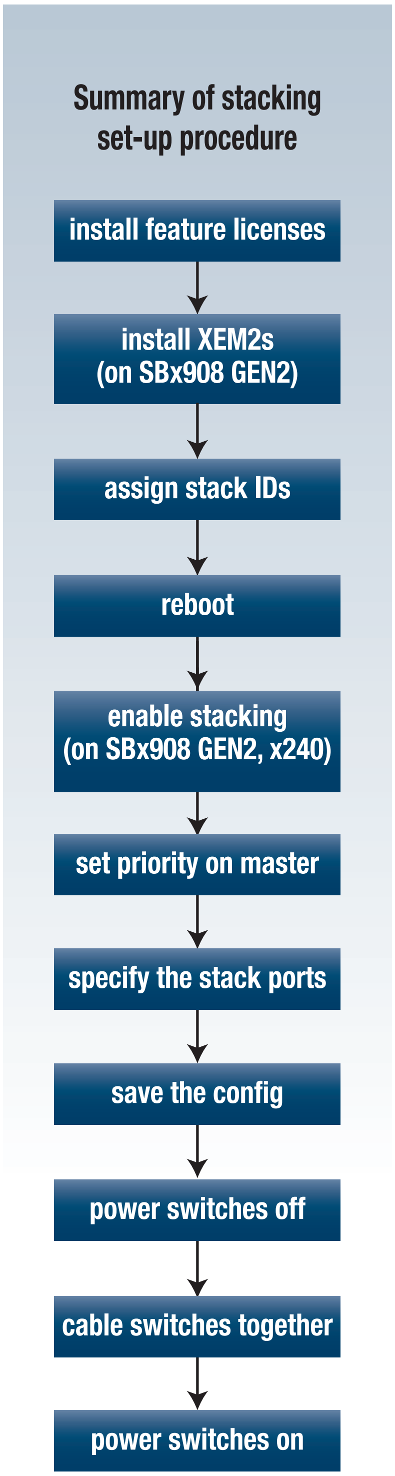
In a stack of four FS980M/28 switches, the cables are connected as follows:

- port1.0.28 <--> port2.0.27
- port2.0.28 <--> port3.0.27
- port3.0.28 <--> port4.0.27
- port4.0.28 <--> port1.0.27



Steps to set up Virtual Chassis Stacking

There are no strict rules for the order in which stack configuration tasks must be completed. However, the following diagram and steps serve as a guideline to ensure a smooth stack creation process:



SBx908 GEN2, x550 Series, x240 Series, and XS900MX Series switches

These steps provide an expansion of the diagram on the previous page.

Step 1: **Ensure that all the switches have the same feature licences installed**

If you have purchased feature licences to enable certain features to operate on the stack, then all stack members need to have the licences installed. If some stack members have licences for features that will not be used on the stack, and other switches do not have those licences, then remove those unnecessary licences.

See the Licensing Feature Overview and Configuration Guide for more information.

Step 2: **If you are stacking SBx908 GEN2 switches, install the XEM2s**

Do this before you enter stacking commands. The switches will reject some stacking commands unless the XEM2s are installed.

- Alternatively, the XEMs can be pre-provisioned prior to being installed by using the following command:

```
awplus(config)# switch 1 bay 3 provision xem2-12
```

Step 3: **Give each of the backup switches the stack ID you want them to have**

The master switch will have a stack ID of 1 by default. Number your backup switches in the order you want the cable them up. The command is:

```
awplus(config)# stack 1 renumber <new-id>
```

This lets you control the stack IDs. Otherwise they are automatically assigned in the order the switches finish booting up.

Step 4: **After setting each switch's stack ID, reboot each switch**

Step 5: **If you are stacking SBx908 GEN2 or x240 Series switches, enable stacking on each switch**

The command is:

```
awplus(config)# stack enable
```

Stacking is enabled by default on x550 and XS900MX Series switches.

Step 6: **On the switch that will be the active master, set the stack priority to 0**

This ensures it becomes the active master. Unless the switch has been stacked before it will have the default stack ID of 1.

The command is:

```
awplus(config)# stack 1 priority 0
```

If you see an error message about stacking not being enabled, you can ignore the message. The switch may display such errors until after you reboot it.

Step 7: Set the stackports on each switch

The commands are:

```
awplus(config)# interface <port-number>
awplus(config-if)# stackport
```

Step 8: Save the configuration

The command is:

```
awplus# write
```

Step 9: Power each switch off

Step 10: Cable the switches together

Before powering any of the switches up, cable them together into the stack. Before cabling the stack, consult "General stacking restrictions by product" on page 44 for information about your product.

Step 11: Power the switches up, starting with the master switch

Turn on each switch, starting with the master switch. It does not matter what order you power the backup switches up in. Note that an SBx908 GEN2 switch may take up to 10 minutes to form a stack.

Step 12: Confirm that the stack is operating

Check that the stack links have all come up successfully with the expected Stack IDs. You can do this by:

- checking the LEDs on the switches or XEMs.
- using the **show stack detail** command. This command provides a snapshot of the stack status.

See the section "Checking stack status" on page 78 for more information about LEDs and the **show stack detail** command.

Step 13: Configure any priority changes

If there is another switch that you want to be the one that takes over as active master, if the active master goes down, then set its priority to a value lower than the other switches:

```
awplus(config)# stack <1-8> priority 10
```

Step 14: Configure the stack as one switch

You are now ready to configure the stack with port channels, VLANs, IP addresses, and so on.

Step 15: Back up the configuration file

Once you are happy with the stack configuration, make a backup copy of the configuration file.

Connecting other switches

This section applies to all stackable AlliedWare Plus switches **except** the SBx8100 Series, SBx908GEN2, x550 Series, x240 Series, and XS900MX Series.

No set rules govern the order in which stack configuration tasks must be carried out. However, these steps offer a guideline to ensure a smooth process for creating the stack:

Step 1: **Prepare the switches**

Before connecting any of the switches together:

- Ensure that all the switches have the same feature licences installed. If you have purchased feature licences to enable certain features to operate on the stack, then all stack members need to have the licences installed. If some stack members have feature licences installed for features that will not be used on the stack, and other switches do not have those licences installed, then remove those unnecessary licences.
- Specify the stacking ports on switches that allow stacking through user-selectable stack ports.
- Decide which stack member will be the active master and set its priority to 0, to ensure it becomes the active master. The command is:

```
awplus(config)# stack 1 priority 0
```

(The switch, if it has not been stacked before, should have the default stack ID of 1.)

Step 2: **Install and power the stack master**

Install and power up the master switch. It will detect that there are no other members in the stack, so it will elect itself master. At bootup, it will output the following messages:

```
13:35:10 awplus-1 VCS[1164]: Member 1 (00-00-cd-28-07-c0) has become the Active
master
```

Then, after the switch has booted, the **show stack** command will show a stack with only one member:

```
awplus#show stack
Virtual Chassis Stacking summary information

ID  Pending ID  MAC address       Priority  Status  Role
1   -           0000.cd27.c4bf    128       Ready   Active master

Operational Status                Standalone unit
Stack MAC address                 0000.cd27.c4bf
```

Step 3: **Install and power the first backup member**

Install the next switch, connecting the stacking cable from that switch to the master. Stack cabling varies between products. Before connecting the stack, consult "General stacking restrictions by product" on page 44 for information about your product.

Power up the switch—it will detect that there is already an active master, and so will come up as a backup member. The active master will assign it the first available stack ID.

At bootup, the new stack member outputs the following messages:

```
Notice: Possible stack, please wait whilst searching for members.
12:26:57 awplus-2 VCS[1043]: Member 1 (0000.cd27.c4bf) has joined stack
12:26:57 awplus-2 VCS[1043]: Please configure 'stack virtual-mac' to minimize
network disruption from failovers
12:26:57 awplus-2 VCS[1043]: Member 1 (0000.cd27.c4bf) has become the Active
master
```

After bootup, the **show stack** command will show that there are two switches in the stack:

```
awplus#show stack
Virtual Chassis Stacking summary information

ID  Pending ID  MAC address        Priority  Status  Role
1   -           0000.cd27.c4bf     128       Ready   Active master
2   -           0000.cd28.0801     128       Syncing Backup Member

Operational Status                 Normal operation
Stack MAC address                  0000.cd27.c4bf
```

The active master will check that the new stack member has the same software version as itself. If the software versions are different, the active master will use the software auto-synchronization mechanism to force the new stack member to run the same software version.

Step 4: Install and power the next backup member

Repeat step 3 for each of the other switches in the stack, following the cabling scheme described in "General stacking restrictions by product" on page 44 for your switch. For the last switch added to the stack, connect this switch to the first installed switch.

Step 5: Confirm that the stack is operating

Check that the stack links have all come up successfully. This can be done by:

- checking the LEDs on the switches.
  The port LEDs for all stack members should be green. Port LEDs that are off or flashing amber indicate that the stack is not operating correctly. The master or status LED will be green on the switch that is the stack master.

- using the **show stack detail** command.
  This command provides a snapshot of the stack status.

See the section "Checking stack status" on page 78 for more information about LEDs and the **show stack detail** command.

Step 6: Check stack numbering

If you are not satisfied with the stack IDs that the switches have been automatically assigned, then this is the right moment to remedy that. To change a stack ID, use the command:

```
awplus(config)# stack <1-8> renumber <1-8>
```

It is usually a good idea to give the stack ID 1 to the active master, as it is quite natural to associate the lowest ID with the active master switch.

To sequentially renumber all stack members, you can use the:

- **select** button on the XEM-STK
- **stack renumber cascade** command
  To renumber the stack members and give stack ID 1 to the active master, use the command:

```
awplus(config)# stack <current ID on master switch> renumber cascade 1
```

Step 7: Reboot the switches

Reboot all the stack members, and check that they all come up with the stack IDs that you are expecting. You can use the command **show stack detail** to check the stack IDs and the status of the neighbor connections.

Step 8: Configure any priority changes

If there is another switch that you want to be the one that takes over as active master, if the active master goes down, then set its priority to a value lower than the other switches:

```
awplus(config)# stack <1-8> priority 10
```

Step 9: Configure the stack as one switch

You are now ready to configure the stack with port channels, VLANs, IP addresses, and so on. For example, to create a port channel that aggregates ports from two different stack members, you would configure as follows:

```
awplus(config)# configure terminal
awplus(config)# interface port1.0.1
awplus(config-if)# channel-group 1 mode active
awplus(config-if)# interface port2.0.1
awplus(config-if)# channel-group 1 mode active
```

Once you are happy with the stack configuration, make a **backup** copy of the configuration file.

VCStack Plus – stacking on the SBx8100

Connecting two SwitchBlade x8100s together can create a single stacked unit, achieving excellent resiliency and simplified management. This is only possible using the CFC960 controller cards, which have 4 x 10 GB SFP+ ports. Because the stacking connections are via fiber SFP+ ports, each SBx8100 switch can be located long distances apart, even kilometers apart if required. The stacking connections must use fiber links, and can be terminated by any Allied Telesis branded SFP+ modules.

For cabling information, see "Stacking on SwitchBlade x8100 switches" on page 18. For information on setting up a resiliency link, see "How to configure the resiliency link feature on a stack" on page 69.

Enabling stacking

Stacking is disabled by default on the SBx8100. When stacking is enabled, with the **stack enable** command, all of the CFC960 SFP+ ports will be reserved for stacking use - they will not be configurable as normal network switchports.

Note: Using the 'stack enable' command automatically enables Virtual-MAC.

Each stack member must have a unique stack ID. This is set with the command:

```
awplus(config)# stack <existing stack-ID> renumber <new stack-ID>
```

The current stack ID of each switch can be seen with the **show stack** command. The SBx8100 must be manually renumbered via the CLI before it is connected in a stack.

Once the stack has formed, the **show stack** command will show the Active CFC.

```
awplus#show stack
Virtual Chassis Stacking summary information

ID   Pending ID  MAC address        Priority  Status  Role
1.5  -           eccd.6db3.586a     128       Ready   Active CFC
1.6  -           eccd.6db3.5919     128       Ready   Backup Member
2.5  -           eccd.6db3.5892     128       Ready   Backup Member
2.6  -           eccd.6db3.58f6     128       Ready   Backup Member

Operational Status                 Normal operation
Stack MAC address                  0000.cd37.0061 (Virtual MAC)
```

- ID 1.5 refers to Stack member 1, the CFC in bay 5
- ID 1.6 refers to Stack member 1, the CFC in bay 6
- ID 2.5 refers to Stack member 2, the CFC in slot 5
- ID 2.6 refers to Stack member 2, the CFC in slot 6

The **show stack detail** command displays a 'Last role change' date/time. Normally if all cards booted up at the same time, the date/time is the same for all cards.

If one card rebooted (or was inserted later), it will have a more recent date/time. If a master (Active CFC) failover occurred, then the stack master will have a more recent date/time than the other backup member CFCs.

```
awplus#show stack detail
Virtual Chassis Stacking detailed information

Stack Status:
------------------------------------------------------------------
Operational Status                      Normal operation
Management VLAN ID                      4094
Management VLAN subnet address          192.168.255.0
Virtual Chassis ID                      97 (0x61)
Virtual MAC address                     0000.cd37.0061
Disabled Master Monitoring              Enabled


Card 1.5:
------------------------------------------------------------------
ID                                      1.5
Pending ID                              -
MAC address                             eccd.6db3.586a
Last role change                        Wed May 4 23:38:46 2022
Product type                            AT-SBx81CFC960
Role                                    Active CFC
Status                                  Ready
Priority                                128
Host name                               awplus-1.5
Resiliency link status                  Not configured
Stack port1.5.1 status                  Learnt neighbor 2.5,connected port2.5.1
Stack port1.5.2 status                  Learnt neighbor 2.6,connected port2.5.2
Stack port1.5.3 status                  Learnt neighbor 2.5,connected port2.6.3
Stack port1.5.4 status                  Learnt neighbor 2.6,connected port2.6.4

Card 1.6:
------------------------------------------------------------------
ID                                      1.6
Pending ID                              -
MAC address                             eccd.6db3.5919
Last role change                        Wed May 4 23:38:47 2022
Product type                            AT-SBx81CFC960
Role                                    Backup Member
Status                                  Ready
Priority                                128
Host name                               awplus-1.6
Resiliency link status                  Not configured
Stack port1.6.1 status                  Learnt neighbor 2.5,connected port2.6.1
Stack port1.6.2 status                  Learnt neighbor 2.6,connected port2.6.2
Stack port1.6.3 status                  Learnt neighbor 2.5,connected port2.5.3
Stack port1.6.4 status                  Learnt neighbor 2.6,connected port2.5.4
```

```
Card 2.5:
------------------------------------------------------------------
ID                                      2.5
Pending ID                              -
MAC address                             eccd.6db3.5892
Last role change                        Wed May 4 23:38:47 2022
Product type                            AT-SBx81CFC960
Role                                    Backup Member
Status                                  Ready
Priority                                128
Host name                               awplus-2.5
Resiliency link status                  Not configured
Stack port2.5.1 status                  Learnt neighbor 1.5,connected port1.5.1
Stack port2.5.2 status                  Learnt neighbor 1.5,connected port1.5.2
Stack port2.5.3 status                  Learnt neighbor 1.6,connected port1.6.3
Stack port2.5.4 status                  Learnt neighbor 1.6,connected port1.6.4

Card 2.6:
------------------------------------------------------------------
ID                                      2.6
Pending ID                              -
MAC address                             eccd.6db3.58f6
Last role change                        Wed May 4 23:38:48 2022
Product type                            AT-SBx81CFC960
Role                                    Backup Member
Status                                  Ready
Priority                                128
Host name                               awplus-2.6
Resiliency link status                  Not configured
Stack port2.6.1 status                  Learnt neighbor 1.6,connected port1.6.1
Stack port2.6.2 status                  Learnt neighbor 1.5,connected port1.6.2
Stack port2.6.3 status                  Learnt neighbor 1.5,connected port1.5.3
Stack port2.6.4 status                  Learnt neighbor 1.6,connected port1.5.4
```

Software release and license requirements

- All CFCs must be running the same software release in order to form a stack. If a CFC is running a different software release, then that CFC will be automatically synchronized to the stack's software release by the VCStack software-auto-sync feature.
- In order to stack, both chassis also need to have their own VCStack-Plus license installed (specific to SBx8100 stacking).
- Both chassis also need to have the same feature licenses installed (this is the same as VCStack on other platforms).
- Under normal circumstances, CFC card 5 should end up the master of each chassis. One of these CFCs will be elected stack master.
- If CFC card 6 becomes master, rather than card 5, then it is likely that card 5 failed to boot or was significantly delayed in booting. There is a limit to how long the switch waits, and it will not change mastership if there is already a master present.

Controller Fabric Card (CFC) operation

When two SBx8100s are stacked together, one CFC card will become the Active CFC for both chassis and the other three CFCs will be in Backup.

In the chassis that does not contain the Active CFC - i.e. just two backup CFCs - one of these backup CFCs becomes the **H/W master** for that chassis. The H/W master relates to how the SBx8100 hardware is designed - only one CFC on the chassis can access the H/W information for that chassis.

In a VCStack Plus stack, the H/W master doesn't have a lot of extra duties to do - the stack master (Active CFC) handles most things. In the backup chassis, the only things the H/W master would do differently to the H/W backup member would be:

- controlling the bootup of the LIFs (Line Interface Fabric/cards) i.e. serving out BOOTP IP and TFTP release.
- environmental monitoring, i.e. temperature/fault sensors in fan tray.
- PoE power allocation (as each chassis has its own power to distribute).

The **Active** CFC will do all of the CPU processing for the stack.

You can infer which CFC is the H/W master in the backup chassis from the highest **uptime** displayed in the **show card** command output.

LIF (Line Interface) operation

All cards communicate with all other cards in the stack, regardless of what chassis they are in, depending on the actual communication.

For example:

- **stackd** communicates with all other cards.
- **ospfd** only communicates with the other CFCs.
- the **BOOTP** client on the LIFs only communicates with the local H/W master.

The insertion or removal of LIF cards in either chassis is controlled by the Active CFC.

The maximum number of supported ports in the SBx8112 stack is **400** switchports in a fully-populated SBx8112 stack with 24 cards in total.

File synchronization

- **VCStack SW auto-sync** ensures the consistent use of the same software release across the stack and automatically upgrades the software if necessary.
- **VCStack file replication** ensures that necessary files from the master are synchronized with the backup member.

If you are combining CFC960v2 and earlier CFC960 cards, you may need to synchronize the AlliedWare Plus software manually. See "Stacking with a mixture of CFC960 v2 and earlier CFC960 in an SBx8100 system" on page 64.

Stack failover

A single CFC960 is elected as the stack master or 'Active CFC' for the entire SBx8100 VCStack. When the Active CFC is removed or rebooted, the backup CFC on the same chassis will take over as master. However, if the backup CFC on the same chassis is not in the equivalent of the VCStack 'Ready' state, then a 'Ready' CFC on the peer chassis will take over as master.

When the entire SBx8100 containing the Active CFC is failed over (e.g. power-cycled), the CFC960 that has H/W mastership on the peer SBx8100 will take over as master. In other words, when an entire SBx8100 chassis is failed, the CFC with the longest system up-time will become the Active CFC.

Failing over any CFC960 in the stack will temporarily halve the available stacking backplane bandwidth. This may cause network disruption if traffic across the stacking backplane exceeds 40Gbps.

CFC failover

When a single Active CFC failover occurs (i.e. the new master CFC is still in the **same** chassis), the interruption to traffic will not be significantly different from CFC failover on a standalone SBx8100.

Chassis failover

When a complete SBx8100 failover occurs (i.e., when the new master CFC is in a different chassis), the interruption to static Layer 2 and Layer 3 unicast traffic forwarding will be less than 3 seconds. The interruption to other protocol traffic, such as multicast, will be best effort, but disruptions may be significantly higher.

Failing over both CFCs in the same chassis at the same time will result in all LIFs in that chassis rebooting as well.

Troubleshooting

The command **show counter stack | grep Master** (note the capital ‘M’) will show you any failovers or conflicts (duplicate masters) that have occurred. Check **show reboot history** and look for any unexpected reboots (note that this will include any times the chassis has been power-cycled). Use the **show stack detail** command to check the 'Last role change' date/time.

General stacking restrictions by product

There are some restrictions to what products and connections you can combine in a single stack. In general:

- switch families cannot be stacked together.
- you cannot combine different stacking methods or cables.

SwitchBlade x908 GEN2

compatible stack partners	A SwitchBlade x908 GEN2 switch can only stack with another SwitchBlade x908 GEN2 switch.
maximum stack size	4 switches (Version 5.4.7-2.x only supports 2-unit stacking over 40G ports).
cable type	For 100G ports: ■ AT-QSFP28 modules For 40G ports: ■ AT-QSFP+ modules ■ AT-QSFP1CU, AT-QSFP3CU: 1m and 3m direct attach cable ■ AT-MTP12-1, AT-MTP12-5: 1m and 5m MTP optical cable for AT-QSFPSR For 10G ports: ■ Allied Telesis 10GbE SFP+ modules ■ AT-SP10TW1, AT-SP10TW3, AT-SP10TW7: 1m, 3m and 7m direct attach cable

SwitchBlade x8100 Series

compatible stack partners	A SwitchBlade x8100 switch can only stack with another SwitchBlade x8100 switch.
maximum stack size	2 switches.
cable type	Fibre: ■ Allied Telesis SFP+ module (two modules required per stacking link).

x950 Series

compatible stack partners	x950 Series switches can only stack with other x950 Series switches. You can create a VCStack of up to eight units: ■ Front port stacking (10G, 40G, 100G) - long distance with any SFP+, QSFP+, QSFP28 module.
maximum stack size	8 switches.
cable type	For 100G ports: ■ AT-QSFP28 modules For 40G ports: ■ AT-QSFP+ modules ■ AT-QSFP1CU, AT-QSFP3CU: 1m and 3m direct attach cable ■ AT-MTP12-1, AT-MTP12-5: 1m and 5m MTP optical cable for AT-QSFPSR For 10G ports: ■ Allied Telesis 10GbE SFP+ modules ■ AT-SP10TW1, AT-SP10TW3, AT-SP10TW7: 1m, 3m, and 7m direct attach cable

x930 Series	**compatible stack partners**	x930 Series switches can only stack with other x930 switches. Within the x930 Series: You can create a VCStack of up to eight units: ■ Front port stacking (40G) - long distance with any SFP+ module. ■ Rear port stacking (160G) - StackQS module,1 meter cables
	maximum stack size	8 switches.
	cable type	■ AT-QSFP1CU (use with AT-StackQS module) - 1 meter QSFP+ direct attach stacking cable ■ AT-SP10TW1, AT-SP10TW3, AT-SP10TW7: 1m, 3m and 7m direct attach cable ■ You can use any Allied Telesis SFP+ module for front port stacking, to get Long Distance Stacking (LD-VCStack)

x550 Series	**compatible stack partners**	x550 Series switches can only stack with other x550 switches. You can create a VCStack of up to four units: ■ Front port stacking (10G, 40G) - long distance with any SFP+ or QSFP+ module.
	maximum stack size	4 switches.
	cable type	Standard Allied Telesis branded Twinax direct attach cables can be used to stack x550 switches. For 40G ports: ■ AT-QSFP+ modules ■ AT-QSFP1CU, AT-QSFP3CU: 1m and 3m direct attach cable ■ AT-MTP12-1, AT-MTP12-5: 1m and 5m MTP optical cable for AT-QSFPSR For 10G ports: ■ Allied Telesis 10GbE SFP+ modules ■ AT-SP10TW1, AT-SP10TW3, AT-SP10TW7: 1m, 3m and 7m direct attach cable

x530 Series	**compatible stack partners**	x530 Series switches can stack with other x530 switches. This means that all the switches in the x530 Series can stack together without restriction. x530 switches can also be stacked with x530L switches, but the whole stack remains at the level of the x530L. This is known as **mixed-mode** stacking. For more detail, see "Mixed-mode VCStacking between x530 and x530L Series switches" on page 30.
	maximum stack size	8 switches.
	cable type	Standard Allied Telesis branded Twinax direct attach cables or SFP+ modules can be used to stack x530 switches. For 10G ports: ■ Allied Telesis 10GbE SFP+ modules ■ AT-SP10TW1, AT-SP10TW3: 1m and 3m direct attach cable
x530L Series	**compatible stack partners**	x530L Series switches can stack with other x530L switches and also with x530 Series switches, but the whole stack remains at the level of the x530L. For more detail, see "Mixed-mode VCStacking between x530 and x530L Series switches" on page 30.
	maximum stack size	8 switches.

cable type	Standard Allied Telesis branded Twinax direct attach cables or SFP+ modules can be used to stack x530L switches. For 10G ports: ■ Allied Telesis 10GbE SFP+ modules AT-SP10TW1, AT-SP10TW3: 1m and 3m direct attach cable.

x330-28GTX and GS970EMX/28

compatible stack partners	x330-28GTX Series switches can only stack with other x330-28GTX switches. GS970EMX/28 Series switches can only stack with other GS970EMX/28 switches.
maximum stack size	6 switches.
cable type	Standard Allied Telesis branded Twinax direct attach cables can be used to stack x330-28GTX switches. For 10G ports: ■ Allied Telesis 10GbE SFP+ modules ■ AT-SP10TW1, AT-SP10TW3, AT-SP10TW7: 1m, 3m and 7m direct attach cable.

x240 Series

compatible stack partners	x240 Series switches can only stack with other x240 switches.
maximum stack size	2 switches.
cable type	Standard Allied Telesis branded Twinax direct attach cables can be used to stack x240 Series switches. For 10G ports: ■ Allied Telesis 10GbE SFP+ modules ■ AT-SP10TW1, AT-SP10TW3: 1m and 3m direct attach cable.

CentreCOM XS900MX Series

compatible stack partners	The CentreCOM XS900MX Series switches can only stack with other CentreCOM XS900MX Series switches.
maximum stack size	2 switches.
cable type	Standard Allied Telesis branded Twinax direct attach cables can be used to stack CentreCOM XS900MX Series switches. For local stacking, use one of the following cables: ■ AT-SP10TW1-1 meter SFP+ direct attach cable. ■ RJ-45 cable - copper cable Cat 6a or above

CentreCOM GS980MX Series

compatible stack partners	CentreCOM GS980MX Series switches can only stack with other CentreCOM GS980MX Series switches.
maximum stack size	4 switches.
cable type	Standard Allied Telesis branded Twinax direct attach cables can be used to stack CentreCOM GS980MX Series switches. For 10G ports: ■ Allied Telesis 10GbE SFP+ modules ■ AT-SP10TW1, AT-SP10TW3, AT-SP10TW7: 1m, 3m and 7m direct attach cable.

CentreCOM FS980M Series

compatible stack partners	CentreCOM FS980M Series switches can only stack with other CentreCOM FS980M Series switches.
maximum stack size	4 switches.
cable type	Standard Allied Telesis branded Twinax direct attach cables can be used to stack CentreCOM FS980M Series switches. For local stacking: ■ AT-SP10TW1—1 meter SFP+ direct attach cable.

Executing commands and managing files on stack members

You can perform certain limited actions on an individual switch member:

- **Executing commands**
  You can use the **reload** command to reboot an individual switch, or use **show** commands to see the individual physical state of a switch and its file system. Show commands that relate to the ports, counters, or configuration are applicable for the stack only.

- **Managing files**
  You can manage files on an **individual** switch. Note that some files are synchronised between switches. See the "Software and configuration file synchronisation" on page 63 section for more information.

  You can also delete files off **all stack members simultaneously**, see "Deleting files from all stack members" on page 49.

Executing commands

Many monitoring commands display information about all stack members, including commands like **show system** and **show system environment**.

For a number of other commands, you can display information from a specific stack member by specifying the stack ID number in the syntax. For example, you can run the following commands on the Master to see information about a specific stack member:

```
awplus# show cpu <stack-ID>

awplus# show cpu history <stack-ID>

awplus# show memory <stack-ID>

awplus# show memory history <stack-ID>

awplus# show process <stack-ID>
```

Additionally, the **reload** command can take a stack ID as an extra parameter. To reboot just the stack member with stack ID 4, use the command:

```
awplus# reload stack-member 4
```

You will get the following question from the stack:

```
stack#reload stack-member 4
reboot stack member 4 system? (y/n):
```

Managing files

If you wish to perform an action on another stack member's file system, use the syntax:

```
awplus# <stack-member-name>/flash:[/]<filename>
```

The ***stack-member-name*** is not the stack ID, but is an extended hostname formed as:

```
awplus# <hostname>-<stack-ID>
```

If the hostname of the stack is **BlueCore**, then the ***stack-member-name*** for switch 2 in the stack is:

```
BlueCore-2
```

If you do not use the ***stack-member-name*** prefix, then the command refers to a file that resides on the stack master.

You can also use the ***stack-member-name*** syntax for the directory command. To view the contents of the flash file system on a specific stack member, you can use the syntax:

```
awplus# dir <stack-member-name>/flash:/
```

Displaying files on all stack members

From 5.4.8-0.1 onwards, you can display files on all stack members with one command. To do this, use the command:

```
awplus# dir stack-wide [all] [recursive] [sort [reverse] {name|size|time}]
[<filename>|flash|nvs|card|usb|debug]
```

This new **dir stack-wide** command behaves the same as the existing **dir** command, except for running on all stack members.

Examples

To show files in the current directory across all stack members, use the command:

```
awplus# dir stack-wide
```

To show files in the root flash directory across all stack members, use the command:

```
awplus# dir stack-wide flash
```

To show all files recursively in the root flash directory across all stack members, use the command:

```
awplus# dir stack-wide all recursive flash
```

Deleting files from all stack members

If you wish to remove file(s) from all stack members simultaneously, use the following command:

```
awplus# delete stack-wide force [recursive] <name>
```

where:

- `recursive`—deletes directories that match the name, including their contents
- *<name>*–the name of the files or directories to delete.

The filename can include the wildcard *

Note: Use the wildcard with caution, because this command is non-interactive and does not ask for confirmation before deleting files. This is indicated by the mandatory **force** parameter.

You can use this command within an AMF working set.

For example, to delete a file that is located in flash memory on all stack members, use the following command:

```
awplus# delete stack-wide force test.scp
```

To remove directories ouput1 and output2 from an external card on all stack members, use the command:

```
awplus# delete stack-wide force recursive card:output*
```

Remote login

Occasionally it can be desirable to login to a specific member of the stack (for example, to manage feature licences per individual unit). Remote login facilitates this by allowing a user on the master switch to log into the CLI of another stack member.

In most respects, once you are logged into the other stack member, the result of entering commands will be is as if you were logged into the stack master, i.e. the command **show ip interface** will show all IP interfaces configured on all switches in the stack - not just those on the stack member that you have connected to with the **remote-login** command. Configuration commands are still broadcast to all stack members.

Some **show** commands that display physical attributes of the switch, and commands that access the file system, are executed locally. Also, commands related to feature licences are executed locally.

Note: Remote-login is only supported for users that exist in the local user database. I.e. If a stack is configured to authenticate user login via remote RADIUS server and the current user is not present in the local user database, then remote-login will fail.

To login from the stack master (stack member 1 in this case) to stack member 2:

```
awplus#remote-login ?
  <1-8>  A specific stack member ID

awplus#remote-login 2
Type 'exit' to return to awplus.

AlliedWare Plus (TM) 5.3.4 05/04/10 11:59:17

awplus-2>en
awplus-2#
```

Notice that the prompt has changed to reflect the stack member (2) that we are currently connected to.

A directory listing will now show the files on stack member 2 only:

```
awplus-2#dir *.cfg
     948 -rw- May  4 2015 20:59:48  flash:/default.cfg
     677 -rw- May  3 2015 18:39:04  flash:/zz.cfg
     2944 -rw- Mar 23 2015 12:55:40  flash:/ospfv3.cfg
```

We can delete a file from stack member 2 as if we were directly connected to it:

```
awplus-2#del zz.cfg
Delete flash:/zz.cfg? (y/n)[n]:y
Deleting..
Successful operation
awplus-2#
```

To return to the stack master, use the **exit** command:

```
awplus-2#exit
awplus#
```

Factory resetting and removing a switch from a stack

We recommend that when a device is permanently removed from a stack to be redeployed elsewhere, that the switch is first returned to factory defaults before being reconfigured for its new purpose.

When a device has been a member of a stack there are certain internal settings which could cause conflicts if the device is reconfigured and deployed elsewhere in the network without first returning the device to factory defaults. This is especially important if Vista Manager EX and device discovery using STOAT is used in the network.

If you want to remove a switch from a stack, use the following steps:

1. Shut down all stacking links to the switch you wish to remove from the stack. This is optional but recommended, especially if someone other than you will do the step of physically uncabling the switch.
2. Remove all of the cabling used for stacking links.
3. Re-cable the remaining stack members together correctly.
4. Log into the disconnected device through its console port. If you use SSH to connect, you will lose connectivity to the switch after resetting it.
5. Reset the switch to its factory default state, using either of the commands:

   ```
   awplus# atmf cleanup
   ```

   or

   ```
   awplus# erase factory-default
   ```

   These commands are synonyms and will erase all NVS, all flash contents except for the boot release, a GUI resource file, and any license files.
6. Reboot the switch.
7. If you want to use the switch as a standalone switch, disable stacking using the command **no stack 1 enable** and reboot the device.

NOTES:
If the full factory-default reset is inconvenient, it is important to change or delete the '`stack virtual-chassis-id`' in the startup configuration.

For devices running software version 5.5.3-0.x or later, reset the device unique-id using the privileged exec command **unique-id** to avoid serious network conflicts.

Steps to replace a stack member

If you need to replace a stack member, use the following steps to achieve a smooth transition:

Step 1: Configure the Stack ID on the replacement switch

Configure the replacement switch with the same stack ID as the one being replaced.

```
awplus(config)# stack <current-stack-ID> renumber <1-8>
```

Step 2: Reboot the replacement switch

Use the command:

```
awplus# reboot
```

Step 3: Configure the feature licences

Ensure that the replacement switch has the same set of feature licences as the existing stack members.

Step 4: On an SBx908 GEN2 switch, enable stacking

On an SBx908 GEN2 switch, you need to enable stacking.

Use the command:

```
awplus(config)# stack enable
```

Step 5: On an XS900MX, x950, x550, or SBx908 GEN2 switch, configure the stack ports

Configure the required ports as stack ports. For example, if ports 3 and 4 are the stack ports, use the commands:

```
awplus(config)# interface port2.0.3-4
awplus(config-if)# stackport
```

If replacing an XS900MX switch, only do this step if you are not using the default ports 15 and 16 for stacking.

Step 6: Save the configuration

Use the command:

```
awplus# write
```

Step 7: Remove the failed switch

Unplug the failed switch from the stack.

Step 8: **Install the replacement switch**

- Connect the stacking cables to the replacement switch and power it up. It will detect the active master and come up as a stack member.
- The active master will check that the new stack member has the same software version as itself. If the software versions are different, the active master will use the software auto-synchronization mechanism to force the new stack member to run the same software version.
- The new switch, configured with the same stack ID as the replaced one, will receive the identical startup configuration and operate identically.

Provisioning

Provisioning allows pre-configuration of ports for future switches or modules, whether for stack expansion or hot-swapping XEM2 modules.

With provisioning, you can configure stack members and their ports even if they are not currently physically present, preparing them for future use. This means you can either pre-configure ports for a bay or switch that has not yet been installed or load a configuration that references these ports.

Provisioning also automatically tracks the configuration that was present on XEM2 modules that have been hot-swapped out of a switch or on units that have been removed from a stack. It maintains a 'placeholder' for XEM2 modules or switches that have been hot-swapped out.

If you provision a switch or bay and later decide to change the stack member ID or bay number before it is installed, you must first unprovision the switch or bay using the command **no switch <stack ID> bay/switch**.

Provisioning a bay

With the **show system** command we can see that the stack member 2, the SBx908 GEN2 switch, does not have a XEM2 in bay 2:

```
awplus#show sys
System Status                                              Thu Jun 22 11:50:18 2023

Stack member 1

Board       ID   Bay     Board Name                         Rev   Serial number
-----------------------------------------------------------------------------
Base        483  Base    AT-SBx908 GEN2                     B-0   A10064A190100022
Expansion   488  Bay1    AT-XEM2-12XT                       X4-0  A05385A171600074
Expansion   545  Bay2    AT-XEM2-12XTm                      C-0   A10109A202800012
Expansion   487  Bay3    AT-XEM2-12XS                       A-0   A10067A181600033
Expansion   545  Bay5    AT-XEM2-12XTm                      B-0   A10109A180800030
Expansion   559  Bay8    AT-XEM2-12XS v2                    A-0   A10179A191500022
PSU         451  PSUA    AT-SBxPWRSYS2/AC                   A-2   6A000250
Fan module  485  FANA    AT-FAN08                            -     -
Fan module  485  FANB    AT-FAN08                            -     -
-----------------------------------------------------------------------------
RAM: Total: 3985112 kB Free: 3369828 kB
Flash: 3.6GB Used: 55.6MB Available: 3.6GB
-----------------------------------------------------------------------------
Environment Status : Normal
Uptime             : 28 days 01:17:51
Bootloader version : 6.2.7


Stack member 2

Board       ID   Bay     Board Name                         Rev   Serial number
------------------------------------------------------------------------------
Base        483  Base    AT-SBx908 GEN2                     X6-0  A10064A172000017
Expansion   488  Bay1    AT-XEM2-12XT                       X4-0  A05385A171600087
Expansion   488  Bay3    AT-XEM2-12XT                       B-0   A10068A190200010
Expansion   545  Bay5    AT-XEM2-12XTm                      B-0   A10109A180800024
Expansion   487  Bay8    AT-XEM2-12XS                       A-0   A10067A172800031
PSU         319  PSUA    AT-SBxPWRSYS1/AC                   A-0   A04580A141300029
Fan module  485  FANA    AT-FAN08                            -     -
Fan module  485  FANB    AT-FAN08                            -     -
------------------------------------------------------------------------------
RAM: Total: 3985112 kB Free: 3394600 kB
Flash: 3.6GB Used: 53.1MB Available: 3.6GB
------------------------------------------------------------------------------
Environment Status : Normal
Uptime             : 28 days 00:55:15
Bootloader version : 6.2.24

We can see that Stack member 1 is the master, and we are connected to the console
port on this switch:

awplus#show stack
Virtual Chassis Stacking summary information

ID   Pending ID  MAC address       Priority  Status  Role
1    -           e01a.ea43.4252    100       Ready   Active Master
2    -           e01a.ea2a.4f4d    110       Ready   Backup Member

Operational Status                 Normal operation
Stack MAC address                  0000.cd37.08e1 (Virtual MAC)
```

On the Stack master (stack member 1) we can provision a XEM2-12 for Stack member 2 in bay 2 (which is currently empty):

Note that the switch automatically provisions all currently installed switches and XEM2 modules as it boots up.

```
awplus(config)#switch 2 bay 2 provision xem2-12

awplus#show run
...
switch 1 provision sbx908gen2
switch 1 bay 1 provision xem2-12
switch 1 bay 2 provision xem2-12
switch 1 bay 3 provision xem2-12
switch 1 bay 4 provision xem2-12
switch 1 bay 5 provision xem2-12
switch 1 bay 8 provision xem2-12
switch 2 provision sbx908gen2
switch 2 bay 1 provision xem2-12
switch 2 bay 2 provision xem2-12
switch 2 bay 3 provision xem2-12
switch 2 bay 4 provision xem2-12
switch 2 bay 5 provision xem2-12
switch 2 bay 8 provision xem2-12
...
!
interface port2.2.1-2.2.12
 switchport
 switchport mode access
!
```

We can see above that we now have ports 2.2.1-2.2.12 available for configuration in the running-config, even though stack member 2 does not actually have a 12 port XEM2 (XEM2-12) physically installed in bay 2 yet.

This means that these ports can now be configured, ready for when the XEM2-12 is installed.

Commands can refer to ports on that provisioned XEM2 as though it were already present:

```
awplus(config)# int port2.2.1
awplus(config-if)# switchport access vlan 2
```

Once a XEM2 is hot-swapped into bay 2, the switch 2 bay 2 provision XEM2-12 will still show in the running configuration, along with the other installed switches and XEM2 modules.

If the XEM2 is removed, the provisioning for it will remain along with the configuration for the ports associated with it.

What happens when a provisioned XEM2 gets hot-swapped out?

In the example below, stack member 1 has an XEM2-4QS installed in bay 1 and its port (port1.1.1) has been configured as a trunk:

```
switch 1 provision sbx908gen2
switch 1 bay 1 provision xem2-4
switch 2 provision sbx908gen2
!
interface port1.1.1
 switchport
 switchport mode trunk
 switchport trunk allowed vlan allowed
 switchport trunk native vlan noneswitchport trunk native vlan none
!
```

If the AT-XEM2-4QS is hot-swapped out of bay 1:

```
awplus#22:24:42 awplus Platform: HOTSWAP stack member 1 Bay1: Expansion - AT-
XEM2-4QS: removed
22:24:44 awplus NSM[1222]: Removal event on bay 1.1 has been completed
```

We can see that the configuration associated with this port is still in the running configuration:

```
interface port1.1.1
 switchport
 switchport mode trunk
 switchport trunk allowed vlan allowed
 switchport trunk native vlan none
```

What happens when the XEM2 is hot-swapped back in?

```
awplus#22:26:19 awplus Platform: HOTSWAP stack member 1 Bay1: Expansion - AT-
XEM2-4QS: inserted
22:26:19 awplus VCS[1164]: AT-XEM2-4QS has been inserted into bay 1.1


22:26:25 awplus IMI[1245]: All users returned to config mode while switch
synchronization is in progress.
22:26:26 awplus NSM[1222]: Insertion event on bay 1.1 has been completed

22:26:28 awplus IMI[1245]: Configuration update completed for port1.1.1-
port1.1.13
```

We can see above that ports1.1.1-1.1.13 have had their configuration updated from the running configuration.

What happens if a different type of XEM2 is hot-swapped in?

If the XEM2-4QS is hot-swapped out and a different type of XEM2 (in this case a AT-XEM2-12XTm) is hot-swapped into bay 1 instead:

```
22:31:39 awplus Platform: HOTSWAP stack member 1 Bay1: Expansion - AT-XEM2-4QS:
removed
22:31:42 awplus NSM[1222]: Removal event on bay 1.1 has been completed

22:31:44 awplus Platform: HOTSWAP stack member 1 Bay1: Expansion - AT-XEM2-12XT:
inserted
22:31:44 awplus VCS[1164]: AT-XEM2-12XT has been inserted into bay 1.1

22:32:12 awplus IMI[1245]: All users returned to config mode while switch
synchronization is in progress.
22:32:14 awplus NSM[1222]: Insertion event on bay 1.1 has been completed

22:32:16 awplus NSM[1222]: Mismatch in board class of unit 1.1: provisioned
(xem2-4) actual (xem2-12)
22:32:16 awplus IMI[1245]: Configuration update completed for port1.1.1-
port1.1.12
```

We can see that the system has detected a provisioning mis-match and provisioning has been modified to reflect the actual hardware installed. The running configuration now has ports1.1.1-1.1.12, which are the 12 ports belonging to the AT-XEM2-12XTm in bay 1.

```
switch 1 provision sbx908gen2
switch 1 bay 1 provision xem2-12
switch 1 provision sbx908gen2
!
interface port1.1.1-1.1.12
 switchport
 switchport mode access
```

Provisioning a switch

For this example, we have a standalone switch that we will be adding another switch to in the future to form a stack:

```
awplus#sh sys
System Status                                  Wed Sep 13 18:48:40 2023

Board     ID  Bay    Board Name                  Rev   Serial number
-----------------------------------------------------------------------------
Base     389  Base   AT-x930-28GSTX              A-0   A04881A151300020
Expansion 418  Bay1   AT-StackQS                  B-0   A04882A151200032
PSU       335  PSU1   PWR800                      X1-0  SN005
-----------------------------------------------------------------------------
RAM: Total: 2009156 kB Free: 1642200 kB
Flash: 253.8MB Used: 214.4MB Available: 39.3MB
-----------------------------------------------------------------------------
```

The current switch has an ID (stack member) of 2.

```
awplus#show stack
Virtual Chassis Stacking summary information

ID   Pending ID  MAC address        Priority  Status  Role
2    -           eccd.6def.f75b     128       Ready   Active master

Operational Status                 Standalone unit
Stack MAC address                  eccd.6def.f75b
```

We can provision stack member 1 so that we can configure the future stack member's ports before we actually have the second switch connected:

```
awplus(config)#switch 1 provision ?
 dc2552   Provision an DC2552 switch
 gs970emx-20 Provision a GS970EMX-20 switch
 gs970emx-28 Provision a GS970EMX-28 switch
 gs970emx-52 Provision a GS970EMX-52 switch
 gs980mx-10  Provision a GS980MX-10 switch
 gs980mx-18  Provision a GS980MX-18 switch
 gs980mx-28  Provision a GS980MX-28 switch
 gs980mx-52  Provision a GS980MX-52 switch
 ie560-12  Provision an IE560-12 switch
 sbx908gen2  Provision an SBx908 GEN2 switch
 sbx908gen3  Provision a sbx908gen3
 se540l-28  Provision a SE540L-28 switch
 x240-10  Provision an x240-10 switch
 x240-26  Provision an x240-26 switch
 x250-18  Provision an x250-18 switch
 x250-28  Provision an x250-28 switch
 x330-20  Provision an x330-20 switch
 x330-28  Provision an x330-28 switch
 x330-52  Provision an x330-52 switch
 x530-10  Provision an x530-10 switch
 x530-18  Provision an x530-18 switch
 x530-28  Provision an x530-28 switch
  x530-52      Provision an x530-52 switch
  x530L-10     Provision an x530L-10 switch
  x530L-18     Provision an x530L-18 switch
  x530L-28     Provision an x530L-28 switch
  x530L-52     Provision an x530L-52 switch
  x540L-28     Provision a x540L-28 switch
  x550-18      Provision an x550-18 switch
  x908         Provision an x908 switch
  x930-28      Provision an x930-28 switch
  x930-52      Provision an x930-52 switch
  x950-28      Provision an x950-28 switch
  x950-52      Provision an x950-52 switch
  xs900-16     Provision an XS900-16 switch
```

Select the model of switch that will be connected in the future (note that you can only stack, and therefore provision, switches of the same basic model).

If you try to provision an x530-28 switch for stack member 1, and the existing switch (stack member 2) is an x930 VCStack, you will get an error message similar to this one:

```
awplus(config)#switch 1 provision x530-28
% Board class x530-28 is incompatible with this system
```

You can successfully provision an x930-28 as follows:

```
awplus(config)#switch 1 provision x930-28
```

The running-config shows that we can now configure the ports (1.0.1-1.0.28) on provisioned stack member 1:

```
switch 1 provision x930-28
switch 2 provision x930-28
!
interface port1.0.1-1.0.28
switchport
switchport mode access
!
```

Configuring the stack

The configuration script on a VCStack is shared between all stack members. When configuring the stack, you will need to be aware that the stack uses a specific port numbering scheme and that there are some minor restrictions to VLAN, IP subnet, and QoS configuration. There are also specific triggers available for stacking that you may wish to use.

See the following sections for more detail:

- "Port numbering" on page 59
- "VLAN and IP subnet restrictions" on page 60
- "Quality of Service (QoS) restriction" on page 60
- "Stacking triggers" on page 61

For information about how the stack synchronizes the startup and running configuration between stack members, see the section "Software and configuration file synchronisation" on page 63.

Port numbering

In the AlliedWare Plus operating system, switch port interfaces are always referenced as portx.y.z. The first number, or 'x', in the three number port identifier is the Stack ID.

For a stand-alone switch that has never been in a stack, the ports are always numbered 1.y.z. When configuring a stack, however, there will be stack members with other stack ID values. To configure ports on these switches, use the stack ID to refer to each physical switch, for example 2.0.1, 2.0.2, 2.0.3, for the first 3 ports on a switch with stack ID 2.

Please note that when a switch is removed from a stack, it maintains its stack ID number. If it is configured as a stand-alone switch, you will need to either change the stack ID back to 1, or use the current stack ID when configuring its ports.

VLAN and IP subnet restrictions

Internal communication between the stack members is carried out using IP packets sent over the stacking links. This stack management traffic is tagged with a specific ID and uses IP addresses in a specified subnet.

By default, the VLAN and subnet used are:

- VLAN 4094
- Subnet 192.168.255.0/27

You may need to change these values if they clash with a VLAN ID or subnet that is already in use in the network.

Use the commands:

```
awplus(config)# stack management subnet <ip-address>
awplus(config)# stack management vlan <2-4094>
```

It is important that the settings for **management subnet** and **management vlan** are the same for all the switches in a stack. If a switch is added to a stack, and its setting for **management vlan** and/or **management subnet** differ from those on the other stack members, the new switch will not be joined to the stack and a log message similar to the following will be created:

```
06:51:41 awplus-vcs VCS[1066]: Member 2 (0009.41fd.dd0b) cannot join stack -
incompatible VCS management subnet
```

Quality of Service (QoS) restriction

The management communication that the stack members exchange over the stacking link is vital for the successful operation of the stack. Be careful to avoid interrupting that communication. This communication traffic is transmitted on egress queue 7 on the stacking ports.

We recommend that you avoid sending any user traffic into queue 7 on a VCStack. In any QoS configuration on a stack, prioritize traffic only into queues 0-6. Moreover, you should ensure that the CoS-to-Queue map does not automatically use queue 7 for the transmission of packets that are received with a CoS value of 7. To ensure this, change the cos-to-queue map to put packets with a CoS value of 7 into queue 6. Use the command:

```
awplus(config)# mls qos map cos-queue 7 to 6
```

This configuration requirement applies to: SBx8100, x530/x530L, and GS980MX.

Stacking triggers

There are five trigger types defined for particular stack events. Any scripts that you configure for triggers are copied from the active master to all stack members.

The triggers are:

```
awplus(config-trigger)# type master-fail
awplus(config-trigger)# type stack member join
awplus(config-trigger)# type stack member leave
awplus(config-trigger)# type stack xem-stk up
awplus(config-trigger)# type stack xem-stk down
```

Licensing

On a stack, the **license** command will activate the license on all the current stack members. If other members are subsequently added to the stack, the license will not be automatically installed into them. They will connect to the stack successfully, even though they have a different license status to the other stack members. A log message warning you that a license mismatch now exists will be generated, but you must manually install the required license(s) into the new member(s). It is important to bring the new member(s) up to the same license level as the existing master. If this is not done, then if a license-lacking member becomes master at some point, then the whole stack will revert to the license level of that master.

To see the license status of stack members, use the command:

```
awplus# show license [<label>] member [<1-8>|all]
```

```
awplus#sh license member 2

Board region: Global

Feature licenses on stack member 2:

Index                         : 1
License name                  : Base License
Customer name                 : Base License
Type of license               : Full
License issue date            : 18-Sep-2023
Features included             : 10G_SFP+, AMF-GUEST, AMF-Starter, BASE-AW+,
                                BGP-64, BW-UNLIMITED, EPSR-MASTER, IPv6Basic,
                               L3-FORWARDING, L3-MC-ROUTE, LAG-FULL, MLDSnoop,
                               No-License-Lock, OSPF-64, RADIUS-100, RIP, UFO,
                                VCStack, VPN, VRRP

Index                         : 2
License name                  : AT-FL-x930-AM20
Customer name                 : ATL-NZ L3 CSG
Quantity of licenses          : 1
Type of license               : Full
License issue date            : 11-Sep-2023
Features included             : AMF-MASTER-20

Index                         : 3
License name                  : AT-FL-X930-01
Customer name                 : ATL-NZ L3 CSG
Quantity of licenses          : 1
Type of license               : Full
License issue date            : 11-Sep-2023
Features included             : BGP-5K, BGP4+, OSPF-FULL, OSPFv3-FULL, PIM,
                                PIM-100, PIM6, PTP-TRANS, RADIUS-FULL, RIPNG,
                                UDLD, VLAN-TRANS-FULL, VRF-LITE, VRF-LITE-63,
                                VlanDT

Index                         : 4
License name                  : AT-FL-x930-8032
Customer name                 : ATL-NZ L3 CSG
Quantity of licenses          : 1
Type of license               : Full
License issue date            : 11-Sep-2023
Features included             : CFM, G8032
```

You can use the **license** command to add a license to all stack members and the **no license** command to remove a license from all stack members.

```
MSWA-0512-CR1#license AT-FL-x930-AM40 6cGr233iQ8f9/v+5nHlDG/
mBTK0KGwsghTwb23+GQyg7iCObsJfkA==
A restart of affected modules may be required.
Would you like to continue? (y/n): y

Stack member 1 installed 1 license
Stack member 2 installed 1 license

2 licenses installed.
```

Software and configuration file synchronisation

A VCStack requires that the software version and the configuration files on all stack members are the same. If these files are not the same, then the stack cannot form or then operate correctly. To make stack formation easy, all these files are synchronized automatically on the stack by the stack master.

The following are synchronized:

- "Software release auto-synchronisation" on page 63
- "Shared running configuration" on page 65
- "Shared startup configuration" on page 65
- "Scripts" on page 66

Software release auto-synchronisation

The VCStack implementation supports a feature called **software auto-synchronization**. The effect of this feature is that when a new member joins a stack and has a software release that is different to the active master, then the active master's software release is copied onto the new member. The new member then reboots and comes up on that release.

The sequence of events that occurs at startup is shown in the output below:

```
Notice: Possible stack, please wait whilst searching for members.
12:37:30 awplus-2 VCS[1057]: Member 1 (0000.cd27.c4bf) has joined stack
12:37:30 awplus-2 VCS[1057]: Please configure 'stack virtual-mac' to minimize network
disruption from failovers
12:37:30 awplus-2 VCS[1057]: Member 1 (0000.cd27.c4bf) has become the Active master
12:37:34 awplus-2 VCS[1057]: Software incompatibility detected.
12:37:34 awplus-2 VCS[1057]: Auto synchronization starts.
Unit will reboot once that finishes. Please wait..
12:39:35 awplus-2 VCS: The new software x930 5.4.5-0.2.rel is set as preferred software and
used at the next reboot.
12:39:35 awplus-2 VCS: The current software x930 5.4.5-0.1.rel is set as backup software.
12:39:35 awplus-2 VCS: VCS SW version auto synchronization successfully completed.

Restarting system
```

The software auto-synchronization feature is enabled on all switches by default. You can enable or disable it using the command:

```
awplus(config)#(no) stack <1-8> software-auto-synchronization
```

- If you disable software auto-synchronization, you may disrupt a stack forming if the stack members have different software releases.
- If a new member joins the stack, and is running a software version that is different to the active master, the active master will reject the new member from the stack if it cannot synchronize the software releases.

Note: This feature can result in the new stack member downgrading its software release if the active master is running an older software version.

Occasionally you will find that the software version running on the existing stack members and the software version on the new stack member are sufficiently incompatible that the stack will not be able to auto synchronize the software version on the new member.

On these occasions, the log on the stack master may contain an alert message:

```
user.alert awplus VCS[994]: Neighbour on link 1.0.2 cannot join stack-incompatible
stack S/W version
```

Also, the internal stacking log on the stack master may contain similar messages, that can be seen by using the command **show stack full-debug | include version.**

```
awplus#show stack full-debug|include version
2015 Jul 3 02:04:12 user.debug awplus-2 VCS[992]: STK TRACE: Topology discovery
version mismatch - received v23.4
2015 Jul 3 02:04:12 user.debug awplus-2 VCS[992]:Neigbour on link 2.0.1 cannot
join stack - incompatible stack S/W version
```

When this occurs you will need to:

1. remove the new member from the stack
2. manually upgrade the software on that switch to the same version as is running on the existing stack members
3. re-connect the new member to the stack.

Stacking with a mixture of CFC960 v2 and earlier CFC960 in an SBx8100 system

If you want to combine CFC960 v2 and earlier CFC960 cards in a stack, make sure that the earlier cards are running 5.5.0-0.x before you combine them.

This applies whether you:

- add a CFC960 v2 card to a stack that contains earlier CFC960 cards, or
- add an earlier CFC960 card to a stack that contains CFC960 v2 cards.

Auto-synchronization will not update the software on the earlier CFC960 cards.

Note that this requirement only applies to CFC960 v2 cards that are labeled “SBx81CFC960 v2” on the front panel of the card. For the purposes of this requirement, all cards that are labeled “SBx81CFC960” are referred to as earlier cards, even if some of them are version 2 as indicated by their documentation. If you combine cards that are running different software, you need to remove the CFC960 v2 card or cards, update the software on the other cards, and then re-install the CFC960 v2 cards.

Shared running configuration

A single, unified running configuration is shared by all the switches in a stack.

This means that the running configuration on each stack member is exactly the same, and contains the configuration information for all stack members. For example, on a two-switch stack, switch A with stack ID 1 will show the configuration for switch B with stack ID 2, as well as its own configuration. If the running configuration on switch B with stack ID 2 is examined, the output will be exactly the same as that produced by switch A.

Shared startup configuration

Once a stack has formed, the startup configuration stored on each stack member is over written by the active master's startup configuration. This means all the switches in a stack have exactly the same startup configuration script.

Every time the startup configuration script on the active master is changed, for example when the running config is copied to the startup config, the updated startup script is copied to all the other stack members.

```
stack(config)#boot config-file interop.cfg
VCS synchronizing file across the stack, please wait..
File synchronization with stack member-2 successfully completed
[DONE]
```

It is important that you take this behavior into account when you first create a stack. If you have carefully crafted a startup configuration on the switch that you intend to be the active master, but some mistake results in a different switch becoming the active master when the stack forms, then the intended startup configuration will be over written by something else (assuming the same filename was used for both configurations).

We recommend that you:

- take care when first installing the VCStack to ensure that you configure the stack priority values correctly. If you do not, the wrong stack member may become the active master on first boot, and overwrite the stack with the wrong startup configuration.
- backup the startup configuration of the intended active master before connecting the switch to the stack. You can make another copy on the switch's flash file system with the command:

```
awplus# copy startup_filename.cfg backup_filename.cfg
```

As well as the running configuration and startup configuration, the stack synchronizes the **boot backup configuration** file, and the **fallback configuration** file. Like the startup configuration file, these are synchronized from the active master's file system, so the same recommendations apply.

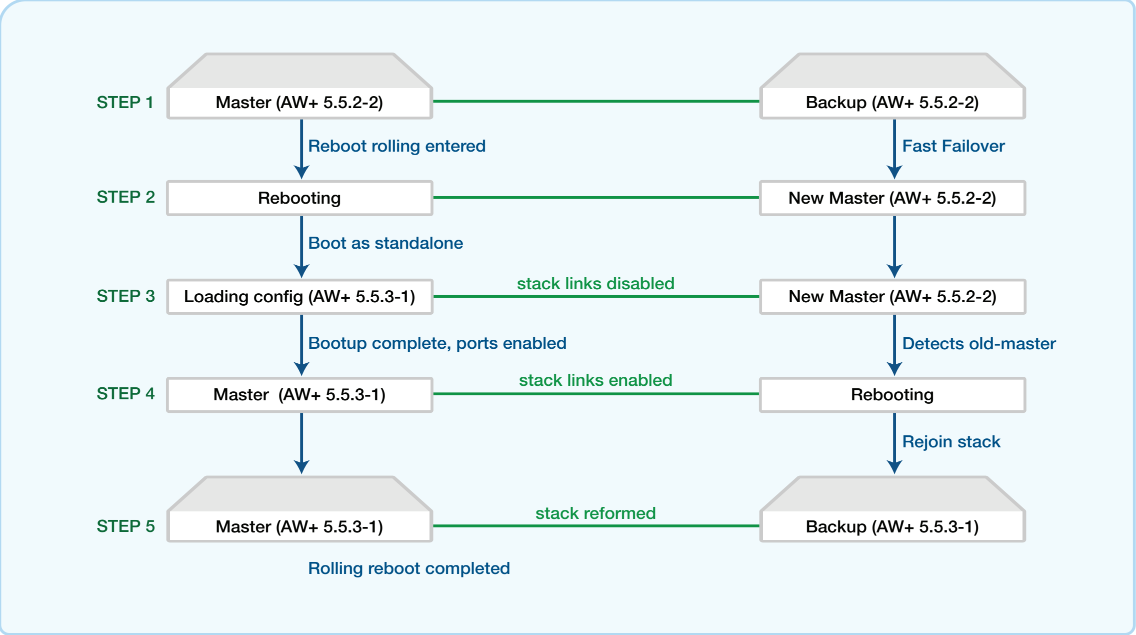
Scripts

Script files stored on the active master's file system are copied to the stack members.

Rolling reboot

A major benefit of virtual chassis stacking is that it provides unit resiliency—if one unit in the stack goes down, the other members of the stack will continue to forward data. It is highly desirable for this continuity of service to persist even when the stack is being rebooted. The purpose of the **reboot rolling** command is to reboot a stack in a manner that maintains continuity of service.

However, it's important to note that while a rolling reboot minimizes downtime during a software upgrade, it is not a 'hitless' feature like ISSU, and some service disruption is expected when using this feature. For detail on minimizing service disruption, see "Rolling reboots with route preservation" on page 68.



The diagram above shows a stack comprised of 2 switches, one Master and one Backup member. You can see that:

- At **step 1** there will be a short disruption as the Master reboots and the Backup takes over. This disruption is the same as would occur during any normal fast-failover.
- At **step 4**, there will be a longer disruption. This is because the Master, at this point in the process, is running a different software version from the previous master and has effectively booted in standalone mode. Therefore, there is no preservation of table or protocol states, and the master

has to re-converge any configured routing protocols and relearn forwarding information. The extent of this disruption will depend on the configured protocols and their re-convergence times.

Occasionally there are restrictions on rolling reboot when upgrading from some software versions on some switch models. Check the Release Note for the new release before using rolling reboot.

The **reboot rolling** command allows a stack to be rebooted in a rolling sequence so that no more than one unit of the stack is in reboot at any given time.

Here we have a stack of three switches:

```
awplus#show stack
Virtual Chassis Stacking summary information

ID  Pending ID  MAC address        Priority  Status  Role
1   -           0015.77c2.4b7d     128       Ready   Backup Member
3   -           0015.77e8.a892     128       Ready   Backup Member
4   -           0015.77c9.73cb     128       Ready   Active master

Operational Status                 Normal operation
Stack MAC address                  0015.77c9.73cb
```

We can see that stack member **4** is the Active master.

First, the stack master is rebooted causing the remaining stack members to failover and elect a new master.

Executing the **reboot rolling** command gives the following:

```
awplus#reboot rolling
The stack master will reboot immediately and boot up with the configuration file
settings. The remaining stack members will then reboot once the master has
finished re-configuring.

Continue the rolling reboot of the stack? (y/n):y

awplus#22:11:15 awplus VCS[995]: Automatically rebooting stack member-4 (MAC:
00.15.77.c9.73.cb) due to Rolling reboot

URGENT: broadcast message:
System going down IMMEDIATELY!

        ... Rebooting at user request ...
```

As soon as the rebooted Active master has reloaded, it becomes the Active master again. While it is rebooting, another switch in the stack will assume the Active master role. Immediately after the Active master has reloaded and assumed its role again, all of the other switches in the stack are rebooted at the same time.

```
Active master booting up:

Loading default configuration
.

done!
Received event network.configured

Rolling reboot, rebooting all other stack members, please wait for stack to
reform.
```

You can see in the Active master's log that the other stack members (1 and 3) have rebooted:

```
2015 May 10 22:12:11 user.crit awplus-4 VCS[995]: Member 4 (0015.77c9.73cb) has become
the Active master
2015 May 10 22:12:37 local6.notice awplus VCS[995]: Link down event on stack link 4.0.2
2015 May 10 22:12:37 local6.notice awplus VCS[995]: Link down event on stack link 4.0.1
2015 May 10 22:13:32 local6.notice awplus VCS[995]: Link up event on stack link 4.0.1
2015 May 10 22:13:32 local6.notice awplus VCS[995]: Link down event on stack link 4.0.1
2015 May 10 22:13:32 local6.notice awplus VCS[995]: Link up event on stack link 4.0.2
2015 May 10 22:13:33 local6.notice awplus VCS[995]: Link down event on stack link 4.0.2
2015 May 10 22:13:36 local6.notice awplus VCS[995]: Link up event on stack link 4.0.1
2015 May 10 22:13:36 user.crit awplus VCS[995]:Member 3 (0015.77e8.a892)has joined stack
2015 May 10 22:13:36 user.notice awplus VCS[995]: Link between members 4 and 3 is up
2015 May 10 22:13:37 local6.notice awplus VCS[995]: Link up event on stack link 4.0.2
2015 May 10 22:13:37 user.crit awplus VCS[995]:Member 1(0015.77c2.4b7d) has joined stack
2015 May 10 22:13:37 user.notice awplus VCS[995]: Link between members 4 and 1 is up
2015 May 10 22:13:37 user.notice awplus VCS[995]: Link between members 3 and 1 is up
```

Rolling reboots with route preservation

As mentioned earlier, rolling reboot does not provide a 'hitless' method of software upgrade and some service disruption will occur.

When performing a rolling reboot, a brief service interruption typically occurs as systems upgrade. You can minimize this by using the **route-preserve** feature, an optional parameter in the **reboot rolling** command:

```
awplus# reboot rolling [route-preserve [timeout <1-65535>]]
```

The route-preserve feature ensures that, during and immediately after the rolling reboot, your routes stay installed, reducing wait times that would otherwise occur while network routing protocols re-converge. The route-preserve feature provides an alternative way to retain routes learned by dynamic routing protocols and can be used alongside or instead of the graceful restart functionality provided by these protocols.

Here's how it works

All unicast IPv4 and IPv6 routes installed by your routing protocols (like BGP, OSPF, RIP, and ISIS) are held in place when you initiate the rolling reboot. Once the reboot of the primary (Master) device is complete, these routes return to their original settings, including metrics, distances, VRFs, and next hops. The primary (Master) device can then immediately resume managing traffic while the rest of the stack reboots.

To prevent outdated routes from lingering, each restored route is temporarily marked as ‘stale’, and a cleanup timer (default 5 minutes) begins. During this period, the network protocols will reconnect, re-establish routes, and remove the stale flag. You can adjust the ‘timeout’ parameter to accommodate different reboot times, depending on your network’s specifics.

When routes are restored using this feature they are flagged as stale in the RIB. Use the command **show (ip|ipv6) route database** to see this information.

During the rolling reboot, the stack master will log a summary, including the number of restored routes and the cleanup timer length. If any routes fail to restore, you’ll receive a notification, allowing you to address it promptly.

Configuring the resiliency feature

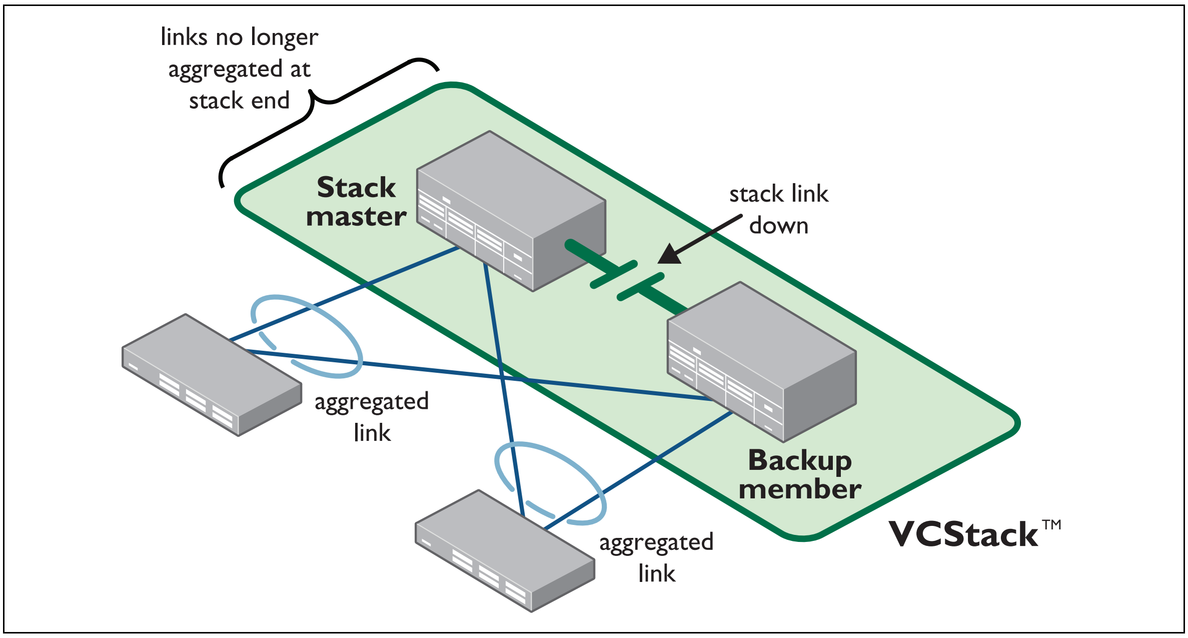
A VCStack behaves in a reliable and consistent manner if any component fails, whether the problem is with a stacking link or stack member. However, before looking at how the stack reacts to each of those scenarios, we first need to understand a generic mechanism that is available in Allied Telesis VCStack to help deal with failure scenarios—namely the resiliency link.

How to configure the resiliency link feature on a stack

If one or more stacking links fail, so that some stack members are no longer in contact with the active master switch, then the other stack members are left with a dilemma. Should they assume that the active master switch has gone down, and re-elect a new active master, or should they assume that the active master is still operating?

This problem is particularly important when a stack has multiple connections to edge switches in the network. In this network scenario, if a stack splits into two active sections that are operating independently, then the edge switches will continue to treat their uplink ports as aggregated and share traffic across the links.

However, the broken stack link could mean that traffic arriving at some of the stack members cannot reach the intended destination.



To ensure continuity, it's necessary to establish a backup mechanism that allows stack members to verify the status of the active master if the normal communication link to the active master is lost.

The mechanism provided in Allied Telesis VCStack is called the **resiliency link**, which may be either the eth0 port (on SBx908 GEN2, SBx8100, x950, and x930 Series switches) or a dedicated VLAN (resiliencylink VLAN) to which switch ports may become members. These resiliency ports are connected together, and the stack members all listen for periodic (one per second) Health Check messages from the master. As long as the active master sends Health Check messages, the other stack members know that the active master is still active.

This means that the stack members can know whether the active master is still operating if they lose contact with the active master through the stacking links.

The out-of-band Ethernet port is configured as a resiliency port with the command:

```
awplus(config)# stack resiliencylink eth0
```

Note that even if you configure the eth0 port as a resiliency port, you can still use it for out-of-band management.

For example, you could use the following commands to configure port1.0.1 on VLAN1000 as a resiliency link:

```
awplus(config)# stack resiliencylink vlan1000
awplus(config)# interface port1.0.1
awplus(config-if)# switchport resiliencylink
```

Note also that this VLAN is dedicated to the resiliency link function and must not be the stack management VLAN or a customer data VLAN.

In the situation where a stack member loses contact through the stacking links, but continues to receive health check messages via the resiliency link, the stack member becomes a disabled master. It runs the same configuration as the active master, but it has its links shutdown.

There is a trigger that can be configured when a switch becomes a disabled master with the commands:

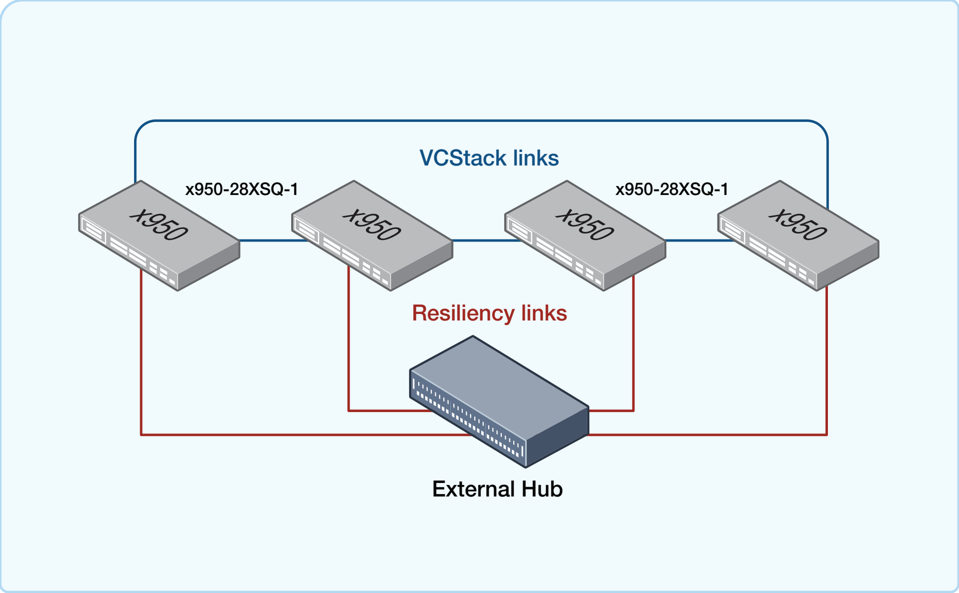
```
awplus(config)# trigger 82
awplus(config-trigger)# type stack disabled master
awplus(config-trigger)# script 1 flash:/disabled.scp
```

Although this command could activate any trigger script, the intention here is that the script will re-activate the links from their previously shutdown state, to enable the user to manage the switch.

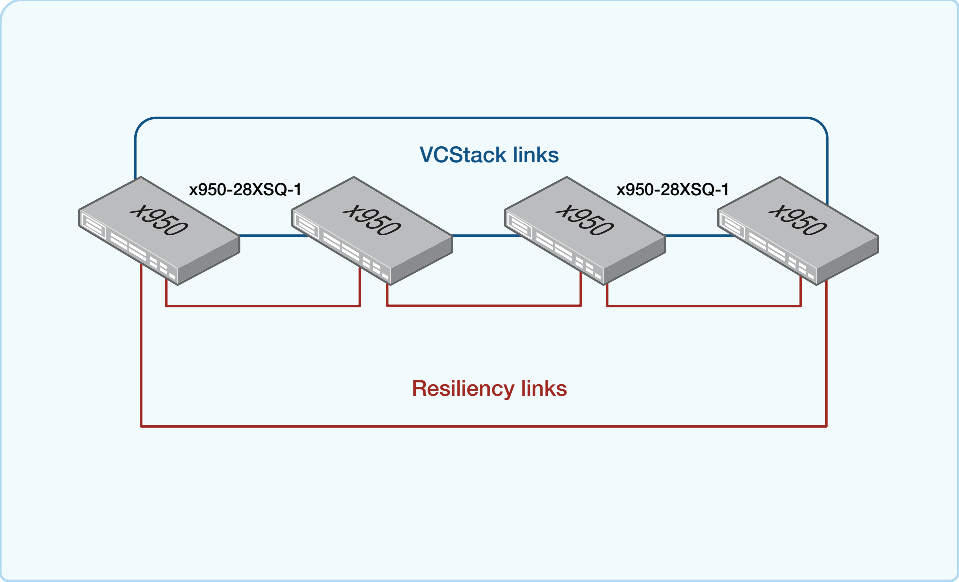
Resiliency link feature on VCStacks of three or more units

For VCStacks of three or more units, there are two supported topologies for connecting resiliency links between stack members:

1. If the resiliency links are configured using the **out of band management port** (eth0), these ports must be connected together via a dedicated external hub as per the diagram below. It is not recommended to use a Layer 2 switch for this purpose due to potential differences in the way non-IP multicast packets may be handled on different Layer 2 switches. If a Layer 2 switch is used, spanning-tree should be disabled and it should treat non-IP multicast packets as broadcast packets and flood them.



2. If the resiliency links are configured using the **switchport resiliencylink** feature, two switch ports on each stack member should be configured, and the resiliency link ports should be connected in a ring as shown in the example diagram below.

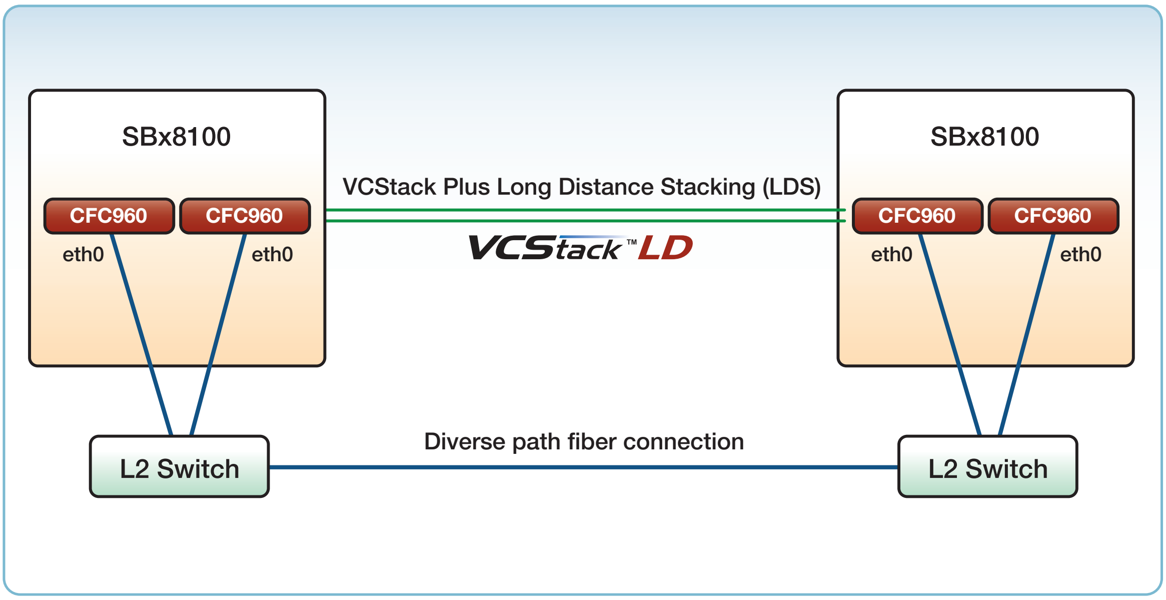


Configuring a resiliency link with VCStack Plus

From software version **5.4.9-0.1** onwards, resiliency link is supported on VCStack Plus. This is where two stacked SBx8100 chassis, each containing two CFC960s are linked via short distance cabling or long distance fiber connections. Each CFC960 Ethernet port member must be connected to an out-of-band (Layer 2) network. As mentioned in the previous section, the out-of-band Ethernet port (eth0) is configured as a resiliency port with the command:

```
awplus(config)# stack resiliencylink eth0
```

Figure 3: Configuring a resiliency link with VCStack Plus



The relevant command information is as follows:

- The global configuration command **stack disabled-master-monitoring** is supported on the SBx81CFC960. This command is enabled by default and recommended for improved resiliency.
- The global configuration command **stack resiliencylink <interface-name>** is supported on the SBx8100 CFC960. Note, the only supported interface is “eth0”.
- The show command **show stack detail** displays the resiliency link information on the SBx8100 CFC960.
- The interface port command **switchport resiliencylink** is not supported the SBx8100 CFC960.

Note: The resiliency link feature is not supported on the SBx8100 **standalone** chassis.

Virtual MAC

To minimize data loss, enable a virtual MAC when a new stack member may need to become the master. The VCStack then uses the virtual MAC address for communication with other devices. As this single virtual MAC address is used for the complete VCStack, there is no change of MAC address if a new stack member is required to become master. In conjunction with Fast Failover, this ensures maximum continuity of network service, as there is no need for other devices in the network to learn a new MAC address into their MAC or ARP tables.

You can configure the virtual MAC address manually by specifying a VCStack virtual chassis ID. Or the VCStack can use its default virtual chassis ID. The ID selected will determine which virtual MAC address the stack will use. The MAC address assigned to a stack must be unique within its network.

Enter the chassis ID in decimal format. The virtual chassis ID entered will form the last 12 bits of a pre-selected MAC prefix component; that is, 0000.cd370xxx.

For example:

```
awplus(config)# stack virtual-mac
awplus(config)# stack virtual-chassis-id 63
```

This will result in a virtual MAC address of: 0000.cd37.003f

Note: VCStacks with virtual MAC disabled are not supported, therefore virtual MAC must be enabled on all VCStacks.

Repairing a broken stack

When two stub stacks are reconnected, the stack will detect if there is more than one master (active, disabled, or fallback). You will see console messages and logs that report a **duplicate master** was detected. The stack carries out an election to choose the true active master, based on priorities and MAC addresses. Any losing master will reboot and become a backup member. The switches that have been running on their fallback configuration (most likely those in the same stub as the losing active master) will reboot and come up on the shared configuration pushed to them by the winning active master.

When a stack detects a duplicate master, a debug snapshot file is created and stored in flash. The file is called:

```
debug-duplicate-master-<time & date>.tgz
```

The debug snapshot does not contain a ***.core*** file, as is produced when a process unexpectedly terminates on a switch. Instead, it contains a snapshot of various pieces of information within the software at the time when the stack detected the duplicate master.

Disabled master

A properly functioning VCStack contains an (active) master and one (backup) member. Under fault conditions it is possible for the stack member to lose connectivity with the stack master. In this situation the stack member without master connectivity will become a ‘disabled master’. Once in this state the disabled master will disable all of its own ports.

Apart from this, the operation and “look and feel” of a disabled master is very similar to an active master. The active master’s ports are unaffected by this change, and they will continue to forward traffic normally.

By disabling all the stub’s switchports, the disabled master avoids potential network connectivity problems that could result from by having two stack masters using the same configuration, or two switches in separated stubs trying to share the same “logical” communications paths such as a non functioning aggregated links. The active master’s ports are unaffected by this, and they will continue to forward traffic normally. Note that status information for members of the stack stub can be accessed by logging into the disabled master, in the same way as obtaining status information for a normal stack.

Disabled Master Monitoring (DMM)

The stack resiliency link and disabled master state offer a unique prevention of catastrophic network connectivity problems. However, when stack members become separated, the network is still left in a fragile state where the stack master no longer has the redundancy of the backup stack member. If the original stack master were to subsequently fail whilst the stack was separated, then all network connectivity would be lost if the disabled master’s switchports remained shut down.

The DMM feature avoids this situation by continuing to monitor the status of the original stack master (the active master) via the stack resiliency link. When the DMM feature is enabled, the disabled master can detect a failure of the original stack master within a few seconds.

If a failure is detected, the disabled master transitions to the active master state and automatically re-enables all its switchports. This allows traffic forwarding via the stack to continue.

Table 1: Stack behaviour and DMM

STACK BEHAVIOR WITH DMM DISABLED	STACK BEHAVIOR WITH DMM ENABLED
■ The VCStack breaks with the Stack Resiliency Link configured and enabled.	■ The VCStack breaks with the Stack Resiliency Link configured and enabled.
■ The separated stack member becomes a disabled master.	■ The separated stack member becomes a disabled master.
■ The disabled master does not monitor the active master.	■ The disabled master monitors the active master.
■ If the active master fails then the disabled master does not become the active master (no state transition).	■ If the active master fails then the disabled master becomes the active master (disabled to active transition).
■ No switchports are re-enabled on the disabled master. No traffic is forwarded.	■ Switchports on the disabled master are re-enabled. Traffic is still forwarded.

To enable the DMM feature, use the commands:

```
awplus# configure terminal
awplus(config)# stack disabled-master-monitoring
```

To disable the DMM feature, use the commands:

```
awplus# configure terminal
awplus(config)# no stack disabled-master-monitoring
```

To show the status of DMM on the VCStack, use the command:

```
awplus# show stack [detail]
```

To apply a trigger upon transition from active master state to disabled master state, use the command:

```
awplus# type stack disabled-master
```

To apply a trigger upon transition from disabled master state to active master state, use the command:

```
awplus# type stack master-fail
```

Note: A disabled master trigger allows you to specify a script to reconfigure the disabled master on the fly, should a catastrophic failure separate the stack. This is useful to configure an alternate IP address so you can still log in to the disabled master via an SSH or a Telnet connection. The trigger script should use the **no shutdown** command to re-enable any switchports needed for an SSH or a Telnet management connection.

Replacing a stack member

A stack member can be removed from a stack (hot-swapped out) with minimal impact on stack traffic. To do this, power-down the stack member, and disconnect its stacking ports. You can seamlessly swap a stack member switch into the stack to replace another with the same configuration. This provides a simple way to replace an out-of-service switch with minimal impact, and minimal administration requirement. Before inserting the replacement device into the stack, make sure that:

- The replacement device is running a compatible firmware version.
- You set the Stack ID on the replacement device to the same ID as the device being replaced.
- The replacement device is installed with the same licenses as the other stack members.

Once these requirements are met, insert the new stack member, reconnect the stacking ports and power-up the new stack member.

Converting stacking ports to network ports

x930 Series

Reconfiguring StackQS stacking module ports as network ports

On x930 Series switches, you can configure each port on the StackQS card as:

- a stacking port, or
- one 40 Gbps port, or
- four 10 Gbps ports (28-port models only).

The ports are configured as stacking ports by default. When converted to network switch ports, they operate as 40 Gbps ports by default.

To convert the ports to network switch ports, you need to disable VCStack on the ports. There are two options for doing this:

- make the switch into a standalone switch, by running the command:
  no stack <*stack-id*> enable, or
- use the 10 Gbps front-panel SFP+ ports for stacking, by running the command:
  stack enable builtin-ports

Operating the ports as four 10 Gbps ports

To use an StackQS port as four 10 Gbps ports, you need a QSFP-4SFP10G-3CU or QSFP-4SFP10G-5CU breakout cable. Then use the command **platform portmode interface** to change the port or ports to 10 Gbps.

For example, to change both the stacking ports into 10 Gbps ports, use the commands:

```
awplus#configure terminal
awplus(config)#no stack 1 enable
awplus(config)#platform portmode int port1.1.1,port1.1.5 10gx4
```

In 10 Gbps mode, the ports are numbered as follows:

SLOT NUMBER	PORT NUMBER	BECOMES
1	1.1.1	1.1.1, 1.1.2, 1.1.3, 1.1.4
2	1.1.5	1.1.5, 1.1.6, 1.1.7, 1.1.8

When changing the portmode setting, you must also remove any interface and channel-group configuration from the specified ports, and then reboot the switch. Note that the StackQS ports can only operate as four 10 Gbps network switch ports on 28-port switch models, not on 52-port switch models.

Monitoring and troubleshooting

The physical connection and the signaling communications between stack members are vital to stacking. Because of this, the **show** commands and debugging facilities for VCStack are oriented around stack port status and stack signaling communications.

Checking stack status

You can check that the stack links have come up successfully by:

- checking the LEDs on the switch
- using the **show stack detail** command, see "Show stack detail command" on page 79

The tables following describe the LED state and functions for each switch series.

LEDs

The stacking LED on the **SwitchBlade x908 GEN2** and the **x950**, **x930**, **x550**, **x530**, **x530L**, **x510**, **x330-28GTX**, **XS900MX**, **GS980MX**, and **GS900MX/MPX Series** is a 7-segment LED that displays the actual stack ID number. This LED is on the front panels of the units.

In addition, there is a link activity LED associated with each transceiver socket. The table below shows the LED states when the socket is used for stacking.

LED	STATE	MEANING
Link activity	Off	The slot is empty, the transceiver has not established a link to a network device, or the LEDs are turned off. To turn on the LEDs, use the eco-friendly button.
	Solid green	The transceiver has established a link at full speed to another switch in the stack.
	Flashing green	The transceiver is receiving or transmitting packets.

SBx8100 The front panel of the **SwitchBlade x8100 CFC960 control card** indicates the control card’s status within the stack.

LED	STATE	MEANING
CFC	Green Flashing green	Active master Startup
	Amber Flashing red	Backup member (standby) Disabled

x930 The front panel of the x930 Series switch has the following LEDs for monitoring back-port stacking:

LED	STATE	MEANING
L/A 1	Off	Stack port 1 has not established a link to a stacking port on another VCStack stacking module.
	Solid Green	Stack Port 1 has established a link to a stacking port on another VCStack stacking module.
	Flashing Green	Stack Port 1 has established a link to a stacking port on another VCStack stacking module and is sending or receiving packet traffic.
L/A 2	Off	Stack port 2 has not established a link to a stacking port on another VCStack stacking module.
	Solid Green	Stack Port 2 has established a link to a stacking port on another VCStack stacking module.
	Flashing Green	Stack Port 2 has established a link to a stacking port on another VCStack stacking module and is sending or receiving packet traffic.

Show stack detail command

The **show stack detail** command provides a snapshot of the stack status. It includes a full summary of the status of all the stack members and the status of their connections to the other member. This allows you to check that all the stack members have active connections to each other and have recognized correctly which neighboring switch is connected to each of their stacking ports.

```
awplus#show stack detail
Virtual Chassis Stacking detailed information
Stack Status:
------------------------------------------------------------------
Operational Status                      Normal operation
Management VLAN ID                      4094
Management VLAN subnet address          192.168.255.0
Virtual Chassis ID                      3224 (0xc98)
Virtual MAC address                     Disabled
Disabled Master Monitoring              Enabled
Mixed Mode                              Disabled

Stack member 1:
------------------------------------------------------------------
ID                                      1
Pending ID                              -
MAC address                             e01a.ea33.cf6b
Last role change                        Wed Apr 27 18:45:50 2022
Product type                            AT-x530-28GPXm
Role                                    Active Master
Status                                  Ready
Priority                                20
Host name                               x530-stack
S/W version auto synchronization        On
Resiliency link status                  Not configured
Stack port1.0.27 status                 Learnt neighbor 2,connected port2.0.52
Stack port1.0.28 status                 Learnt neighbor 3,connected port3.0.27

Stack member 2:
------------------------------------------------------------------
ID                                      2
Pending ID                              -
MAC address                             e01a.ea50.7748
Last role change                        Wed Apr 27 18:45:52 2022
Product type                            AT-x530-52GPXm
Role                                    Backup Member
Status                                  Ready
Priority                                50
Host name                               x530-stack-2
S/W version auto synchronization        On
Resiliency link status                  Not configured
Stack port2.0.51 status                 Learnt neighbor 3,connected port3.0.28
Stack port2.0.52 status                 Learnt neighbor 1,connected port1.0.27
Stack member 3:
------------------------------------------------------------------
ID                                      3
Pending ID                              -
MAC address                             e01a.ea3d.29c2
Last role change                        Wed Apr 27 18:45:51 2022
Product type                            AT-x530-28GPXm
Role                                    Backup Member
Status                                  Ready
Priority                                128
Host name                               x530-stack-3
S/W version auto synchronization        On
Resiliency link status                  Not configured
Stack port3.0.27 status                Learnt neighbor 1, connected port1.0.28
Stack port3.0.28 status                Learnt neighbor 2, connected port2.0.51
x530-3-stack#
```

The clearest way to receive stack debug output is to enable console logging of VCStack messages. Using this method, you receive just the stack debug messages without the other messages that are seen when you enable terminal monitor.

To enable console logging of VCStack messages, use the command:

```
awplus(config)# log console program VCS
```

You must enter the parameter **VCS** in uppercase letters when you enter this command. This command enables you to receive a record of stack link and communication events.

For example, if a stack port goes down, and then comes up again, the series of messages output is:

```
awplus#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
awplus(config)#conf tlog console program VCS
awplus(config)#exit
awplus#
awplus#08:37:01 awplus VCS[670]: Link down event on stack link port1.0.2
08:37:01 awplus VCS[670]: STK TRACE: Xbar update event is queued.
08:37:01 awplus VCS[670]: STK TRACE: Link port1.0.1: --> 2 (0000.cd28.5377)
08:37:01 awplus VCS[670]: STK TRACE: Link port1.0.2: N/A
08:37:01 awplus VCS[670]: STK TRACE: Stack topology has changed - updating stac
k H/W routes for L2 connectivity
08:37:02 awplus VCS[670]: STK TRACE: stack H/W route update complete

08:37:45 awplus VCS[670]: Link up event on stack link port1.0.2
08:37:45 awplus VCS[670]: Beginning neighbor discovery on link port1.0.2
08:37:45 awplus-2 VCS[669]: Link up event on stack link port2.0.1
08:37:46 awplus VCS[670]: STK TRACE: > Txd Hello msg on port1.0.2 (nbr ?):
08:37:46 awplus VCS[670]: STK TRACE: < Rxd Hello msg on port1.0.2 (nbr ?):
08:37:46 awplus VCS[670]: Already have one link between Member 1 and Member 2
08:37:46 awplus VCS[670]: STK TRACE: Xbar update event is queued.
08:37:46 awplus VCS[670]: STK TRACE: Link port1.0.1: --> 2 (0000.cd28.5377)
08:37:46 awplus VCS[670]: STK TRACE: Link port1.0.2: --> 2 (0000.cd28.5377)
08:37:46 awplus VCS[670]: STK TRACE: Ring topology is now complete
08:37:46 awplus VCS[670]: STK TRACE: > Txd Hello msg on port1.0.2 (nbr 2):
08:37:46 awplus VCS[670]: STK TRACE: > Txd Topology DB msg on L2: sender 1
08:37:46 awplus VCS[670]: Neighbor discovery on link port1.0.2 has successfully
completed
08:37:46 awplus VCS[670]: STK TRACE: Stack topology has changed - updating stac
k H/W routes for L2 connectivity
08:37:46 awplus VCS[670]: STK TRACE: < Rxd Topology DB msg on L2: sender 2
08:37:46 awplus VCS[670]: STK TRACE: Received topology DB message from unconnec
ted neighbor Member 2
08:37:47 awplus VCS[670]: STK TRACE: stack H/W route update complete

awplus#
```

If stack-link communication to a stack member is completely lost, the series of messages output is:

```
BlueCore#
BlueCore#01:04:55 BlueCore VCS[994]: STK TRACE: < Rxd Link-down msg on L2: 4 (00
15.77c9.
73cb) <--> 2 (0015.77e8.a87d)
01:04:55 BlueCore VCS[994]: Link between members 4 and 2 is down
01:04:55 BlueCore VCS[994]: STK TRACE: Link 1.0.1: --> 4 (0015.77c9.73cb)
01:04:55 BlueCore VCS[994]: STK TRACE: Link 1.0.2: --> 3 (0015.77e8.a892) --> 2
 (0015.77e8.a87d)
01:04:55 BlueCore VCS[994]: STK TRACE: Stack topology has changed - updating st
ack H/W routes for L2 connectivity
01:04:55 BlueCore-3 VCS[996]: Link down event on stack link 3.0.2
01:04:56 BlueCore VCS[994]: STK TRACE: stack H/W route update complete
01:04:55 BlueCore-4 VCS[995]: Link down event on stack link 4.0.1
01:04:56 BlueCore VCS[994]: STK TRACE: < Rxd Link-down msg on L2: 3 (0015.77e8.
a892) <--> 2 (0015.77e8.a87d)
01:04:56 BlueCore VCS[994]: Link between members 3 and 2 is down
01:04:56 BlueCore VCS[994]: STK TRACE: Link 1.0.1: --> 4 (0015.77c9.73cb)
01:04:56 BlueCore VCS[994]: STK TRACE: Link 1.0.2: --> 3 (0015.77e8.a892)
01:04:56 BlueCore VCS[994]: Member 2 (0015.77e8.a87d) is leaving the stack (unr
eachable via stack links)
01:04:56 BlueCore VCS[994]: STK TRACE: Shutting down access to member 2's file
system
01:04:56 BlueCore VCS[994]: STK TRACE: Member 2 (0015.77e8.a87d) state Backup M
ember --> Leaving
01:04:56 BlueCore VCS[994]: Member 2 (0015.77e8.a87d) has left stack
01:04:56 BlueCore VCS[994]: STK TRACE: Stack topology has changed - updating st
ack H/W routes for L2 connectivity
01:04:56 BlueCore VCS[994]: STK TRACE: stack H/W route update complete
01:04:56 BlueCore VCS: Updating stack file system access...
01:04:57 BlueCore VCS: Shutting down access to member-2's file system
01:04:59 BlueCore VCS[994]: HA monitoring detected member-1 no change
01:04:59 BlueCore VCS[994]: HA monitoring detected member-3 no change
01:04:59 BlueCore VCS[994]: HA monitoring detected member-4 no change
01:04:59 BlueCore VCS[994]: HA monitoring detected member-2 left stack
01:04:59 BlueCore NSM[997]: Removal event on unit 2.0 has been completed
```

To see a very detailed log of stacking related events after they have occurred, and other VCStack debug information, use the command:

```
awplus# show stack full-debug [<1-8>]
```

Even if you had not been capturing stack log output at the moment an event occurred, you can still retrospectively obtain the logging information by using this command. If you do not specify a stack ID, then each stack member’s output is displayed, one after the other.

This command produces a large amount of output, as shown in the following figure:

```
BlueCore#show stack full-debug
Detailed debugging information for stack member 1

VCS host configuration
------------------------------------------------------------
Unit stackable, stack H/W present
VCS mgmt VLAN 4094, subnet 192.168.255.0
Topology: Ring
Neighbor port-1: 4
Neighbor port-2: 3
Stack Virtual MAC: feature disabled
  ID Mac Address     PP  LJ  Cfg  IP              Status  Role
> 1  0015.77c2.4b7d  00   N    Y  192.168.255.1   Ready   Active master
  2  0015.77e8.a87d  00   Y    Y  192.168.255.2   Syncing Backup Member
  3  0015.77e8.a892  00   N    Y  192.168.255.3   Ready   Backup Member
  4  0015.77c9.73cb  00   N    Y  192.168.255.4   Ready   Backup Member
VCS Inter-process connectivity configuration
------------------------------------------------------------
Type       Lower      Upper      Port Identity              Publication
-------------------------------- -------------------------- ------------------
0          16781313   16781313   <1.1.1:14753793>           14753794   zone
           16781314   16781314   <1.1.2:3494871041>         3494871042
           16781315   16781315   <1.1.3:3173122049>         3173122050
           16781316   16781316   <1.1.4:1199587329>         1199587330
1          1          1          <1.1.1:14761987>           14761988   node
9500       1          1          <1.1.1:14745638>           14745639   cluster
           2          2          <1.1.2:3494862887>         3494862888
           3          3          <1.1.3:3173113895>         3173113896
           4          4          <1.1.4:1199579174>         1199579175
           100        100        <1.1.1:14753794>           14753795   node
           200        200        <1.1.1:14786564>           14786565   node

                                 <1.1.1:14745679>           14745680   node
                                 <1.1.1:14753852>           14753853   node
                                 <1.1.1:14745674>           14745675   node
                                 <1.1.1:14745672>           14745673   node
                                 <1.1.1:14745670>           14745671   node
                                 <1.1.1:14745663>           14745664   node
                                 <1.1.1:14876727>           14876728   node
                                 <1.1.1:14819381>           14819382   node
                                 <1.1.1:14745651>           14745652   node
                                 <1.1.1:15073328>           15073329   node
                                 <1.1.1:14991406>           14991407   node
                                 <1.1.1:14745630>           14745631   node
                                 <1.1.1:14745629>           14745630   node
                                 <1.1.1:14745625>           14745626   node
                                 <1.1.1:14753816>           14753817   node
                                 <1.1.1:14745623>           14745624   node
                                 <1.1.1:14770198>           14770199   node
                                 <1.1.1:14753813>           14753814   node
                                 <1.1.1:14745620>           14745621   node
                                 <1.1.1:14762003>           14762004   node
                                 ...

[Continued on next page]
```

```
Link <multicast-link>
  Window:20 packets
  RX packets:102 fragments:0/0 bundles:0/0
  TX packets:42244 fragments:0/0 bundles:0/0
  RX naks:118 defs:2 dups:115
  TX naks:4 acks:5 dups:231
  Congestion bearer:0 link:0  Send queue max:18 avg:5

Link <1.1.1:vlan4094-1.1.2:vlan4094>
  ACTIVE  MTU:1500  Priority:10  Tolerance:3000 ms  Window:50 packets
  RX packets:261 fragments:0/0 bundles:0/0
  TX packets:236 fragments:0/0 bundles:0/0
  TX profile sample:28 packets  average:42 octets
  0-64:93% -256:7% -1024:0% -4096:0% -16354:0% -32768:0% -66000:0%
  RX states:115 probes:29 naks:0 defs:0 dups:0 tos:0
  TX states:130 probes:41 naks:0 acks:1 dups:0
  Congestion bearer:0 link:0  Send queue max:1 avg:0
Link <1.1.1:vlan4094-1.1.3:vlan4094>
  ACTIVE  MTU:1500  Priority:10  Tolerance:3000 ms  Window:50 packets
  RX packets:1401 fragments:0/0 bundles:0/0
  TX packets:1368 fragments:0/0 bundles:0/0
  TX profile sample:86 packets  average:43 octets
  0-64:98% -256:2% -1024:0% -4096:0% -16354:0% -32768:0% -66000:0%
  RX states:7745 probes:1234 naks:0 defs:0 dups:0 tos:0
  TX states:10440 probes:3858 naks:0 acks:1 dups:0
  Congestion bearer:0 link:0  Send queue max:5 avg:0

Link <1.1.1:vlan4094-1.1.4:vlan4094>
  ACTIVE  MTU:1500  Priority:10  Tolerance:3000 ms  Window:50 packets
  RX packets:187 fragments:0/0 bundles:0/0
  TX packets:179 fragments:0/0 bundles:0/0
  TX profile sample:48 packets  average:42 octets
  0-64:98% -256:2% -1024:0% -4096:0% -16354:0% -32768:0% -66000:0%
  RX states:7994 probes:1333 naks:0 defs:0 dups:0 tos:0
  TX states:10532 probes:4009 naks:0 acks:0 dups:0
  Congestion bearer:0 link:0  Send queue max:5 avg:0
VCS management traffic
------------------------------------------------------------
Mon May 17 01:09:05 UTC 2015

vlan4094  Link encap:Ethernet  HWaddr 00:15:77:C2:4B:7D
          inet addr:192.168.255.1  Bcast:192.168.255.15  Mask:255.255.255.240
          inet6 addr: fe80::215:77ff:fec2:4b7d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:138575 errors:0 dropped:0 overruns:0 frame:0
          TX packets:205773 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:13436002 (12.8 MiB)  TX bytes:23187142 (22.1 MiB)

            Pkts       Bytes
Non-VCS Q7: 0          0
Rx AIS:     21173      2559494
Rx mcast:   1912

VCS packet Replication
Type           Tx         Rx         Drops
Total          0          0          0
Bytes          0          0
STP            0          0          0
 [Continued on next page]
```

```
EPSR           0          0          0
LACP           0          0          0
Mcast          0          0          0
sflow          0          0          0
BcExc          0          0          0
Other          0          0          0
PortAuth       0          0          0
TxBuff min: 952 86268ms ago
TxBuff cur: 1008
Last Rx:    11ms ago
Last Tx:    14ms ago

           CPU0
 16:       4135   IPIC   Level Enabled   0     serial
 17:     186739   IPIC   Level Enabled   0     linux-kernel-bde
 22:          0   IPIC   Level Enabled   0     LM81 1
 23:          0   IPIC   Level Enabled   0     LM81 2
 24:        287   IPIC   Level Enabled   0     mpc83xx_spi
 25:        749   IPIC   Level Enabled   0     i2c-mpc
 26:          0   IPIC   Level Enabled   0     i2c-mpc
 74:          0   IPIC   Edge  Enabled   0     GPIO
 75:          0   IPIC   Edge  Enabled   0     GPIO
BAD:          0
VCS debug
------------------------------------------------------------
2015 May 16 23:39:23 kern.info awplus kernel: TIPC: Activated (version 1.6.4 com
piled May  4 2010 11:45:08)
2015 May 16 23:39:23 kern.info awplus kernel: TIPC: Started in single node mode
2015 May 16 23:39:25 user.info awplus VCS[960]: Parsing 'stack' commands from co
nfig file /flash/default.cfg
2015 May 16 23:39:25 kern.info awplus kernel: TIPC: Started in network mode
2015 May 16 23:39:25 kern.info awplus kernel: TIPC: Own node address <1.1.1>, ne
twork identity 4711
2015 May 16 23:39:31 user.debug awplus VCS[994]: SFL: Base feature license alloc
ated
2015 May 16 23:39:31 user.info awplus VCS[994]: SFL: [stackd] LicenceCheck: Virt
ual-MAC is active
2015 May 16 23:39:31 user.info awplus VCS[994]: SFL: [stackd] LicenceCheck: retu
rns Success.
2015 May 16 23:39:31 user.debug awplus VCS[994]: 2. init app stk-topo-event sess
Pt=0x100696e8, addr=0x4800e000
2015 May 16 23:39:31 user.debug awplus VCS[994]: 2. init app stk-port-1 sessPt=0
x10067480, addr=0x4800f000
2015 May 16 23:39:31 user.debug awplus VCS[994]: 2. init app stk-port-2 sessPt=0
x10069c90, addr=0x48010000
2015 May 16 23:39:31 user.debug awplus VCS[994]: 2. init app stk-topo-msg sessPt
=0x10069b40, addr=0x48011000
2015 May 16 23:39:31 user.debug awplus VCS[994]: 2. init app stk-topo-error sess
Pt=0x10069b58, addr=0x48012000
2015 May 16 23:39:31 user.debug awplus VCS[994]: 2. init app stk-resiliency-link
 sessPt=0x10066418, addr=0x48013000
2015 May 16 23:39:31 user.debug awplus VCS[994]: STK DEV  : Stacking Resiliency
Link counters register is successful.
2015 May 16 23:39:45 user.debug awplus VCS[994]: STK DISC : Member-1 XEMs presen
t: Bay 0: XEM-STK,
2015 May 16 23:39:45 user.info awplus VCS[994]: Parsing 'stack resiliencylink vl
an' commands from config file /flash/default.cfg
2015 May 16 23:39:45 user.info awplus VCS[994]: Stacking Ports were discovered o
n the mainboard of member 1
```

Counters

You can obtain detailed counters relating to stack events and signaling packets with the **show counter stack** command. You can use these for tracking down whether signaling packets are being lost, by checking if there are discrepancies between the number sent from one switch and the number received by its neighbor. The event counters make it possible to see if unexpected events have been occurring on the stack. The following few pages contain some example output from the **show counter stack** command:

```
BlueCore#show counter stack
Virtual Chassis Stacking counters

Stack member 1:

Topology Event counters
Units joined          ......... 4
Units left            ......... 1
Links up              ......... 8
Links down            ......... 4
ID conflict           ......... 0
master conflict       ......... 0
master failover       ......... 0
master elected        ......... 1
master discovered     ......... 0
SW autoupgrades       ......... 0

Stack Port 1 Topology Event counters
Link up               ......... 2
Link down             ......... 1
Nbr re-init           ......... 0
Nbr incompatible      ......... 0
Nbr 2way comms        ......... 1
Nbr full comms        ......... 2

Stack Port 2 Topology Event counters
Link up               ......... 1
Link down             ......... 0
Nbr re-init           ......... 0
Nbr incompatible      ......... 0
Nbr 2way comms        ......... 1
Nbr full comms        ......... 0
Topology Message counters
Tx Total              ......... 37
Tx Hellos             ......... 3
Tx Topo DB            ......... 2
Tx Topo update        ......... 2
Tx Link event         ......... 0
Tx Reinitialise       ......... 0
Tx Port 1             ......... 3
Tx Port 2             ......... 2
Tx 1-hop transport    ......... 5
Tx Layer-2 transport  ......... 32
Rx Total              ......... 87
Rx Hellos             ......... 4
Rx Topo DB            ......... 2
Rx Topo update        ......... 10
Rx Link event         ......... 6
[Continued on next page]
```

```
Rx Reinitialise       ......... 0
Rx Port 1             ......... 2
Rx Port 2             ......... 2
Rx 1-hop transport    ......... 4
Rx Layer-2 transport  ......... 14

Topology Error counters
Version unsupported   ......... 0
Product unsupported   ......... 0
XEM unsupported       ......... 0
Too many units        ......... 0
Invalid messages      ......... 0

Resiliency Link counters
Health status good    ......... 0
Health status bad     ......... 0
Tx                    ......... 0
Tx Error              ......... 0
Rx                    ......... 0
Rx Error              ......... 0
Stack member 3:

Topology Event counters
Units joined          ......... 4
Units left            ......... 1
Links up              ......... 8
Links down            ......... 4
ID conflict           ......... 0
master conflict       ......... 0
master failover       ......... 0
master elected        ......... 0
master discovered     ......... 1
SW autoupgrades       ......... 0

Stack Port 1 Topology Event counters
Link up               ......... 1
Link down             ......... 0
Nbr re-init           ......... 0
Nbr incompatible      ......... 0
Nbr 2way comms        ......... 1
Nbr full comms        ......... 3

Stack Port 2 Topology Event counters
Link up               ......... 4
Link down             ......... 3
Nbr re-init           ......... 0
Nbr incompatible      ......... 0
Nbr 2way comms        ......... 2
Nbr full comms        ......... 0
[Continued on next page]
```

```
Topology Message counters
Tx Total              ......... 37
Tx Hellos             ......... 4
Tx Topo DB            ......... 3
Tx Topo update        ......... 3
Tx Link event         ......... 1
Tx Reinitialise       ......... 0
Tx Port 1             ......... 2
Tx Port 2             ......... 5
Tx 1-hop transport    ......... 7
Tx Layer-2 transport  ......... 30
Rx Total              ......... 87
Rx Hellos             ......... 6
Rx Topo DB            ......... 3
Rx Topo update        ......... 9
Rx Link event         ......... 5
Rx Reinitialise       ......... 0
Rx Port 1             ......... 2
Rx Port 2             ......... 5
Rx 1-hop transport    ......... 7
Rx Layer-2 transport  ......... 80

Topology Error counters
Version unsupported   ......... 0
Product unsupported   ......... 0
XEM unsupported       ......... 0
Too many units        ......... 0
Invalid messages      ......... 0

Resiliency Link counters
Health status good    ......... 0
Health status bad     ......... 0
Tx                    ......... 0
Tx Error              ......... 0
Rx                    ......... 0
Rx Error              ......... 0

Stack member 4:

Topology Event counters
Units joined          ......... 4
Units left            ......... 1
Links up              ......... 8
Links down            ......... 4
ID conflict           ......... 0
master conflict       ......... 0
master failover       ......... 0
master elected        ......... 0
master discovered     ......... 1
SW autoupgrades       ......... 0

Stack Port 1 Topology Event counters
Link up               ......... 5
Link down             ......... 4
Nbr re-init           ......... 0
Nbr incompatible      ......... 0
Nbr 2way comms        ......... 4
Nbr full comms        ......... 5
 [Continued on next page]
```

```
Stack Port 2 Topology Event counters
Link up               ......... 1
Link down             ......... 0
Nbr re-init           ......... 0
Nbr incompatible      ......... 0
Nbr 2way comms        ......... 1
Nbr full comms        ......... 0

Topology Message counters
Tx Total              ......... 52
Tx Hellos             ......... 9
Tx Topo DB            ......... 5
Tx Topo update        ......... 4
Tx Link event         ......... 3
Tx Reinitialise       ......... 0
Tx Port 1             ......... 9
Tx Port 2             ......... 3
Tx 1-hop transport    ......... 12
Tx Layer-2 transport  ......... 40
Rx Total              ......... 76
Rx Hellos             ......... 9
Rx Topo DB            ......... 5
Rx Topo update        ......... 7
Rx Link event         ......... 3
Rx Reinitialise       ......... 0
Rx Port 1             ......... 10
Rx Port 2             ......... 3
Rx 1-hop transport    ......... 13
Rx Layer-2 transport  ......... 63

Topology Error counters
Version unsupported   ......... 0
Product unsupported   ......... 0
XEM unsupported       ......... 0
Too many units        ......... 0
Invalid messages      ......... 0

Resiliency Link counters
Health status good    ......... 0
Health status bad     ......... 0
Tx                    ......... 0
Tx Error              ......... 0
Rx                    ......... 0
Rx Error              ......... 0
```

Disabling a faulty XEM on SBx908 GEN2 and x950 Series switches

From software version 5.4.9-0.1 onwards, you can use the **no xem <bayid> enable** command to keep a disabled XEM while a reboot or a hot swap on that XEM bay occurs. This means you can keep a faulty XEM disabled while waiting for its replacement to arrive.

This command is available on SBx908 and x950 Series. It is accessed in global configuration mode, and is therefore savable. This means you can power off a XEM bay to stay across a hot swap of the associated bay or a reboot.

Stacking on legacy switches

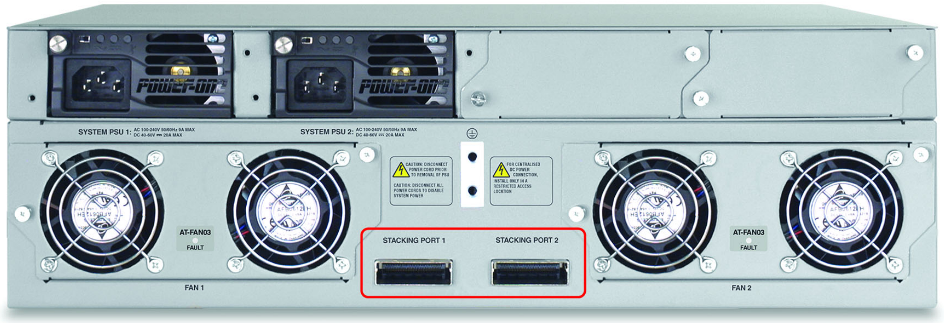
This section is dedicated to delivering comprehensive stacking information for our legacy switches, ensuring that you have all the knowledge and resources needed to harness the potential of your existing infrastructure.

The following table lists legacy product series and their last supported software release:

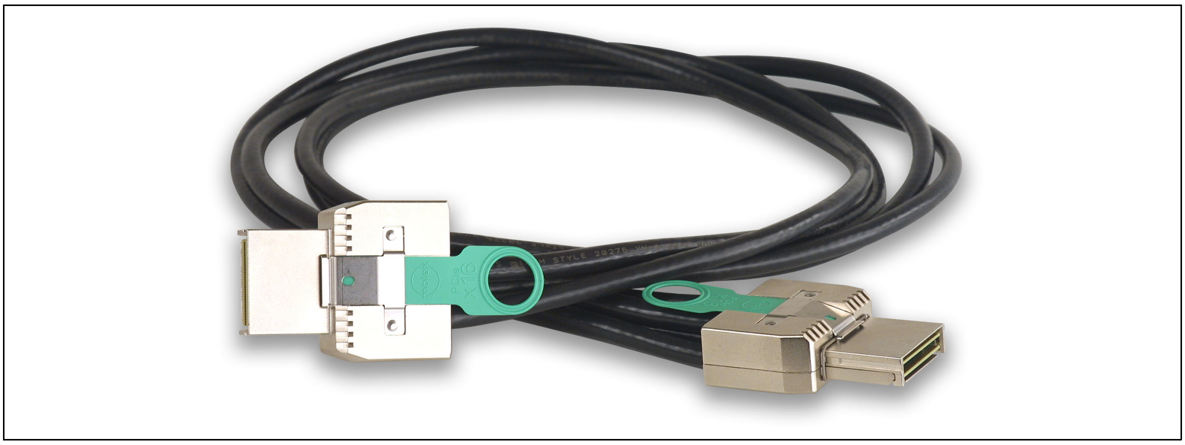
PRODUCT SERIES	LAST SUPPORTED SOFTWARE RELEASE
IX5	5.5.1-2.x
x510	5.5.1-2x
x310	5.5.1-2x
GS900MX	5.5.1-2x
SBx908	5.4.8-1.x
DC2552XS	5.4.8-0.x
x610	5.4.7-0.x
x900	5.4.4-4.x
x600	5.4.2-3.6

Back-port stacking on SwitchBlade x908 switches

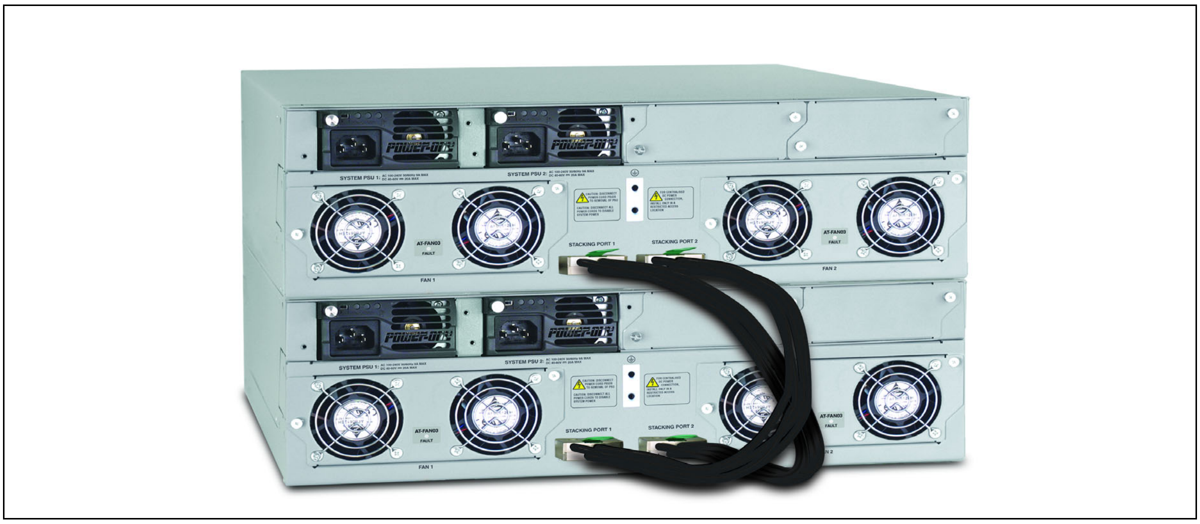
On the rear (lower center) of the SwitchBlade x908 chassis, there is a pair of fixed stacking ports.


SYSTEM PSU 1: AC 100-240V 50/60Hz 9A MAX DC 40-60V ⎓ 20A MAX
SYSTEM PSU 2: AC 100-240V 50/60Hz 9A MAX DC 40-60V ⎓ 20A MAX
CAUTION: DISCONNECT POWER CORD PRIOR TO REMOVAL OF PSU
CAUTION: DISCONNECT ALL POWER CORDS TO DISABLE SYSTEM POWER
FOR CENTRALISED DC POWER CONNECTION, INSTALL ONLY IN A RESTRICTED ACCESS LOCATION
AT-FAN03
FAULT
FAN 1
STACKING PORT 1
STACKING PORT 2
AT-FAN03
FAULT
FAN 2

Back port stacking requires a specific cable. The product code is AT-HS-STK-CBL650.



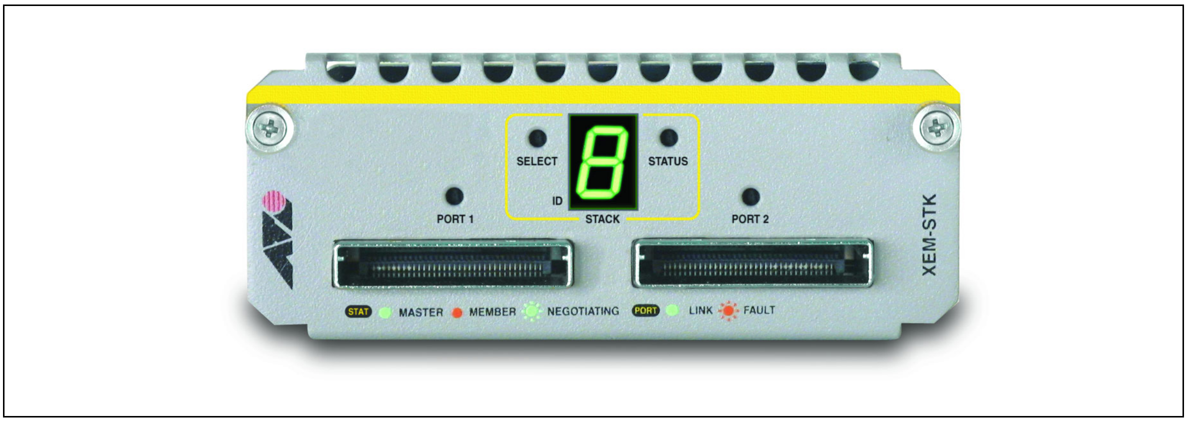
You can stack two SwitchBlade x908 switches together using these ports and cables.



Note that the cables are crossed over—port 1 of the top switch is connected to port 2 of the bottom switch, and vice versa.

Front-port stacking using XEM-STKs on x900 Series switches

You can fit the XEM bays on x900 Series switches with a specialized stacking XEM called the XEM-STK.

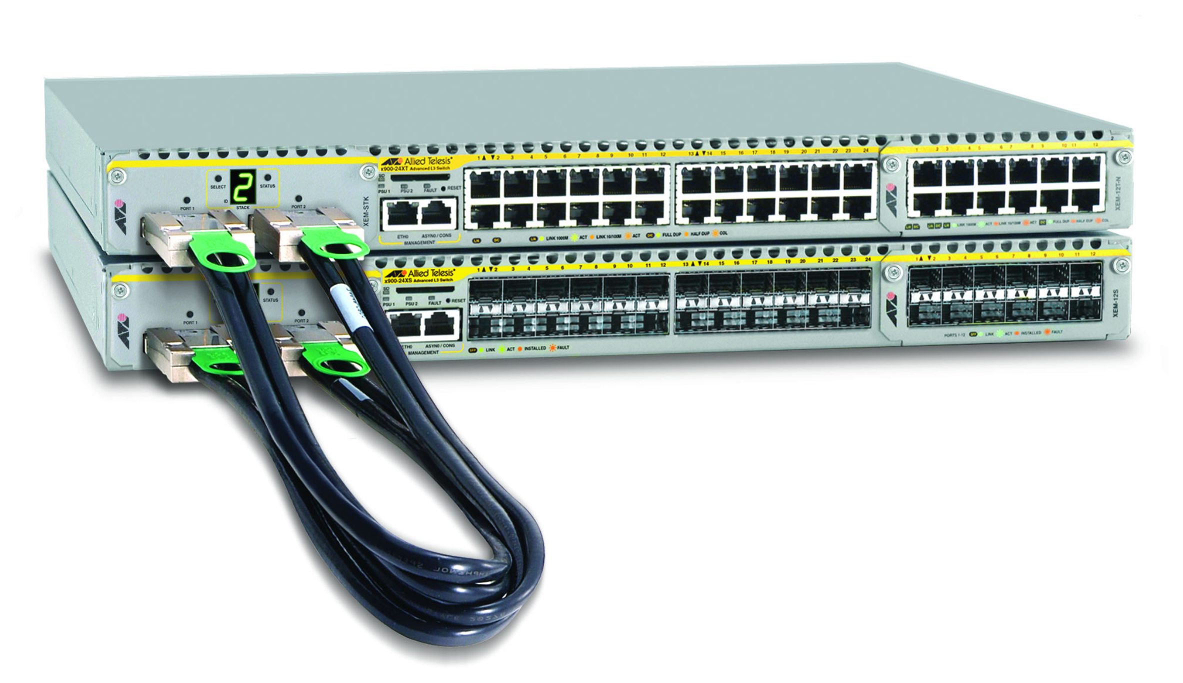


The LED number on the front of the XEM-STK shows the stack ID of the switch that the XEM-STK is installed in. See the section "Identifying switches in a stack" on page 12 for more information about stack IDs.

The specific cable type that connects these XEMs are purchased individually as either 350mm or 2 meter long cables. The product codes are:

- AT-XEM-STK-CBL350
- AT-XEM-STK-CBL2.0

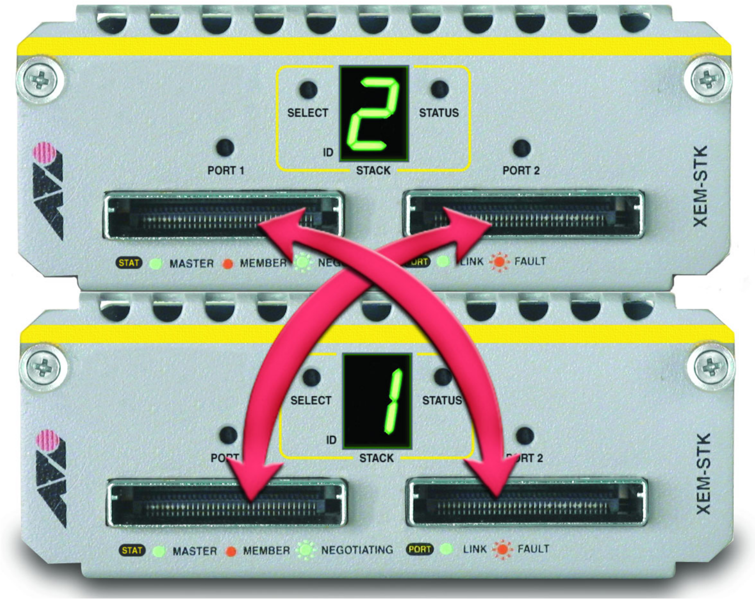
With these XEMs and cables, you can create a stack of two x900 Series switches.



Connecting the cables to the switches

Stacked switches are connected in a ring, with port 1 of one switch connected to port 2 of the next switch. The last switch in the stack then connects to the first switch using port 2 of the first switch and port 1 of the last switch.

In two switch stacks, this means that the connection consists of just a crossed pair of cables. You may also use only one cable to connect the switches, but you will halve the bandwidth between stack members.



Front-port stacking on CentreCOM GS900MX/MPX Series switches

The CentreCOM GS900MX/MPX Series switches allow 4 units to be stacked using front-port stacking, providing 40Gbps bandwidth. The GS900MX/MPX Series have four 10GbE SFP+ ports, two of which may be used for stacking instead of network connectivity.

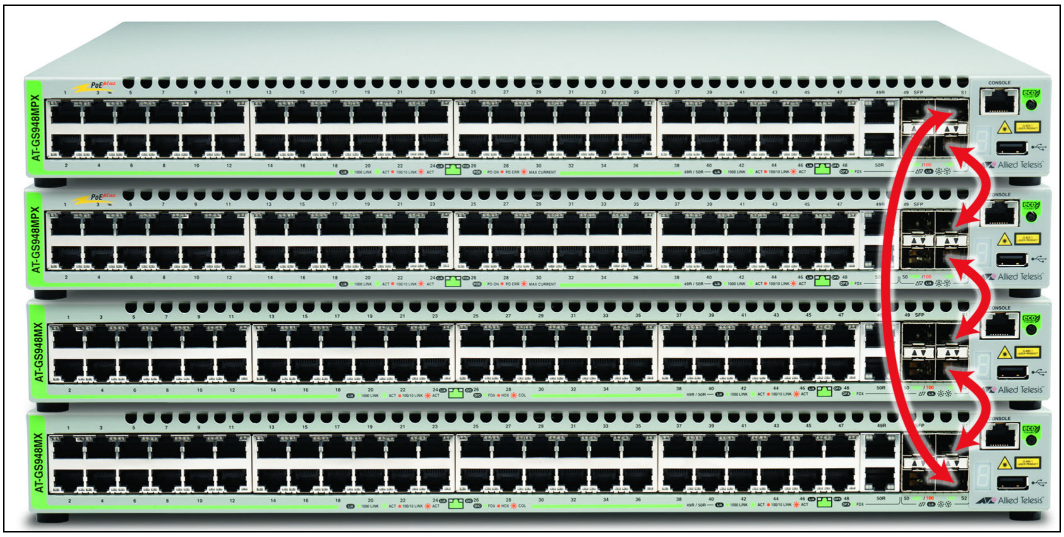
The CentreCOM GS900MX/MPX Series consists of the following switches: GS924MX, GS924MPX, GS948MX, GS948MPX.

Stacking cables

Use the following cable to stack GS900MX/MPX Series switches:

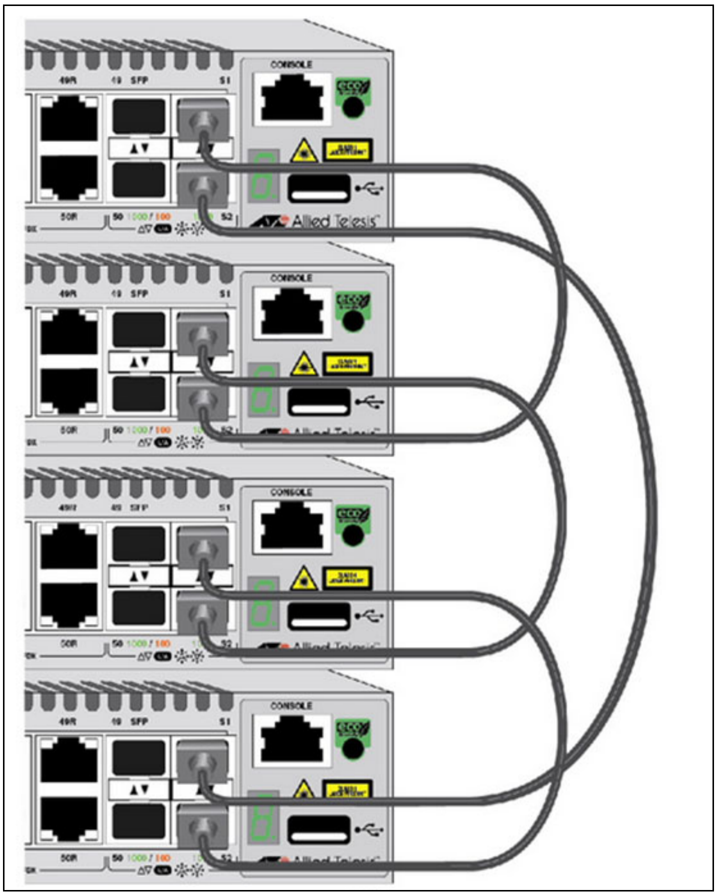
- AT-SP10TW1-1 meter SFP+ direct attach cable

The stacking cables must form a ring, as shown in the diagram below:



Front-port stacking on x310 switches

x310 switches can be connected in stacks of up to 4 units. The stacking ports are the S1 and S2 ports on the front of the unit. The stacking connections go from the S1 port of one switch to the S2 port of the next switch, and can be formed into a full ring.



The stacking ports S1 and S2 can be used as network ports, when stacking has been disabled by the **no stack enable** command, and the switch has been rebooted.

The port numbers allocated to the stacking ports when stacking has been disabled are the sequential numbers beyond the end of the rest of the port numbers, i.e. port1.0.27 and port1.0.28 on the 28-port switches, and port1.0.51 and port1.0.52 on the 52-port switches.

Front-port stacking on DC2552XS/L3 switches

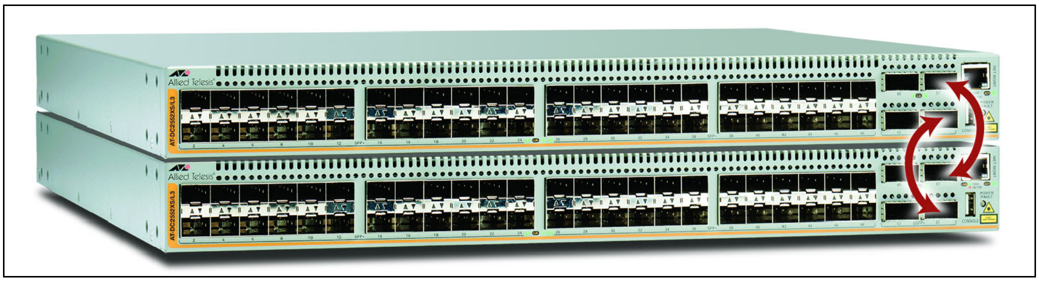
The DC255XS/L3 Series switch provides four QSFP+ 40G ports which are **stacking** ports by default. For stacking purposes, the DC2552XS/L3 Series switch allows two units to be stacked using front-port stacking. There are two stacking options available:

1. Use all four QSFP+ ports, providing 320Gbps connectivity.
2. Use two QSFP+ ports, to provide 160Gbps connectivity and still provide redundancy.

Note: When VCStack is configured on the DC255XS/L3 Series switch, **all** four QSFP+ ports become **stacking** ports, and none are available for network connections, even if only one or two ports are used for stacking.

Unlike the SwitchBlade and x-Series switches, the stacking cables on the DC2552XS/L3 must connect ports of the same number. A stack will form regardless of which port numbers are used for linking, as long as the same numbers are linked together.

For example, connecting port 1.0.57 on switch 1 to port 2.0.57 on switch 2 and port 1.0.61 on switch 1 to port 2.0.61 on switch 2, as shown below, is a valid option:

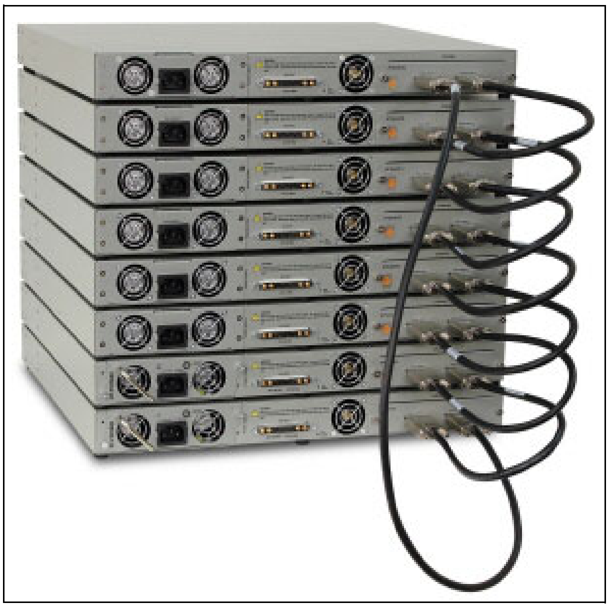


Whereas, connecting port 1.0.57 on switch 1 to port 2.0.61 on switch 2 and port 1.0.57 on switch 1 to port 2.0.61 on switch 2, would not work.

Back-port stacking on x610 Series switches

With a choice of 24-port and 48-port versions and optional 10 Gigabit uplink units, the x610 Series can connect anything from a small workgroup to a large business. A fully resilient solution is created where up to eight units can form a single virtual chassis, with dual connections to key servers and access switches. Expansion modules are available for local and long-distance stacking or can be configured to provide two additional 10G ports.

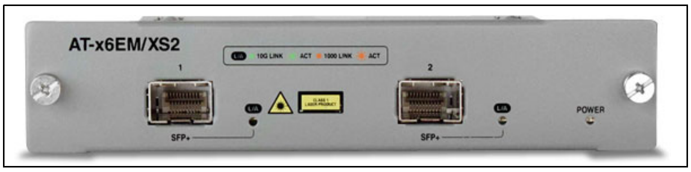
The stacking cables must form a ring, as shown in the diagram below.



Expansion modules

The x610 Series switches use the following two expansion modules:

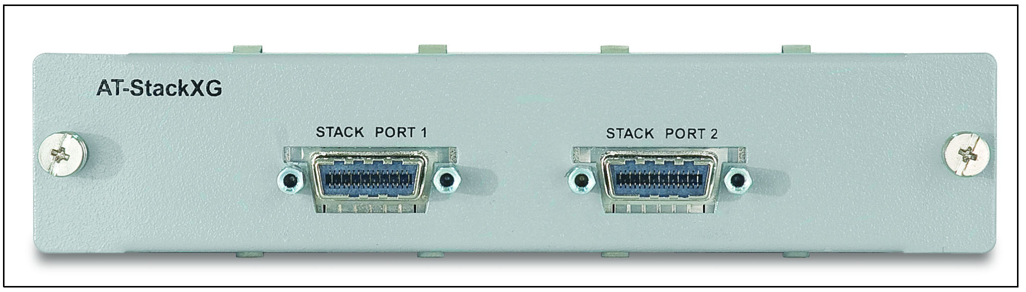
- AT-x6EM/XS2-00 —Expansion module (2 x SFP+) for long-distance stacking or for use as two additional 10GbE ports.



If the AT-x6EM/XS2 module is installed in the switch, and stacking is enabled, with the command **stack enable** (enabled by default), then the two ports on the AT-x6EM/XS2 module will operate as stacking ports.

If stacking is disabled using the command **no stack enable** (followed by saving the configuration and rebooting the switch), then the two ports on the AT-x6EM/XS2 module are re-purposed as normal 10Gbps network ports. The port numbers assigned to the ports are 1.1.1 and 1.1.2.

- AT-StackXG-00—Expansion module with two full-duplex, 12 Gbps stacking ports, one AT-StackXG/0.5-00 cable included. This is exactly the same module as is used in x600 Series switches.



Stacking cables

A variety of stacking cables are available for use with the x610 Series switches.

Cables that can be used with AT-StackXG-00:

- AT-StackXG/0.5-00—0.5 meter cable for stacking
- AT-StackXG/1-00—1 meter cable for stacking

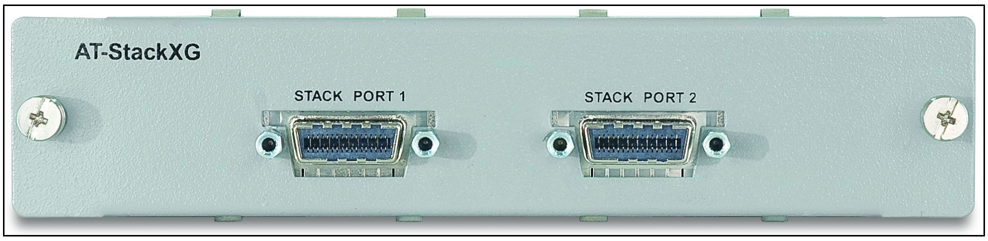
Cables that can be used with AT-x6EM/XS2-00

- AT-SP10TW1—1 meter SFP+ direct attach cable
- AT-SP10TW3—3 meter SFP+ direct attach cable
- AT-SP10TW7—7 meter SFP+ direct attach cable

In addition, standard SFP+ units can be inserted into the x6EM/XS2-00, and connected by cables of whatever length the SFP+ modules can support.

AT-StackXG slide-in modules on x600 Series switches

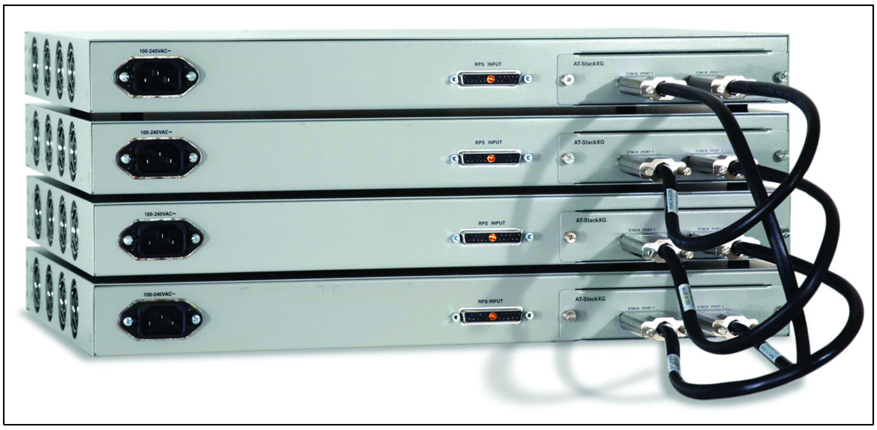
On the rear of the x600 chassis you can insert a slide-in module, the AT-StackXG-00. This connector can also be used in x610 Series switches, see "Expansion modules" on page 97.



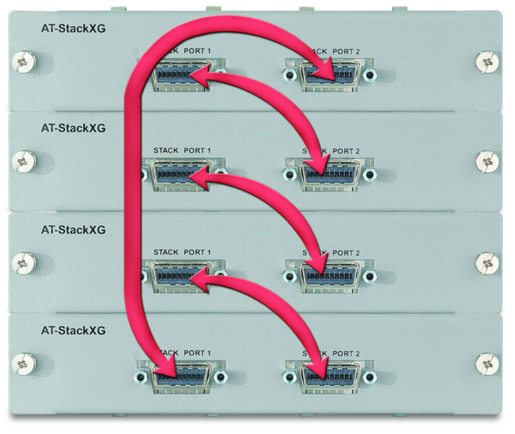
The specific cable type that connects the AT-StackXG are purchased as either 0.5 or 1 meter long cables. The product codes are:

- AT-STACKXG/0.5-00—0.5 meter cable for stacking
- AT-STACKXG/1-00—1 meter cable for stacking

You can stack up to four x600 switches using the AT-StackXG cables.



Like the other stacking methods, the connections are crossed-over–port 1 on one switch is connected to port 2 on its neighbor and the switches are connected in a ring. The following figure shows how to connect a 4 switch stack of x600 Series switches. Once again, you could connect the switches without one of the cables, but you would halve the bandwidth between stack members.



Renumbering the whole stack using the XEM Select button

On switches connected with XEM-STKs, you can use the **Select** button on the front of the XEM-STK. This achieves the same effect as the **renumber cascade** command. When you press the **Select** button on a XEM-STK, the switch with that XEM-STK is assigned stack ID 1, and the other members of the stack are numbered sequentially from there.

General stacking restrictions by product on legacy switches

There are some restrictions to what products and connections you can combine in a single stack. In general:

- switch families cannot be stacked together
- you cannot combine different stacking methods or cables.

SwitchBlade x908

compatible stack partners	A SwitchBlade x908 switch can only stack with another SwitchBlade x908 switch.
maximum stack size	2 switches.
cable type	■ For back-port stacking: AT-HS-STK-CBL650

DC2552XS/L3

compatible stack partners	DC2552XS/L3 switches can only stack with other DC2552XS/L3 switches.
maximum stack size	2 switches.
cable type	QSFP+ Stack cables: ■ AT-QSFP1CU, AT-QSFP3CU: 1m and 3m direct attach cable ■ AT-MTP12-1, AT-MTP12-5: 1m and 5m MTP optical cable for AT-QSFPSR

x900 Series

compatible stack partners	x900 Series switches can only stack with other x900 switches. Within the x900 Series: ■ x900-12XT/S switches can only stack with x900-12XT/S switches ■ x900-24XS and x900-24XT can stack together
maximum stack size	2 switches.
cable type	■ AT-XEM-STK-CBL350 (350mm length) or AT-XEM-STK-CBL2.0 (2 meter length).

x610 Series

compatible stack partners	x610 Series switches can stack with x600 switches, but the x600 stack restrictions will apply, so the whole stack is at the level of the x600. Within the x610 Series, you can stack any model with another model. This means that all the switches in the x610 Series can stack together without restriction.
maximum stack size	8 switches.
cable type	■ AT-StackXG/0.5-00—0.5 meter cable for stacking ■ AT-StackXG/1-00—1 meter cable for stacking ■ AT-SP10TW1—1 meter SFP+ direct attach cable ■ AT-SP10TW3—3 meter SFP+ direct attach cable ■ AT-SP10TW7—7 meter SFP+ direct attach cable ■ Allied Telesis SFP+ module with AT-X6EM/XS2 for long distance stacking

x600 Series

compatible stack partners	x600 Series switches can stack with x610 switches, but the x600 stack restrictions will apply, so the whole stack is at the feature level of the x600. Within the x600 Series, you can stack any model with another model. This means that all the switches in the x600 Series can stack together without restriction.
maximum stack size	4 switches.
cable type	■ AT-STACKXG/0.5 (0.5 meter length) or ■ AT-STACKXG/1 (1 meter length).

x510 and IX5 Series

compatible stack partners	x510 Series switches can stack with other x510 switches. They can also stack with x510L and IX5 switches.
maximum stack size	4 switches.
cable type	Standard Allied Telesis branded Twinax direct attach cables or SFP+ modules can be used to stack x510 Series switches. For **local** stacking: ■ For switches running AlliedWare Plus **prior** to 5.4.5: AT-SP10TW1 —1 meter stacking cable ■ For switches running AlliedWare Plus 5.4.5 or later: ■ AT-SP10TW1—1 meter SFP+ direct attach cable ■ AT-SP10TW3—3 meter SFP+ direct attach cable ■ AT-SP10TW7—7 meter SFP+ direct attach cable For **long distance** stacking: ■ Fiber cable ■ For switches running AlliedWare Plus **prior** to 5.4.5: AT-StackOP/0.3 (300m) or AT-StackOP/9.0 (9km) optical stacking modules (two per switch). ■ For switches running AlliedWare Plus 5.4.5 or **later**, you can use any Allied Telesis SFP+ module.

x310 Series

compatible stack partners	x310 Series switches can only stack with other x310 switches, (currently available in 10-100 switches only).
maximum stack size	4 switches.
cable type	Standard Allied Telesis branded Twinax direct attach cables can be used to stack x310 switches. For local stacking: ■ AT-SP10TW1 —1 meter stacking cable.

CentreCOM GS900MX/MPX Series

compatible stack partners	CentreCOM GS900MX/MPX Series switches can only stack with other CentreCOM GS900MX/MPX Series switches.
maximum stack size	4 switches.
cable type	Standard Allied Telesis branded Twinax direct attach cables can be used to stack CentreCOM GS900MX/MPX Series switches. For local stacking: ■ AT-SP10TW1—1 meter SFP+ direct attach cable.

Stacking capability for legacy switches

VCStack is implemented on a number of different series of Allied Telesis switch products. The implementations on different product series have different capabilities. The table below provides a summary of the capabilities available on each product series:

Table 2: Stacking capability by product series - legacy

PRODUCT SERIES	MAX UNITS IN STACK	LONG DISTANCE SUPPORT	MAX STACKING BANDWIDTH
SBx908	2	No	160Gbps
x900	2	No	60Gbps
x610	8	Yes	48Gbps
x600	4	No	48Gbps
x510	4	Yes	40Gbps
x510L	4	Yes	40Gbps
IE510-28GSX	4	Yes	40Gbps
IX5	4	Yes	40Gbps
x310	4	No	4Gbps

Note: Stacking is not supported in Secure Mode.

Monitoring legacy switches

The tables following describe the LED state and functions for each switch series.

x610 The front panel of the **x610 Series** switch has the following LEDs for monitoring stacking:

LED	STATE	MEANING
MSTR	Off	The switch is not part of a stack or is a member unit of the stack.
	Solid Green	The switch is the master unit of the stack.
L/A 1	Off	Stack port 1 has not established a link to a stacking port on another VCStack stacking module.
	Solid Green	Stack Port 1 has established a link to a stacking port on another VCStack stacking module.
	Flashing Green	Stack Port 1 has established a link to a stacking port on another VCStack stacking module and is sending or receiving packet traffic.
L/A 2	Off	Stack port 2 has not established a link to a stacking port on another VCStack stacking module.
	Solid Green	Stack Port 2 has established a link to a stacking port on another VCStack stacking module.
	Flashing Green	Stack Port 2 has established a link to a stacking port on another VCStack stacking module and is sending or receiving packet traffic.
PRES	Off	Expansion slot for VCStack stacking module is empty.
	Solid Green	A VCStack stacking module is installed in the switch.

SBx908 back-port stacking The front panel of the **SwitchBlade x908** has LEDs for monitoring back-port stacking:

LED	STATE	MEANING
Port 1 and Port 2	Green	A stacking link is established.
	Amber (flashing slowly)	The link has transmission fault.
	Off	The stacking link is down.
Master	Green	The switch is the stack master.
	Amber	The switch is a backup member.
	Green (flashing)	The stack is selecting a stack master.
	Off	The switch is not a stack member.

XEM-STK for x900 and SBx908

The LEDs on the **XEM-STK** show the following:

LED	STATE	MEANING
Port 1 and Port 2	Green	A stacking link is established.
	Amber (flashing slowly)	The link has transmission fault.
	Off	The stacking link is down.
Status	Green	The switch is the stack master.
	Amber	The switch is a backup member.
	Green (flashing)	The stack is selecting a stack master.
	Off	The switch is not a stack member.
Numeric ID	1 to 8	The stack member ID.
	Off	The switch is not a stack member.

x600

The front panel of the **x600 Series** switch has the following LEDs for monitoring stacking:

LED	STATE	MEANING
MSTR	Green	The switch is the stack master.
	Off	The switch is a backup member.
L/A 1 and L/A 2	Green	A stacking link is established on that link.
	Green (flashing)	The link is transmitting or receiving data.
	Off	The stacking link is down.
PRES	On	An AT-STACKXG module is correctly installed in the switch.
	Off	There is no AT-STACKXG installed in the switch, or the module is installed incorrectly.

Displaying the stack IDs

Most AlliedWare Plus switches display the stack ID number on an 8-segment LED on the front panel.

Alternatively, to see the stack ID on a switch before it is connected to a stack, use the command:

```
awplus# show stack
```

The output will show the current stack ID and any pending change to the stack ID.

```
awplus#show stack
Virtual Chassis Stacking summary information

ID  Pending ID  MAC address       Priority  Status  Role
1   -           0000.cd27.c4bf    128       Ready   Active master

Operational Status                Standalone unit
Stack MAC address                 0000.cd27.c4bf
```

You can also use this command to see the stack numbers on a two-switch stack. To match the physical switch with the right stack ID number, look for the **active master** LED on the front panel. Then use the **show stack** command to show all members of the stack. You can match the LED status to the **show** command output.

```
awplus#show stack
Virtual Chassis Stacking summary information

ID  Pending ID  MAC address       Priority  Status  Role
1   -           0000.cd27.c4bf    128       Ready   Active master
2   -           0000.cd28.0801    128       Syncing Backup Member

Operational Status                Normal operation
Stack MAC address                 0000.cd27.c4bf
```

The **show stack** command is available on all stacking switches running the AlliedWare Plus operating system.

On x900 Series switches, on the front of the XEM-STK there is an indicator that displays the stack ID of the switch that the XEM is installed in.

On x600 and x610 Series switches, you can use the command:

```
awplus# show stack indicator <1-8>|all [time <1-500>]
```

This causes the **master** LED on the switch to flash in a sequence which indicates the stack ID number. The pattern will be a number of flashes in quick succession followed by a longer pause; where the number of flashes equals the stack member ID.

For example, on a four-switch stack with the stack IDs 1, 2, 3, and 4, the switch with stack ID:

- 1 will flash on and off without pausing * * * * * * * * * * *
- 2 will flash twice then pause * *   * *   * *   * *   * *   * *
- 3 will flash three times then pause * * *   * * *   * * *
- 4 will flash four times then pause * * * *   * * * *   * * * *

You can extend the time that the **master** LEDs will flash by up to 500 seconds to give you more time to reach the stack's location, by using the optional **time** parameter with the command. By default the LEDs will flash for 5 seconds.

Converting stacking ports to network ports on legacy switches

x610 Series

Reconfiguring x6EM/XS2 stacking module ports as network ports

The Stack module x6EM/XS2 contains two 10 GbE SFP+ ports. By default, these ports are configured for stacking. However, you can reconfigure them as network switch ports. To do this, reconfigure the switch as a non-stacked switch, using the command **no stack <*stack-ID*> enable**, then reboot to apply this configuration.

In the non-stacked mode these ports will appear as configurable switch ports, even without the SFP+ Stack module being inserted within the switch. When configuring these ports, you can identify them by their value 1 in the middle address component of the port number triplet. For example the port number 1.1.2 has the components:
1 (stack member identifier; always a 1 in non-stack mode) **.1** (the board identifier; always a 1 for the Stack module) **.2** (the port number on the Stack module: can have the value 1 or 2).

DC2552XS/L3 Series

Reconfiguring the front ports as network ports

On DC2552XS/L3 series switches, the front 4 ports can be configured as stacking ports or network ports. By **default**, the ports are configured as **stacking** ports.

To convert stacking ports to **network** ports, VCStack needs to be disabled using the following command:

```
no stack <stack-id> enable
```

Allied Telesis NETWORK SMARTER

North America Headquarters | 19800 North Creek Parkway | Suite 100 | Bothell | WA 98011 | USA | T: +1 800 424 4284 | F: +1 425 481 3895
Asia-Pacific Headquarters | 11 Tai Seng Link | Singapore | 534182 | T: +65 6383 3832 | F: +65 6383 3830
EMEA & CSA Operations | Incheonweg 7 | 1437 EK Rozenburg | The Netherlands | T: +31 20 7950020 | F: +31 20 7950021

alliedtelesis.com