System Settings

This chapter provides instructions for configuring the following StarOS options.

It is assumed that the procedures to initially configure the system as described in *Getting Started* have been completed.

Important The commands used in the configuration examples in this section are the most likely-used commands and/or keyword options. In many cases, other optional commands and/or keyword options are available. Refer to the *Command Line Interface Reference* for complete information.

- Configuring a Second Management Interface, on page 1
- Verifying and Saving Your Interface and Port Configuration, on page 2
- Configuring System Timing, on page 3
- Configuring SF Boot Configuration Pause, on page 7
- Enabling CLI Timestamping, on page 7
- Configuring CLI Confirmation Prompts, on page 8
- Configuring System Administrative Users, on page 10
- Configuring TACACS+ for System Administrative Users, on page 19
- IPv6 Address Support for TACACS+ Server, on page 23
- Separating Authentication Methods, on page 23
- Configuring a Chassis Key, on page 26
- Configuring MIO/UMIO/MIO2 Port Redundancy, on page 28
- Configuring Data Processing Card Availability, on page 31
- Enabling Automatic Reset of FSC Fabric, on page 32
- Configuring ASR 5500 Link Aggregation, on page 32
- Configuring a Demux Card, on page 39

Configuring a Second Management Interface

Refer to *Getting Started* for instructions on configuring a system management interface on the Management Input/Output (MIO/UMIO/MIO2) card. This section provides described how to configure a second management interface.

Use the following example to configure a second management interface:

```
configure
  context local
    interface interface_name
      ip address ipaddress subnetmask
      exit
    ip route 0.0.0.0 0.0.0.0 gw_address interface_name
    exit
  port ethernet slot#/port#
    bind interface interface_name local
    no shutdown
    media rj45
    end
```

Notes:

- For **port ethernet** *slot#*, use the actual chassis slot in which the active MIO/UMIO/MIO2 resides (slot number 5 or 6).
- Enter IP addresses using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.
- For **port ethernet** *port#*, use the physical port on the MIO/UMIO/MIO2 card – port 1 or 2.
- The MIO/UMIO/MIO2 is equipped with RJ-45 (1000Base-T copper) interfaces. The RJ-45 interfaces connect the system to the management network via CAT3 or CAT5 Ethernet cable.
- *Option:* In the Ethernet Port configuration mode, configure the port speed, if needed, by entering the **medium** command. Refer to the *Command Line Interface Reference* for a complete explanation of this command.
- In the **{ ip | ipv6 } route** command, other keyword options, instead of the gateway IP address, are available and include: **next-hop** IP address, **point-to-point**, and **tunnel**.

Verifying and Saving Your Interface and Port Configuration

Verify that your interface configuration settings are correct by entering the following command:

show ip interface

The output from this command should be similar to that shown below. In this example an interface named *mgmt2* was configured in the local context.

```
Intf Name:       mgmt2
Intf Type:       Broadcast
Description:     management2
VRF:             None
IP State:        UP (Bound to 5/2)
IP Address:      192.168.100.3        Subnet Mask:     255.255.255.0
Bcast Address:   192.168.100.255      MTU:             1500
Resoln Type:     ARP                  ARP timeout:     60 secs
L3 monitor LC-port switchover: Disabled
Number of Secondary Addresses: 0
```

Verify that the port configuration settings are correct by entering the following command:

show configuration port *slot#/port#*

slot# is the chassis slot number of the line card where the physical port resides. slot# is either 5 or 6. *port#* is the number of the port (either 1 or 2).

This following command produces an output similar to the one shown below. It displays the configuration of port 2 of the MIO/UMIO/MIO2 installed in chassis slot 5. In this example, the port is bound to an interface called *mgmt2*.

```
config
  port ethernet 5/2
    description management2
    no shutdown
    bind interface mgmt2 local
end
```

Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring System Timing

The system is equipped with a clock that supplies the timestamp for statistical counters, accounting records, logging, and event notification. After the initial configuration of the system clock, you can configure the system to communicate with one or more Network Time Protocol (NTP) server(s) to ensure that the clock is always accurate.

In the event of a power outage, the clock is maintained with an accuracy of ± one minute per month for up to 10 years. This ensures that when power is restored, the system is ready to process sessions and generate accounting, log, and event data with accurate timestamps.

In addition to configuring the timing source, you must configure the system's time zone.

Setting the System Clock and Time Zone

Use the following command example to configure the system clock and time zone:

```
clock set date:time
configure
   clock timezone timezone [ local ]
   end
```

Notes:

- Enter the date and time in the format YYYY:MM:DD:HH:mm or YYYY:MM:DD:HH:mm:ss.
- Refer to the online Help for the **clock timezone** command for a complete list of supported time zones.
- The optional **local** keyword indicates that the time zone specified is the local timezone.
- Daylight Savings Time is automatically adjusted for time zones supporting it.

Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Verifying and Saving Your Clock and Time Zone Configuration

Enter the following command to verify that you configured the time and time zone correctly:

```
show clock
```

The output displays the date, time, and time zone that you configured.

Configuring Network Time Protocol Support

This section provides information and instructions for configuring the system to enable the use of the Network Time Protocol (NTP).

Important Configure the system clock and time zone prior to implementing NTP support. This greatly reduces the time period that must be corrected by the NTP server.

Many of the services offered by the StarOS require accurate timekeeping derived through NTP. If the time reference(s) used by StarOS are not accurate, the services may be unreliable. For this reason it should be assumed that normal system operation requires that NTP be configured.

The system uses NTP to synchronize its internal clock to external time sources (typically GPS NTP sources, or other Stratum 2 or 3 servers, switches or routers).

By default, NTP is not enabled externally and should be configured when the system is initially installed. When enabled, the active MIO/UMIO/MIO2 will synchronize with external sources. If not enabled, the active MIO/UMIO/MIO2 will use its local clock as a time source. In the event of an NTP server or network outage, an already running MIO/UMIO/MIO2 will continue to use NTP to maintain time accuracy, but in a holdover mode.

All cards with CPUs synchronize to the active MIO/UMIO/MIO2 internally. This occurs even if an external NTP server is not configured. In the event of a MIO/UMIO/MIO2 switchover, all other cards will start synchronizing with the newly active MIO/UMIO/MIO2 automatically.

The system should have:

- NTP enabled.
- NTP configured for use in the local context only. Use of other contexts (which can be specified in the enable configurable) will cause issues.
- NTP configured for at least three external NTP servers. With three or more servers, outlyers and broken or misconfigured servers can be detected and excluded. Generally, the more servers the better (within reason).

Important Do not configure any external NTP servers using the **prefer** keyword. The NTP clock selection algorithms already have the built-in ability to pick the best server. Use of **prefer** usually results in a poorer choice than NTP can determine for itself.

Important Do not change the **maxpoll**, **minpoll**, or **version** keyword settings unless instructed to do so by Cisco TAC.

Use the following example to configure the necessary NTP association parameters:

```
configure
  ntp
```

```
enable
server ip_address1
server ip_address2
server ip_address3
end
```

Notes:

- By default *context_name* is set to *local*. This is the recommended configuration.
- A number of options exist for the **server** command. Refer to the *NTP Configuration Mode Commands* chapter in the *Command Line Interface Reference* for more information.
- Enter the IP address of NTP servers using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

Important Configure the system with at least three (preferably four) NTP servers.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring NTP Servers with Local Sources

NTP can use network peers, local external clocks (such as GPS devices), or a local clock with no external source.

A local clock with no external source is usually a last-resort clock when no better clock is available. It is typically configured on a site's intermediate NTP server so that when a WAN network outage occurs, hosts within the site can continue to synchronize amongst themselves.

You can configure this in ntpd or on many commercially available NTP devices. This local clock should always have a high stratum number (8+) so that under normal conditions (when real sources are available) this local clock will not be used.

Using a Load Balancer

The NTP daemon and protocol assume that each configured server is running NTP. If a NTP client is configured to synchronize to a load balancer that relays and distributes packets to a set of real NTP servers, the load balancer may distribute those packets dynamically and confuse the NTP client. NTP packets are latency and jitter sensitive. Relaying them through a load balancer can confuse the NTP client and is not a supported practice.

Verifying the NTP Configuration

Verify the NTP configuration is correct. Enter the following command at the Exec mode prompt:

```
show ntp associations
```

The output displays information about all NTP servers. See the output below for an example deploying two NTP servers.

```
+----Peer Selection: ( ) - Rejected / No Response
|                    (x) - False Tick
```

```
|                    (.) - Excess
|                    (-) - Outlyer
|                    (+) - Candidate
|                    (#) - Selected
|                    (*) - System Peer
|                    (o) - PPS Peer
v
     remote            refid      st t when poll reach   delay   offset  jitter
==============================================================================
*10.81.254.202   .GPS.            1 u  160 1024  377   21.516    0.019   0.009
```

The following table describes the parameters output by the **show ntp associations** command.

Table 1: NTP Parameters

Column Title	Description
remote	List of the current NTP servers. One of these characters precedes each IP address to show the server's current condition: • ( ) Rejected/No response • X False tick • . Excess • - Outlyer • + Candidate • # Selected • * System peer • (o) PPS peer
refid	Last reported NTP reference to which the server is synchronizing.
st	NTP server stratum level.
t	Communication type: broadcast, multicast, etc.
when	Number of seconds since the last contact.
poll	Polling interval between the system and the NTP server.
reach	Octal value of the reachability shift register indicating which responses were received for the previous eight polls to this NTP server.
delay	Round-trip delay (in milliseconds) for messages exchanged between the system and the NTP server.
offset	Number of milliseconds by which the system clock must be adjusted to synchronize it with the NTP server.

Column Title	Description
jitter	Jitter in milliseconds between the system and the NTP server.

Configuring SF Boot Configuration Pause

Under certain circumstances, within VPC-DI deployments, the CF applies the boot configuration before all SFs have completed their boot process.

The following Configuration Mode command, **wait cards active**, pauses configuration until all specified cards are operational or the timeout period expires (whichever criteria is met first). The pause occurs immediately following local management context creation and ntp/snmp configuration.

This command corrects a scenario where SFs come online late following chassis load or reload and the configuration pertaining to those SFs is not applied (and thereby lost).

```
configure
  [ no ] wait cards active { all | number } [ standby number ] timeout seconds

  end
```

Notes:

- **all**: Pause until all active mode cards attain operational status.
- *number* : Pause until the specified number of active mode cards attain operational status.*number* is 0 through the number of active mode cards.
- **standby** *number* : (Optional) Also wait for the specified number of non-active mode cards to attain operational status.

  number is 0 through the number of service slots not configured for active mode SFs.
- **timeout** *seconds*: Wait from 1 through 3600 *seconds* for the specified card set to attain operational status. The wait is terminated early when or if this condition is satisfied. Otherwise the wait is terminated when the timeout period expires.

The following example command instructs the system to wait up to 120 seconds for all active cards and 1 standby card to become active:

wait cards active all standby 1 timeout 120

Enabling CLI Timestamping

To display a timestamp (date and time) for every command that is executed on the CLI, enter the following command at the root prompt for the Exec mode:

```
timestamps
```

The date and time appear immediately after you execute the command.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring CLI Confirmation Prompts

A number of Exec mode and Global Configuration mode commands prompt users for a confirmation (Are you sure? [Yes|No]:) prior to executing the command.

This section describes configuration settings that:

- Automatically confirm commands for the current CLI session (Exec mode) or for all CLI sessions and users (Global Configuration mode).
- Requires confirmation prompting only for the Exec mode **configure** command and **autoconfirm** command.
- Selectively requires confirmation of Exec mode configuration commands.

Enabling Automatic Confirmation

You can use the **autoconfirm** command to disable confirmation prompting for configuration commands. The **autoconfirm** command is available in the Exec mode and Global Configuration mode. Enabling the autoconfirm feature automatically supplies a "Yes" response to configuration command prompts, including for critical commands such as **reload** and **shutdown**. By default autoconfirm is disabled.

In the Exec mode, autoconfirm applies only to the current interactive CLI session.

In the Global Configuration mode, autoconfirm applies to all CLI sessions for all CLI users:

```
configure
   autoconfirm
   end
```

To disable autoconfirm once it has been enabled, use the **no autoconfirm** command.

Important If commandguard is enabled, autoconfirm will disable commandguard.

Autoconfirm is intended as an "ease-of-use" feature. It presumes that the answer to "Are you sure? [Y/N]" prompts will be "Yes", and skips the prompt. Its use implies that the user is an expert who does not need these "safety-net" prompts.

Requiring Confirmation for autoconfirm and configure Commands

You can require confirmation prompting for the **autoconfirm** (Exec mode and Global Configuration mode) and **configure** (Exec mode) commands via the Global Configuration mode **commandguard** command.

Important If autoconfirm is enabled, commandguard will not take effect until autoconfirm is disabled in both Exec and Global Configuration modes.

The following command sequence enables the commandguard feature:

```
configure
   commandguard
   end
```

With commandguard enabled the confirmation prompt appears as shown in the example below:

```
[local]host_name# configure
Are you sure? [Yes|No]: yes
[local]host_name(config)#
```

To disable commandguard once it has been enabled, use the **no commandguard** command.

The status of **commandguard** is output in **show configuration** commands.

Requiring Confirmation for Specific Exec Mode Commands

A keyword for the **commandguard** command allows you to apply mandatory prompting for specified categories of Exec mode configuration commands, even when autoconfirm is enabled.

The command syntax is as follows:

```
configure
   commandguard exec-command exec_mode_category
   end
```

Notes:

- **exec-command** *exec_mode_category* specifies one of the following categories of Exec mode configuration commands.
  - card
  - clear
  - copy
  - debug
  - delete
  - filesystem
  - hd
  - reload
  - rename
  - shutdown
  - task
  - upgrade
- You can enter multiple **commandguard exec-command** *exec_mode_category* commands.
- All Exec mode commands beginning with the specified category word will prompt for confirmation, regardless if autoconfirm is enabled.
- You can turn off confirmation prompting for a specific category using **no commandguard exec-command** *exec_mode_category*.
- If autoconfirm is overridden by **commandguard exec-command** for an Exec mode command, StarOS displays an informational message indicating why autoconfirm is being overridden when you attempt to execute the command.
- Users may selectively override confirmation prompting for any Exec mode configuration command that supports the **-noconfirm** keyword.

For example, with **commandguard exec-command card** enabled, the confirmation prompt appears as shown below:

```
[local]host_name# card busy-out 1
Info: commandguard prevents autoconfirm of this command
```

```
Are you sure? [Yes|No]: yes
[local]host_name#
```

Configuring System Administrative Users

Getting Started describes how to configure a context-level security administrator for the system.

This section provides instructions for configuring additional administrative users having the following privileges:

- **Security Administrators**: have read-write privileges and can execute all CLI commands, including those available to Administrators, Operators, and Inspectors
- **Administrators**: have read-write privileges and can execute any command in the CLI except for a few security-related commands that can only be configured by Security Administrators. Administrators can configure or modify system settings and execute all system commands, including those available to the Operators and Inspectors.
- **Operators**: have read-only privileges to a larger subset of the Exec Mode commands. They can execute all commands that are part of the inspector mode, plus some system monitoring, statistic, and fault management functions. Operators do not have the ability to enter the Config Mode.
- **Inspectors**: are limited to a few read-only Exec Mode commands. The bulk of these are **show** commands for viewing a variety of statistics and conditions. An Inspector cannot execute **show configuration** commands and does not have the privilege to enter the Config Mode.

Configuration instructions are categorized according to the type of administrative user: context-level or local-user.

Important For information on the differences between these user privileges and types, refer to *Getting Started*.

User Name Character Restrictions

User names can only contain alphanumeric characters (a-z, A-Z, 0-9), hyphen, underscore, and period. The hyphen character cannot be the first character. This applies to AAA user names as well as local user names.

If you attempt to create a user name that does not adhere to these standards, you will receive the following message: "Invalid character; legal characters are "0123456789.-_abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ".

Configuring Context-level Administrative Users

This user type is configured at the context-level and relies on the AAA subsystems for validating user names and passwords during login. This is true for both administrative user accounts configured locally through a configuration file or on an external RADIUS or TACACS+ server. Passwords for these user types are assigned once and are accessible in the configuration file.

This section contains information and instructions for configuring context-level administrative user types.

It is possible to configure the maximum number of simulations CLI sessions on a per account or per authentication method basis. It will protect certain accounts that may have the ability to impact security configurations and attributes or could adversely affect the services, stability and performance of the system. The maximum number of simultaneous CLI sessions is configurable when attempting a new Local-User login and a new AAA context-based login. If the maximum number of sessions is set to 0, then the user is authenticated regardless of the login type. When the CLI task starts, a check is complete to identify the count. In this case, the CLI determines that the sessions for that user is 1 which is greater than 0 and it will display an error message in the output, it generate starCLIActiveCount and starCLIMaxCount SNMP MIB Objects and starGlobalCLISessionsLimit and starUserCLISessionsLimit SNMP MIB Alarms.

The **max-sessions** keyword for the **local-user username** *Global Configuration Mode* command configures the maximum number of simultaneous sessions available for a local user.

The **max-sessions** *Context Configuration Mode* command allows administrative users to configure the maximum simultaneous sessions allowed for corresponding users.

Refer to the *Command Line Interface Reference* for detailed information about these commands.

Configuring Context-level Security Administrators

Use the example below to configure additional security administrators:

```
configure
  context local
    administrator user_name { [ encrypted ] [ nopassword ] password password
 }
    end
```

Notes:

- Additional keyword options are available that identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **administrator** command.
- The **nopassword** option allows you to create an administrator without an associated password. Enable this option when using ssh public keys (**authorized key** command in SSH Configuration mode) as a sole means of authentication. When enabled this option prevents someone from using an administrator password to gain access to the user account.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Context-level Administrators

Use the example below to configure context-level configuration administrators:

```
configure
  context local
    config-administrator user_name { [ encrypted ] [ nopassword ] password
 password }
    end
```

Notes:

- Additional keyword options are available that identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **config-administrator** command.

- The **nopassword** option allows you to create a config-administrator without an associated password. Enable this option when using ssh public keys (**authorized key** command in SSH Configuration mode) as a sole means of authentication. When enabled this option prevents someone from using a config-administrator password to gain access to the user account.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Context-level Operators

Use the example below to configure context-level operators:

```
configure
  context local
    operator user_name { [ encrypted ] [ nopassword ] password password }
    end
```

Notes:

- Additional keyword options are available that identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **operator** command.
- The **nopassword** option allows you to create an operator without an associated password. Enable this option when using ssh public keys (**authorized key** command in SSH Configuration mode) as a sole means of authentication. When enabled this option prevents someone from using an operator password to gain access to the user account.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Context-level Inspectors

Use the example below to configure context-level inspectors:

```
configure
  context local
    inspector user_name { [ encrypted ] [ nopassword ] password password }
    end
```

Notes:

- Additional keyword options are available that identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **inspector** command.
- The **nopassword** option allows you to create an inspector without an associated password. Enable this option when using ssh public keys (**authorized key** command in SSH Configuration mode) as a sole means of authentication. When enabled this option prevents someone from using an inspector password to gain access to the user account.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring LI Administrators

Important For security reasons, **li-administration** accounts must be restricted for use only with Lawful Intercept (LI) functionality and not for general system administration. Only security administrators and administrators can provision LI privileges. To ensure security in accordance with Law Enforcement Agency (LEA) standards, LI administrative users must access the system using the Secure Shell (SSH) protocol only. LI privileges can be optionally configured for use within a single context system-wide. For additional information, see the *Lawful Intercept Configuration Guide* and Provisioning Lawful Intercept, on page 17.

Use the example below to configure a context-level LI administrator:

```
configure
   context context_name
      administrator user_name { [ encrypted ] [ nopassword ] password password
 li-administrator}
     end
```

LI Administrators and non-LI Administrators can configure Lawful-Intercept CLI commands. However, only LI Administrators can view the encrypted Lawful-Intercept CLI commands in Trusted Builds and in Normal builds, if the Global Configuration mode **require segregated li-configuration** command is enabled. For additional information, see the *Lawful Intercept Configuration Guide* and Segregating System and LI Configurations, on page 13 .

Segregating System and LI Configurations

Lawful Intercept (LI) configuration includes sensitive information. By default in a Normal build, an administrator without li-administration privilege can view the LI configuration commands. However, display of the LI configuration commands can be restricted or segregated from the rest of the system configuration.

The Global Configuration mode **require segregated li-configuration** command permanently segregates display of System and Lawful Intercept CLI. The CLI commands with Lawful-Intercept keyword are encrypted and can only be viewed by an administrator with li-administration privilege.

Important In a Trusted build, LI segregation is turned on and cannot be disabled. The **require segregated li-configuration** command is invisible.

Segregating LI configuration from system configuration has the following impacts on StarOS:

- Only administrators with li-administration privilege can see Lawful Intercept CLI commands in the output of the **show configuration** command.
- Executing the **save configuration** command will automatically encrypt Lawful Intercept CLI configuration commands.
- When loading a saved configuration file via CLI command (for example, **configure** <*url*>), encrypted Lawful Intercept CLI commands will be decrypted and executed only for an administrator with LI privilege. For an administrator without LI privilege, encrypted Lawful Intercept CLI commands will not be decrypted and executed.

- During a system boot wherein the boot config is loaded, encrypted Lawful Intercept configuration will be decrypted and loaded silently, in other words Lawful Intercept CLI configuration will not be visible on the console port.
- The Exec mode **configure** command now supports a keyword that allows an LI administrator to load only encrypted Lawful Intercept configuration from a saved configuration file (for example, **configure encrypted** <*url*>). The **encrypted** keyword can only be executed by an LI Administrator.
- If you are running a system with encrypted Lawful Intercept configuration (segregated LI), the output of the **show boot initial-config** command contains a line indicating whether it needed to run the second pass or not during the initial boot. This line displays "encrypted li" if the encrypted Lawful Intercept configuration was processed. If the line reads "encrypted li errors" then the second pass was not successful, or gave some output which was not expected or informational in nature.
- A user with li-administration privileges can view the boot config output for the encrypted Lawful Intercept configuration with the **show logs encrypted-li** command.

For a detailed description of the Global Configuration mode **require segregated li-configuration** and associated commands, see the *Lawful Intercept CLI Commands* appendix in the *Lawful Intercept Configuration Guide*.

Note The *Lawful Intercept Configuration Guide* is not available on www.cisco.com. Contact your Cisco account representative to obtain a copy of this guide.

In Release 21.4 and higher (Trusted builds only):

- Users can only access the system through their respective context interface.
- If the user attempts to log in to their respective context through a different context interface, that user will be rejected.
- Irrespective of whether the users are configured in any context with 'authorized-keys' or 'allowusers', with this feature these users will be rejected if they attempt to log in via any other context interface other than their own context interface.
- Users configured in any non-local context are required to specify which context they are trying to log in to. For example:

  ssh username@ctx_name@ctx_ip_addrs

Verifying Context-level Administrative User Configuration

Verify that the configuration was successful by entering the following command:

show configuration context local

This command displays all of the configuration parameters you modified within the Local context during this session. The following displays sample output for this command. In this example, a security administrator named *testadmin* was configured.

```
config
  context local
    interface mgmt1
      ip address 192.168.1.10 255.255.255.0
    #exit
    subscriber default
```

```
    #exit
    administrator testadmin encrypted password fd01268373c5da85
    inspector testinspector encrypted password 148661a0bb12cd59
exit
  port ethernet 5/1
    bind interface mgmt1 local
  #exit
```

Configuring Local-User Administrative Users

The local user type supports ANSI T1.276-2003 password security protection. Local-user account information, such as passwords, password history, and lockout states, is maintained in /flash. This information is saved immediately in a separate local user database subject to AAA based authentication and is not used by the rest of the system. As such, configured local-user accounts are not visible with the rest of the system configuration.

Important In release 20.0 and higher Trusted StarOS builds, the local user database is disabled. The Global Configuration mode **local-user** commands, and Exec mode **show local-user** and **update local-user** commands are unavailable. For additional information on Trusted builds, see the *System Operation and Configuration* chapter.

Use the example below to configure local-user administrative users:

```
configure
   local-user username name
   end
```

Notes:

- Additional keyword options are available identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **local-user username** command.

For additional information on the local-user database, see Updating and Downgrading the local-user Database, on page 16.

Verifying Local-User Configuration

Verify that the configuration was successful by entering the following command:

```
show local-user verbose
```

This command displays information on configured local-user administrative users. A sample output for this command appears below. In this example, a local-user named *SAUser* was configured.

```
Username:                SAUser
  Auth Level:            secadmin
  Last Login:            Never
  Login Failures:        0
  Password Expired:      Yes
  Locked:                No
  Suspended:             No
  Lockout on Pw Aging:   Yes
  Lockout on Login Fail: Yes
```

Updating Local-User Database

Update the local-user (administrative) configuration by running the following Exec mode command. This command should be run immediately after creating, removing or editing administrative users.

```
update local-user database
```

Updating and Downgrading the local-user Database

Prior to release 20.0, local-user passwords were hashed with the MD5 message digest-algorithm and saved in the local-user database. In release 20. 0, PBKDF2 (Password Based Key Derivation Function - Version 2) is now used to derive a key of given length, based on entered data, salt and number of iterations. Local-user account passwords are hashed using the PBKDF2 method with a randomly generated salt coupled with a large number of iterations to make password storage more secure.

When upgrading to release 20.0, existing user passwords in the local-user database are not automatically upgraded from MD5 to PBKDF2 hashing (only hashed password values are stored). Since hash functions are one-way, it is not possible to derive user passwords from the stored hash values. Thus it is not possible to convert existing hashed passwords to strongly hashed passwords automatically.

To update the database, a Security Administrator must run the Exec mode **update local-user database** CLI command. When this command is executed, StarOS reads the database from the /flash directory, reconstructs the database in the new format, and writes it back to the disk.

The database upgrade process does not automatically convert MD5 hashed passwords into the PBKDF2 format. StarOS continues to authenticate users using the old encryption algorithm. It flags the users using the old encryption algorithm with a "Weak Hash" flag. This flag appears in the output of the **show local-user [verbose]** Exec mode CLI command. When users re-login with their credentials, StarOS verifies the entered password using the MD5 algorithm, then creates a new hash using the PBKDF2 algorithm and then saves the result in the database. StarOS then clears the "Weak Hash" flag for that user.

Important Since hash functions are one-way, it is not possible to convert PBKDF2 hashed passwords to the MD5 format. The local-user database must be downgraded prior to reverting to StarOS releases prior to 20.0.

To downgrade the local-user database to use the MD5 hash algorithm, a Security Administrator must run the Exec mode **downgrade local-user database** command. StarOS prompts for confirmation and requests the Security Administrator to reenter a password. The entered password re-authenticates the user prior to executing the downgrade command. After verification, the password is hashed using the appropriate old/weak encryption algorithm and saved in the database to allow earlier versions of StarOS to authenticate the Security Administrator.

The downgrade process does not convert PBKDF2 hashed passwords to MD5 format. The downgrade process re-reads the database (from the /flash directory), reconstructs the database in the older format, and writes it back to the disk. Since the PBKDF2 hashed passwords cannot be converted to the MD5 hash algorithm, and earlier StarOS releases cannot parse the PBKDF2 encryption algorithm, StarOS suspends all those users encrypted via the PBKDF2 algorithm. Users encrypted via the MD5 algorithm ("Weak Hash" flag) can continue to login with their credentials. After the system comes up with the earlier StarOS release, suspended users can be identified in the output of the **show local-user [verbose]**command.

To reactivate suspended users a Security Administrator can:

- Set temporary passwords for suspended users, using the Exec mode **password change local-user** *username* command.

- Reset the suspend flag for users, using the Configuration mode **no suspend local-user** *username* command.

Provisioning Lawful Intercept

Lawful Intercept (LI) functionality allows a network operator to intercept control and data messages to and from targeted mobile users. Accompanied by a court order or warrant, a Law Enforcement Agency (LEA) initiates a request for the network operator to start the interception for a particular mobile user.

There are different standards followed for Lawful Intercept in different countries. The *LI Configuration Guide* describes how the feature works as well as how to configure and monitor the feature for each of the StarOS services that support Lawful Intercept. This guide is not available on www.cisco.com. It can only be obtained by contacting your Cisco account representative.

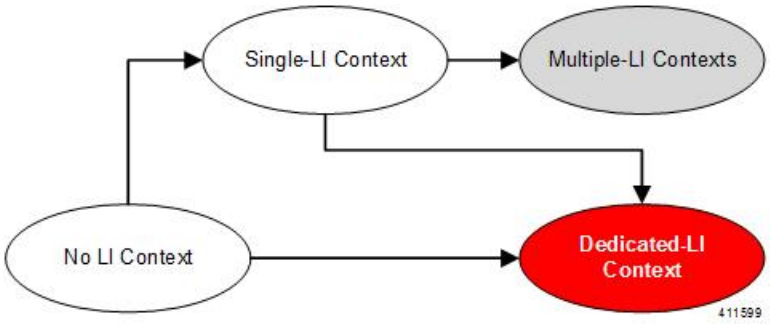
Security-related limitations on Lawful Intercept provisioning are described in *Lawful Intercept Restrictions* section of the *System Security* chapter.

LI can be provisioned within one or more StarOS contexts. An administrative user with **li-administration** privilege is associated with the context(s) that support LI capability. That administrator has access to the CLI configuration commands that provision LI functionality.

There are several types of LI configurations supported within a StarOS system configuration.

- **No LI Context** – The LI configuration was never entered for any context.
- **Single LI Context** – The LI configuration was entered within one context, but was never been entered within any other context. In this state, the Single LI Context can be converted to Multiple LI contexts if another context is configured with an LI configuration, or this context can be converted into the Dedicated-LI context by entering the Context Configuration mode **dedicated-li** command.
- **Multiple LI Contexts** – Two or more contexts have been configured with the LI configuration. A Multiple-LI context configuration can never be re-configured as any other type of LI configuration.
- **Dedicated LI Context** – If the existing system configuration is a No LI Context or a Single LI Context system, it can be converted to a Dedicated-LI Context system by entering the Context Configuration mode **dedicated-li** command. A Dedicated LI context limits access to the LI configuration to the one VPN context which requires it. Once configured as a Dedicated-LI context system, it can never be re-configured any other type of LI context system. Refer to the *Lawful Intercept Configuration Guide* before attempting to create a Dedicated-LI context.

Figure 1: LI Context Configurations



In Release 21.4 and higher (Trusted builds only):

- Users can only access the system through their respective context interface.

- If the user attempts to log in to their respective context through a different context interface, that user will be rejected.
- Irrespective of whether the users are configured in any context with 'authorized-keys' or 'allowusers', with this feature these users will be rejected if they attempt to log in via any other context interface other than their own context interface.
- Users configured in any non-local context are required to specify which context they are trying to log in to. For example:

  ssh username@ctx_name@ctx_ip_addrs

Restricting User Access to a Specified Root Directory

By default an admin user who has FTP/SFTP access can access and modify any files under the /mnt/user/ directory. Access is granted on an "all-or-nothing" basis to the following directories: /flash, /cdrom, /hd-raid, /records, /usb1 and /usb2.

An administrator or configuration administrator can create a list of SFTP subsystems with a file directory and access privilege. When a local user is created, the administrator assigns an SFTP subsystem. If the user's authorization level is not security admin or admin, the user can only access the subsystem with read-only privilege. This directory is used as the user's root directory. The information is set as environmental variables passed to the openssh sftp-server.

You must create the SFTP root directory before associating it with local users, administrators and config administrators. You can create multiple SFTP directories; each directory can be assigned to one or more users.

Configuring an SFTP root Directory

The **subsystem sftp** command allows the assignment of an SFTP root directory and associated access privilege level.

```
configure
   context local
      server sshd
         subsystem sftp  [ name sftp_name root-dir pathname mode { read-only
 | readwrite } ]
```

Notes:

- *sftp_name* is an alphanumeric string that uniquely identifies this subsystem.
- *pathname* specifies the root directory to which SFTP files can be transferred. Options include:
  - /hd-raid/records/cdr
  - /flash

Associating an SFTP root Directory with a Local User

The **local-user username** command allows an administrator to associate an SFTP root directory with a specified username.

```
configure
   local-user username user_name authorization-level level ftp sftp-server
```

```
sftp_name password password
  exit
```

Associating an SFTP root Directory with an Administrator

The **administrator** command allows an administrator to associate an SFTP root directory for a specified administrator.

```
configure
   context local
     administrator user_name password password ftp sftp-server sftp_name
     exit
```

Associating an SFTP root Directory with a Config Administrator

The config-administrator command allows an administrator to associate an SFTP root directory with a specified configuration administrator.

```
configure
   context local
     config-administrator user_name password password ftp sftp-server sftp_name
     exit
```

Configuring TACACS+ for System Administrative Users

This section describes TACACS+ (Terminal Access Controller Access Control System+) AAA (Authentication Authorization and Accounting) service functionality and configuration on the ASR 5500.

Operation

TACACS+ is a secure, encrypted protocol. By remotely accessing TACACS+ servers that are provisioned with the administrative user account database, the ASR 5500 system can provide TACACS+ AAA services for system administrative users. TACACS+ is an enhanced version of the TACACS protocol that uses TCP instead of UDP.

The system serves as the TACACS+ Network Access Server (NAS). As the NAS the system requests TACACS+ AAA services on behalf of authorized system administrative users. For the authentication to succeed, the TACACS+ server must be in the same local context and network accessed by the system.

The system supports TACACS+ multiple-connection mode. In multiple-connection mode, a separate and private TCP connection to the TACACS+ server is opened and maintained for each session. When the TACACS+ session ends, the connection to the server is terminated.

TACACS+ is a system-wide function on the ASR 5500. TACACS+ AAA service configuration is performed in TACACS Configuration Mode. Enabling the TACACS+ function is performed in the Global Configuration Mode. The system supports the configuration of up to three TACACS+ servers.

Once configured and enabled on the system, TACACS+ authentication is attempted first. By default, if TACACS+ authentication fails, the system then attempts to authenticate the user using non-TACACS+ AAA services, such as RADIUS.

It is possible to configure the maximum number of simulations CLI sessions on a per account or per authentication method basis. It will protect certain accounts that may have the ability to impact security

configurations and attributes or could adversely affect the services, stability and performance of the system. The maximum number of simultaneous CLI sessions is configurable when attempting a new TACACS+ user login. The recommendation is to use the max-sessions feature is through the TACACS+ server attribute option **maxsess**. The second way is though the StarOS CLI configuration mode TACACS+ mode using the **maxsess** keyword in the **user-id** command. If the maximum number of sessions is set to 0, then the user is authenticated regardless of the login type. When the CLI task starts, a check is complete to identify the count. In this case, the CLI determines that the sessions for that user is 1 which is greater than 0 and it will display an error message in the output, it generate starCLIActiveCount and starCLIMaxCount SNMP MIB Objects and starGlobalCLISessionsLimit and starUserCLISessionsLimit SNMP MIB Alarms.

The **max-sessions** *TACACS+ Configuration Mode command* configures the maximum number of sessions available for TACACS+. Also the **default** option for the **user-id** *TACACS+ Configuration Mode command* configures the default attributes for a specific TACACS+ user identifier. Refer to the *Command Line Interface Reference* for detailed information about these commands.

Important The user can define the maximum number of simulations CLI sessions available in both the StarOS and TACACS+ server configuration. However, this option is extremely discouraged.

Important *For releases after 15.0 MR4,* TACACS+ accounting (CLI event logging) will not be generated for Lawful Intercept users with privilege level set to 15 and 13.

User Account Requirements

Before configuring TACACS+ AAA services, note the following TACACS+ server and StarOS user account provisioning requirements.

TACACS+ User Account Requirements

The TACACS+ server must be provisioned with the following TACACS+ user account information:

- A list of known administrative users.
- The plain-text or encrypted password for each user.
- The name of the group to which each user belongs.
- A list of user groups.
- TACACS+ privilege levels and commands that are allowed/denied for each group.

Important TACACS+ privilege levels are stored as Attribute Value Pairs (AVPs) in the network's TACACS+ server database. Users are restricted to the set of commands associated with their privilege level. A mapping of TACACS+ privilege levels to StarOS CLI administrative roles and responsibilities is provided in the table below.

To display the default mapping of TACACS+ privilege levels to CLI administrative roles, run the Exec mode **show tacacs priv-lvl** command. The default mapping varies based on the StarOS release and build type.

TACACS+ priv-levels can be reconfigured from their default StarOS authorization values via the TACACS+ Configuration mode **priv-lvl** and **user-id** commands. For additional information, see the *TACACS+ Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Important In release 20.0 and higher Trusted StarOS builds, FTP is not supported.

StarOS User Account Requirements

TACACS+ users who are allowed administrative access to the system must have the following user account information defined in StarOS:

- username
- password
- administrative role and privileges

Important For instructions on defining users and administrative privileges on the system, refer to *Configuring System Administrative Users*.

Configuring TACACS+ AAA Services

This section provides an example of how to configure TACACS+ AAA services for administrative users on the system.

Caution When configuring TACACS+ AAA services for the first time, the administrative user must use non-TACACS+ services to log into the StarOS. Failure to do so will result in the TACACS+ user being denied access to the system.

Log in to the system using non-TACACS+ services.

Use the example below to configure TACACS+ AAA services on the system:

```
configure
   tacacs mode
      server priority priority_number ip-address tacacs+srvr_ip_address
      end
```

Note:

- **server priority** *priority_number*: Must be an integer from 1 to 3 (*releases prior to 18.2*) or 1 through 4 (*releases 18.2+*), that specifies the order in which this TACACS+ server will be tried for TACACS+ authentication. 1 is the highest priority, and 3 or 4 is the lowest. The priority number corresponds to a configured TACACS+ server.
- **ip-address**: Must be the IPv4 address of a valid TACACS+ server that will be used for authenticating administrative users accessing this system via TACACS+ AAA services.

- By default, the TACACS+ configuration will provide authentication, authorization, and accounting services.

Enable TACACS+ on the StarOS:

```
configure
   aaa tacacs+
   end
```

For additional information, see Disable TACACS+ Authentication for Console, on page 23.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Important For complete information on all TACACS+ Configuration Mode commands and options, refer to the *TACACS Configuration Mode Commands* chapter in the *Command Line Reference*.

Configuring TACACS+ for Non-local VPN Authentication

By default TACACS+ authentication is associated with login to the local context. TACACS+ authentication can also be configured for non-local context VPN logins. TACACS+ must configured and enabled with the option described below.

A **stop** keyword option is available for the TACACS+ Configuration mode **on-unknown-user** command. If TACACS+ is enabled with the command-keyword option, the VPN context name into which the user is attempting a login must match the VPN name specified in the username string. If the context name does not match, the login fails and exits out.

Without this option the login sequence will attempt to authenticate in another context via an alternative login method. For example, without the **on-unknown-user stop** configuration, an admin account could log into the local context via the non-local VPN context. However, with the **on-unknown-user stop** configuration, the local context login would not be attempted and the admin account login authentication would fail.

```
configure
   tacacs mode
      on-unkown-user stop &quest;
      end
```

Verifying the TACACS+ Configuration

This section describes how to verify the TACACS+ configuration.

Log out of the system CLI, then log back in using TACACS+ services.

Important Once TACACS+ AAA services are configured and enabled on the StarOS, the system first will try to authenticate the administrative user via TACACS+ AAA services. By default, if TACACS+ authentication fails, the system then continues with authentication using non-TACACS+ AAA services.

At the Exec Mode prompt, enter the following command:

```
show tacacs [ client | priv-lvl | session | summary ]
```

The output of the **show tacacs** commands provides summary information for each active TACACS+ session such as username, login time, login status, current session state and privilege level. Optional filter keywords provide additional information.

An example of this command's output is provided below. In this example, a system administrative user named *asradmin* has successfully logged in to the system via TACACS+ AAA services.

```
active session #1:
   login username              : asradmin
   login tty                   : /dev/pts/1
   time of login               : Fri Oct 22 13:19:11 2011
   login server priority       : 1
   current login status        : pass
   current session state       : user login complete
   current privilege level     : 15
   remote client application   : ssh
   remote client ip address    : 111.11.11.11
   last server reply status    : -1
total TACACS+ sessions         : 1
```

Important For details on all TACACS+ maintenance commands, refer to the *Command Line Interface Reference*.

IPv6 Address Support for TACACS+ Server

Separating Authentication Methods

You can configure separate authentication methods for accessing the Console port and establishing SSH/telnet sessions (vty lines).

If you configure TACACS+ globally, access to the Console and vty lines are both authenticated using that method.

Since the Console port is a last resort access to StarOS, you can configure local authentication for the Console and employ TACACS+ for the vty lines.

Important This feature extends to AAA (Authentication, Authorization and Accounting) service as well as local users. For example, local-users may have only Console access and AAA (VPN context) users with access only via vty lines.

Separating authentication methods (Console versus vty lines) requires disabling Console access for users based on the type of authentication.

Disable TACACS+ Authentication for Console

A **noconsole** keyword for the Global Configuration mode **aaa tacacs+** command disables TACACS+ authentication on the Console line.

```
configure
  aaa tacacs+ noconsole
  exit
```

By default, TACACS+ server authentication is performed for login from a Console or vty line. With **noconsole** enabled, TACACS+ authentication is bypassed in favor of local database authentication for a console line; on vty lines, TACACS+ remains enabled.

Important When **aaa tacacs+ noconsole** is configured, a local user with valid credentials can log into a Console port even if **on-authen-fail stop** and **on-unknown-user stop** are enabled via the TACACS+ Configuration mode. If the user is not a TACACS+ user, he/she cannot login on a vty line.

Disable AAA-based Authentication for Console

A **noconsole** keyword for the Global Configuration mode **local-user allow-aaa-authentication** command disables AAA-based authentication on the Console line.

```
configure
  local-user allow-aaa-authentication noconsole
  exit
```

Since local-user authentication is always performed before AAA-based authentication and **local-user allow-aaa-authentication noconsole** is enabled, the behavior is the same as if **no local-user allow-aaa-authentication** is configured. There is no impact on vty lines.

Important This command does not apply for a Trusted build because the local-used database is unavailable.

Disable TACACS+ Authentication at the Context Level

When you enable **aaa tacacs+** in the Global Configuration mode, TACACS+ authentication is automatically applied to all contexts (local and non-local). In some network deployments you may wish to disable TACACS+ services for a specific context(s).

You can use the **no aaa tacacs+** Context Configuration command to disable TACACS+ services within a context.

```
configure
  context ctx_name
    no aaa tacacs+
```

Use the **aaa tacacs+** Context Configuration command to enable TACACS+ services within a context where it has been previously disabled.

Important AAA TACACS+ services must be enabled in the Global Configuration mode (all contexts) before you can selectively disable the services at the context level. You cannot selectively enable TACACS+ services at the context level when it has not been enabled globally.

Limit local-user Login on Console/vty Lines

As a security administrator when you create a StarOS user you can specify whether that user can login through the Console or vty line. The **[noconsole | novty]** keywords for the Global Configuration mode **local-user username** command support these options.

```
configure
  local-user username <username> [ noconsole | novty ]
  exit
```

The **noconsole** keyword prevents the user from logging into the Console port. The **novty** keyword prevents the user from logging in via an SSH or telnet session. If neither keyword is specified access to both Console and vty lines is allowed.

Important Use of the **noconsole** or **novty** keywords is only supported on the new local-user database format. If you have not run **update local-user database**, you should do so before enabling these keywords. Otherwise, **noconsole** and **novty** keywords will not be saved in the local-user database. After a system reboot, all users will still be able to access the Console and vty lines. For additional information, see the Updating and Downgrading the local-user Database, on page 16.

Important This command does not apply for a Trusted build because the local-used database is unavailable.

Limit Console Access for AAA-based Users

AAA-based users normally login through on a vty line. However, you may want to limit a few users to accessing just the Console line. If you do not use the local-user database (or you are running a Trusted build), this needs to be done by limiting access to the Console line for other AAA-based users. Enable the **noconsole** keyword for all levels of admin users that will not have access to the Console line.

The **noconsole** keyword is available for the Context Configuration mode commands shown below.

```
configure
  context <ctx_name>
    administrator <username> { encrypted | nopassword | password } noconsole

    config-administrator <username> { encrypted | nopassword | password }
noconsole
    inspector <username> { encrypted | nopassword | password } noconsole
    operator <username> { encrypted | nopassword | password } noconsole
    exit
```

The **noconsole** keyword disables user access to the Console line. By default **noconsole** is not enabled, thus all AAA-based users can access the Console line.

Important The **local-user allow-aaa-authentication noconsole** command takes precedence. In that case, all AAA-based users cannot access the Console line.

Verify Configuration Changes

You can verify changes made related to the separation of authentication methods via the Exec mode **show configuration** command. After saving the configuration changes, run **show configuration |grep noconsole** and **show configuration |grep novty**. The output of these commands will indicate any changes you have made.

Configuring a Chassis Key

A chassis key should be configured for each system. This key is used to decrypt encrypted passwords found in configuration files.

Overview

The chassis key is used to encrypt and decrypt encrypted passwords in the configuration file. If two or more chassis are configured with the same chassis key value, the encrypted passwords can be decrypted by any of the chassis sharing the same chassis key value. As a corollary to this, a given chassis key value will not be able to decrypt passwords that were encrypted with a different chassis key value.

The chassis key is used to generate the chassis ID which is stored in a file and used as the master key for protecting sensitive data (such as passwords and secrets) in configuration files

For release 15.0 and higher, the chassis ID is an SHA256 hash of the chassis key. The chassis key can be set by users through a CLI command or via the Quick Setup Wizard. If the chassis ID does not exist, a local MAC address is used to generate the chassis ID.

For release 19.2 and higher, the user must explicitly set the chassis key through the Quick Setup Wizard or CLI command. If it is not set, a default chassis ID using the local MAC address will not be generated. In the absence of a chassis key (and hence the chassis ID), sensitive data will not appear in a saved configuration file. The chassis ID is the SHA256 hash (encoded in base36 format) of the user entered chassis key plus a 32-byte secure random number. This assures that the chassis key and chassis ID have 32-byte entropy for key security.

If a chassis ID is not available encryption and decryption for sensitive data in configuration files will not work.

Configuring a New Chassis Key Value

CLI Commands

Important Only a user with Security Administrator privilege can execute the **chassis key value** and **chassis keycheck** commands.

Use the Exec mode **chassis key value** *key_string* command to enter a new chassis key.

The *key_string* is an alphanumeric string of 1 through 16 characters. The chassis key is stored as a one-way encrypted value, much like a password. For this reason, the chassis key value is never displayed in plain-text form.

The Exec mode **chassis keycheck** *key_string* command generates a one-way encrypted key value based on the entered *key_string*. The generated encrypted key value is compared against the encrypted key value of the previously entered chassis key value. If the encrypted values match, the command succeeds and keycheck passes. If the comparison fails, a message is displayed indicating that the key check has failed. If the default chassis key (MAC address) is currently being used, this key check will always fail since there will be no chassis key value to compare against.

Use the **chassis keycheck** command to verify whether multiple chassis share the same chassis key value.

Important For release 19.2 and higher, in the absence of an existing chassis ID file the **chassis keycheck** command is hidden.

For additional information, refer to the *Exec Mode Commands* chapter in the *Command Line Interface Reference*.

Beginning with Release 15.0, the chassis ID will be generated from the chassis key using a more secure algorithm. The resulting 44-character chassis ID will be stored in the same file.

Release 14 and Release 15 chassis IDs will be in different formats. Release 15 will recognize a Release 14 chassis ID and consider it as valid. Upgrading from 14.x to 15.0 will not require changing the chassis ID or configuration file.

However, if the chassis key is reset in Release 15 through the Quick Setup Wizard or CLI command, a new chassis ID will be generated in Release 15 format (44 instead of 16 characters). Release14 builds will not recognize the 44-character chassis ID. If the chassis is subsequently downgraded to Release 14, a new 16-character chassis ID will be generated. To accommodate the old key format, you must save the configuration file in pre-v12.2 format before the downgrade. If you attempt to load a v15 configuration file on the downgraded chassis, StarOS will not be able to decrypt the password/secrets stored in the configuration file.

For release 19.2 and higher, in a chassis where the chassis ID file already exists nothing is changed. However, if the chassis ID file is lost in both management cards, all existing configuration files become invalid. Entering a new chassis key that is the same as the original value will not resolve the issue because of the new method used to generate the chassis ID.

Caution After setting a new chassis key, you must save the configuration before initiating a reload. See the *Verifying and Saving Your Configuration* chapter.

Quick Setup Wizard

The Quick Setup Wizard prompts the user to enter a chassis key value. If a chassis key value is not entered a default chassis is generated using the chassis' MAC address (releases prior to 20.0).

For releases 20.0 and higher, if the chassis ID file does not exist, the Quick Setup Wizard prompts the user to enter a chassis key. A default chassis ID is not generated if a chassis key is not entered.

To run the Quick Setup Wizard, execute the Exec mode **setup** command.

```
[local]host_name# setup
1. Do you wish to continue with the Quick Setup Wizard[yes/no]: y
2. Enable basic configuration[yes/no]: y
3. Change chassis key value[yes/no]: y
4. New chassis key value: key_string
```

Configuring MIO/UMIO/MIO2 Port Redundancy

Port redundancy for MIO cards provides an added level of redundancy that minimizes the impact of network failures that occur external to the system. Examples include switch or router port failures, disconnected or cut cables, or other external faults that cause a link down error.

Caution To ensure that system card and port-level redundancy mechanisms function properly, disable the Spanning Tree protocol on devices connected directly to any system port. Failure to turn off the Spanning Tree protocol may result in failures in the redundancy mechanisms or service outage.

By default, the system provides port-level redundancy when a failure occurs, or you issue the **port switch to** command. In this mode, the ports on active and standby MIO/UMIO/MIO2 cards have the same MAC address, but since only one of these ports may be active at any one time there are no conflicts. This eliminates the need to transfer MAC addresses and send gratuitous ARPs in port failover situations. Instead, for Ethernet ports, three Ethernet broadcast packets containing the source MAC address are sent so that the external network equipment (switch, bridge, or other device) can re-learn the information after the topology change. However, if card removal is detected, the system sends out gratuitous ARPs to the network because of the MAC address change that occurred on the specific port.

With port redundancy, if a failover occurs, only the specific port(s) become active. For example; if port 5/1 fails, then port 6/1 becomes active, while all other active ports on the line card in slot 5 remain in the same active state. In port failover situations, use the **show port table** command to check that ports are active on both cards and that both cards are active.

Take care when administratively disabling a port that is one of a redundant pair. A redundant pair comprises both the active and standby ports—for example 5/1 and 6/1. If 5/1 is active, administratively disabling 5/1 through the CLI does not make 6/1 active. It disables both 5/1 and 6/1 because an action on one port has the same effect on both. Refer to *Creating and Configuring Ethernet Interfaces and Ports* in *System Interface and Port Configuration Procedures*.

With automatic card-level redundancy, there is no port-level redundancy in an MIO/UMIO failover. The standby MIO/UMIO/MIO2 becomes active and all ports on that card become active. The system automatically copies all the MAC addresses and configuration parameters used by the failed MIO/UMIO/MIO2 to its redundant counterpart. The ports on MIOs keep their original MAC addresses, and the system automatically copies the failed MIO/UMIO/MIO2's configuration parameters to its redundant counterpart.

Port redundancy can be configured to be revertive or non-revertive. With revertive redundancy service is returned to the original port when service is restored.

This feature requires specific network topologies to work properly. The network must have redundant switching components or other devices that the system is connected to. The following diagrams show examples of a redundant switching topologies and how the system reacts to various external network device scenarios.

Figure 2: Network Topology Example Using MIO/UMIO Port Redundancy

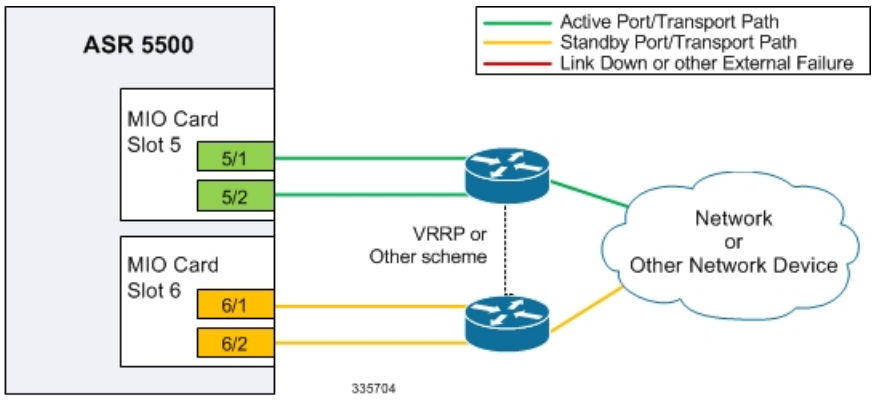
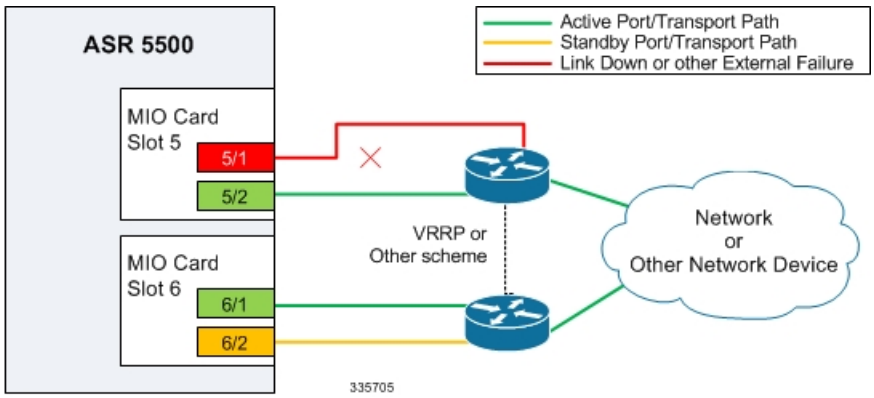
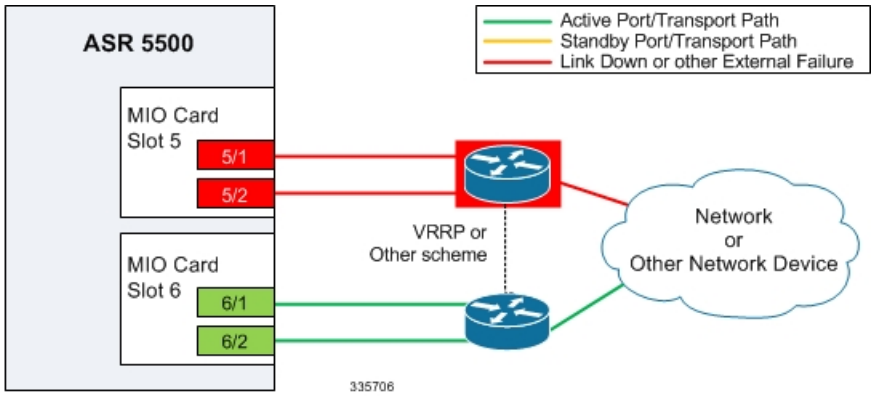


Figure 3: Port Redundancy Failover in Cable Defect Scenario



In the example above, an Ethernet cable is cut or unplugged, causing the link to go down. When this event occurs, the system, with port-mode redundancy enabled, recognizes the link down state and makes port 6/1 the active port. The switching device, using some port redundancy scheme, recognizes the failure and enables the port on the secondary switch to which the MIO/UMIO/MIO2 in slot 6 is connected, allowing it to redirect and transport data.

Figure 4: Port Redundancy Failover in External Network Device Failure Scenario



In the example above, a switch failure causes a link down state on all ports connected to that switch. This failure causes all redundant ports on the line card in slot 6 to move into the active state and utilize the redundant switch.

Configuring MIO/UMIO/MIO2 Port Redundancy Auto-Recovery

You can configure a port auto-recovery feature. When a port failure occurs and the preferred port is returned to service (link is up), control is automatically returned to that port. By default, ports are in a non-revertive state, meaning that no ports are preferred; a manual port switch is required to return use to the original port.

Important This feature is applied on a per port basis (via the **preferred slot** keyword), allowing you to configure specific ports to be used on individual MIO cards. For example, you could configure ports 10 through 19 as preferred on the MIO/UMIO in slot 5, and configure ports 20 through 29 as the preferred ports on the MIO/UMIO in slot 6.

Use the following example to configure a preferred port for revertive, automatic return to service when a problem has cleared:

```
configure
  port ethernet slot#/port#
    preferred slot slot#
    end
```

Notes

- If you do specify a preference, redundancy is revertive to the specified card. If you do not specify a preference, redundancy is non-revertive.
- Repeat for each additional port that you want to make preferred.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Verifying Port Redundancy Auto-Recovery

Verify port information by entering the following command

show port info *slot#/port#*

slot# is the chassis slot number of the MIO/UMIO/MIO2 card on which the physical port resides.

port# is the physical port on the MIO/UMIO/MIO2.

The following shows a sample output of this command for port 1 on the MIO/UMIO/MIO2 in slot 5:

```
[local]host_name# show port info 5/1
Port: 5/1
  Port Type               : 1000 Ethernet
  Role                    : Management Port
  Description             : (None Set)
  Redundancy Mode         : Port Mode
  Redundant With          : 6/1
  Preferred Port          : Non-Revertive
  Physical ifIndex        : 83951616
  Administrative State    : Enabled
  Configured Duplex       : Auto
  Configured Speed        : Auto
```

```
Configured Flow Control : Enabled
Interface MAC Address   :  02-05-47-B8-2F-41
Fixed MAC Address       :  02-05-47-B8-2F-41
Link State              : Up
Link Duplex             : Full
Link Speed              : 1000 Mb
Flow Control            : Disabled
Link Aggregation Group  : None
Logical ifIndex         : 83951617
Operational State       : Up, Active
```

Configuring Data Processing Card Availability

As discussed in the *Understanding the System Boot Process* section of *Understanding System Operation and Configuration*, when the system initially boots up, all installed DPC/UDPCsor DPC2/UDPC2s are placed into standby mode. You must activate some of these cards in order to configure and use them for session processing. One DPC/UDPC or DPC2/UDPC2 may remain in standby mode for redundancy.

This section describes how to activate DPC/UDPCs or DPC2/UDPC2s and specify their redundancy.

Important Refer to the *ASR 5500 Installation Guide* for information about system hardware configurations and redundancy.

Enter the following command to check the operational status of all DPC types:

```
show card table
```

This command lists the DPC types installed in the system by their slot number, their operational status, and whether or not the card is a single point of failure (SPOF).

Use the following example to configure DPC/UDPC or DPC2/UDPC2 availability:

```
configure
  card slot#
    mode { active | standby }
    end
```

Notes:

- When activating cards, remember to keep at least one DPC/UDPC or DPC2/UDPC2 in standby mode for redundancy.
- Repeat for every other DPC/UDPC or DPC2/UDPC2 in the chassis that you wish to activate.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Verifying Card Configurations

Verify that the configuration was successful. Enter the following command:

```
show card table
```

Any DPC/UDPC or DPC2/UDPC2 that you made active should now have an operational status of *Active*.

Enabling Automatic Reset of FSC Fabric

By default if an excessive number of discarded fabric egress packets occurred in the switch fabric, a manual reset of the Fabric Storage Card(s) is required for fabric recovery.

You can optionally enable automatic resets of FSCs if an excessive number of discarded fabric egress packets is detected.

A Global Configuration mode **fabric fsc-auto-recover** command enables or disables automatic FSC resets upon detection of an excessive number of discarded fabric egress packets.

The following command sequence enables this feature:

```
configure
   fabric fsc-auto-recovery { disable | enable } [ max-attempts [
number_attempts | unlimited ] ]
   end
```

max-attempts [*number_attempts* **| unlimited]** specifies how many times StarOS will attempt to reset each FSC as an integer from 1 to 99 or unlimited (will not stop until FSC is reset). The default setting is 1.

Important To enable this feature, you must first configure the Fabric Egress Drop Threshold via the Global Configuration mode **fabric egress drop-threshold** command.

Configuring ASR 5500 Link Aggregation

A Link Aggregation Group (LAG) works by exchanging control packets via Link Aggregation Control Protocol (LACP) over configured physical ports with peers to reach agreement on an aggregation of links as defined in IEEE 802.3ad. The LAG sends and receives the control packets directly on physical ports.

A LAG can have up to 32 member ports, which is 16 ports from MIO/UMIO/MIO2 cards assuming there are two MIO/UMIO/MIO2 cards.

Link aggregation (also called trunking or bonding) provides higher total bandwidth, auto-negotiation, and recovery by combining parallel network links between devices as a single link. A large file is guaranteed to be sent over one of the links, which removes the need to address out-of-order packets.

LAG and Master Port

Logical port configurations (VLAN and binding) are defined in the master port of the LAG. If the master port is removed because of a card removal/failure, another member port becomes the master port (resulting in VPN binding change and outage), unless there is a redundant master port available.

Important The master port on which VLAN can be created for VPN binding must always be configured on the active/master MIO/UMIO/MIO2. The redundancy between the MIO/UMIO/MIO2 in slot 5 and the MIO/UMIO/MIO2 in slot 6 automatically causes both ports to be the master with the same VLANs configured and active.

LAG and Port Redundancy

ASR 5500 LAG implementation assumes that:

- LAG ports on MIO/UMIO/MIO2-slot 5 and MIO/UMIO/MIO2-slot 6 are connected to two Ethernet switches.
- LAG ports on MIO/UMIO/MIO2-slot 5 and MIO/UMIO/MIO2-slot 6 are both active at the same time.
- Ports on MIO/UMIO/MIO2-slot 5 and MIO/UMIO/MIO2-slot 6 are redundant with each other.

All ports in a LAG can be auto-switched to another MIO/UMIO/MIO2 when certain active port counts or bandwidth thresholds are crossed.

LAG and Multiple Switches

This feature connects subscriber traffic ports on MIOs to ports on Ethernet switches. A port failure/switch forces all ports in a LAG to switch to the other MIO/UMIO/MIO2 when a specified threshold is crossed. This works in a way similar to the auto-switch feature for port redundancy. LACP runs between the ASR 5500 and the Ethernet switch, exchanging relevant pieces on information, such as health status.

The following table summarizes typical LAG functionality on an MIO/UMIO/MIO2 card.

Table 2: MIO/UMIO/MIO2 LAG Functionality

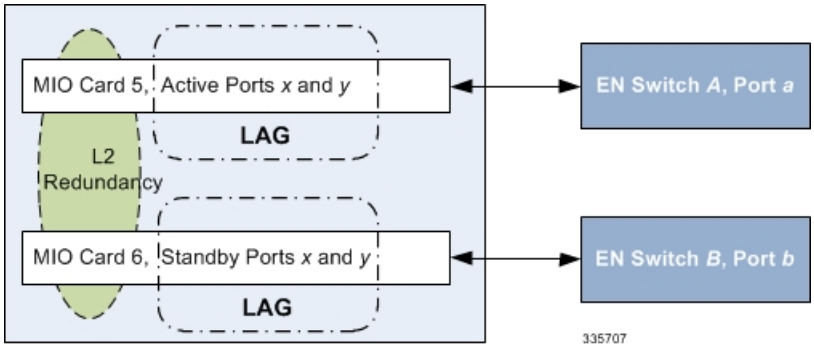
ASR 5500	LAGID	Ethernet Switch A	Ethernet Switch B
MIO/UMIO/MIO2 Port 11	1	Port 1	----
MIO/UMIO/MIO2 Port 12	1	Port 2	----
MIO/UMIO/MIO2 Port 13	1	----	Port 1

Multiple Switches with L2 Redundancy

To handle the implementation of LACP without requiring standby ports to pass LACP packets, two separate instances of LACP are started on redundant cards. The two LACP instances and port link state are monitored to determine whether to initiate an auto-switch (including automatic L2 port switch).

The figure below shows an LAG established across two MIO/UMIO/MIO2 daughter card ports with L2 redundancy.

Figure 5: LAG with L2 Redundancy, Two Ethernet Switches



An LACP implementation with L2 redundancy cannot pass traffic even though standby ports have link up. For example, with two MIO/UMIO/MIO2 cards connected to two different Ethernet switches and all ports in the same LAG, failure of ports would not trigger a LAG switch until the active port number ratio flipped (more ports down than up).

Port States for Auto-Switch

Ports are classified in one of four states to determine whether to start auto-switching. See the table below.

For counters, State(x) represents the number of ports on a card in that state.

Table 3: Auto-Switch Port States

State	Counter	Description
Link	L(x)	Physical link up
Standby	S(x)	Link up but in standby mode
Waiting	W(x)	Waiting for Link Aggregation Control Protocol negotiation
Aggregated	A(x)	Aggregation formed

Hold Time

Once the LAG manager switches to another LACP instance, it does not consider another change for a short period to let link and LACP negotiation settle down. This "hold time" is configurable.

The LAG manager also enters/extends the hold period when an administrator manually switches ports to trigger a card switch.

Preferred Slot

You can define which card is preferred per LAG group as a **preferred slot**. When a preferred MIO/UMIO/MIO2 slot is specified, it is selected for the initial timeout period to make the selection of a switch less random.

Port preference is not allowed in this mode.

Auto-Switch Criteria

The following criteria determine the switching of card x to card y to provide better bandwidth while allowing manual intervention. The evaluation of the criteria occurs outside of the hold period.

Ports are automatically switched from card x to card y when A(y) = 1, at least one port is in aggregated state on card y, and one of the following conditions is true (in order of precedence):

- L(x) L(y) Less ports with link Up on card *x* than card *y*
- S(x) S(y) More ports in Standby state on card *x* than card *y*
- W(x) W(y) More ports in Waiting state on card *x* than card *y*
- A(x) A(y) Fewer ports in Aggregated state on card *x* than card *y*
- Card *y* is preferred
- Card *y* is selected.

Link Aggregation Control

One port in an aggregation group is configured as a master so that all traffic (except control traffic) in the aggregation group logically passes through this port. It is recommended that you configure link-aggregation on the master port first when enabling LAG, and unconfigure the master port last when disabling LAG.

The following command creates link aggregation group *N* with port *slot#/port#* as master. Only one master port is allowed for a group. *N* must be in the range of [1–255].

```
configure
   port ethernet slot#/port#
      link-aggregation master group N
      exit
```

Important Link Aggregation Control Protocol (LACP) starts running only when the master port is enabled.

Use the following command to add a port as member of link aggregation group number *N* only if the master port is assigned. Otherwise, it is added to the group when the master port is assigned:

```
   port ethernet slot#/port#
      link-aggregation member group N
      exit
```

Important The VPN can only bind the master port, and a VLAN can only be created on the master port. A failure message is generated if you attempt to bind to a link aggregation member port.

Each system that participates in link aggregation has a unique system ID that consists of a two-byte priority (where the lowest number [0] has the highest priority) and a six-byte MAC address derived from the first port's MAC address. The following command sets the system priority used to form the system ID. *P* is a hex in the range [0x0000..0xFFFF]. The default is 0x8000.

```
card slot#
   link-aggregation system-priority P
```

Ports in a system are assigned keys. The group number maps directly to the key, whereupon only ports with the same key can be aggregated. Ports on each side of the link use a different aggregation key.

The system ID, port key and port ID of two peers form the Link Aggregation Group Identifier (LAGID). You can aggregate links having the same LAGID. Systems are often configured initially with each port in its own aggregation (requiring a separate key per port), or with all ports in the same aggregation (a single key for all ports). Negotiation via LACP would qualify the actual aggregation.

Systems exchange information about system ID, port key and port ID with peers across the physical links using LACP.

LACP packets are defined with the Slow Protocol format. Each system sends out its own ("actor") information and its last received information about its peer ("partner") over the physical link.

Use the following commands to set the LACP parameters. LACP can run in active mode to send LACP packets periodically, or in passive mode, in which it only responds to LACP packets it receives.

LACP can send packets at either a auto (30s) or fast (1s) rate. The defaults for this release are **Active** and **Auto**; see the sample configuration below:

```
config
   port ethernet slot#/port#
      link-aggregation lacp { active | passive } [ rate { auto | fast }
| timeout { long | short } ]
```

Peers send out LACP packets when the state changes or if a difference is found from a received LACP packet about its own state.

Corresponding ports on an MIO/UMIO/MIO2 redundant pair cannot be active at the same time. Redundant ports share the same MAC address, so after a failover is resolved, the original port rejoins the link aggregation group.

Minimum Links

A minimum links option specifies that a Link Aggregation Group (LAG) is up (usable) only when a minimum number of links are available for aggregation. This guarantees that a minimum amount of bandwidth is available for use.

When this feature is enabled, a LAG is not usable when the number of links in a LAG goes below the configured min-link value. Switchover to another LAG bundle (if available) automatically occurs when the number of links in the current active bundle goes below the configured min-link value.

Use the **min-link** keyword option in the Global Configuration mode **link-aggregation** command to enable this feature.

```
configure
   port ethernet slot/port
      link-aggreagation master ( global | group } number
         min-link number_links
         end
```

Redundancy Options

For L2 redundancy set the following option on the master port for use with the whole group:

link-aggregation redundancy standard [hold-time *sec* **] [preferred slot {** *card_number* **| none }**

Standard redundancy treats all cards in the group as one group.

Horizontal Link Aggregation with Two Ethernet Switches

When a LAG contains two sets of ports each connecting to a different switch, the operator has the ability to specify the slot/port (connected to the destination switch) when switching ports.

The Exec mode **link-aggregation port switch to** *slot/port* command configures this option. The *slot/port* is any valid port connected to the destination switch. The following criteria apply to the setting of this option:

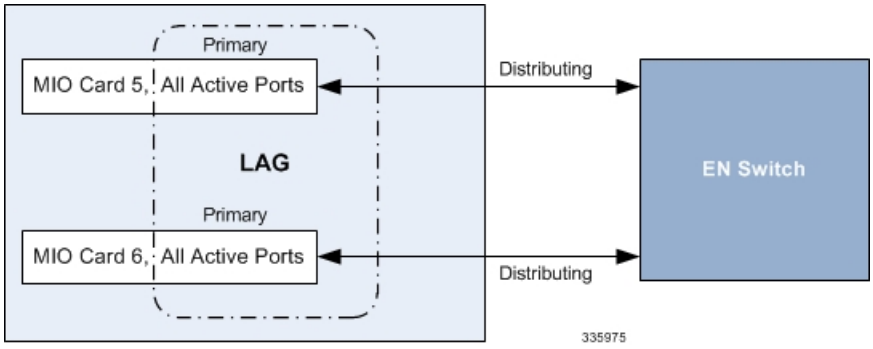
- *slot/port* must support LAG.
- *slot/port* must be configured with LAG.
- *slot/port* must not be already actively distributing
- *slot/port* must have negotiated a link aggregation partner in standard mode.
- *slot/port*'s partner must have an equal or higher in standard mode.
- *slot/port*'s partner bundle must have equal or higher bandwidth in standard mode.
- Switching to *slot/port* must not violate preference within hold-time in standard mode.

Non-Redundant (Active-Active) LAG

LAG can be deployed in a non-redundant mode in which the ports from both MIO/UMIO/MIO2 cards are connected to the same switch.

As shown in the following figure, all ports in a LAG used on both the cards function in a non-redundant mode (Active/Active).

Figure 6: Non-Redundant LAG Configuration with Single LAG Group



In the above configuration, there is a single, primary LAG. All ports work as a single bundle of ports that distribute the traffic.

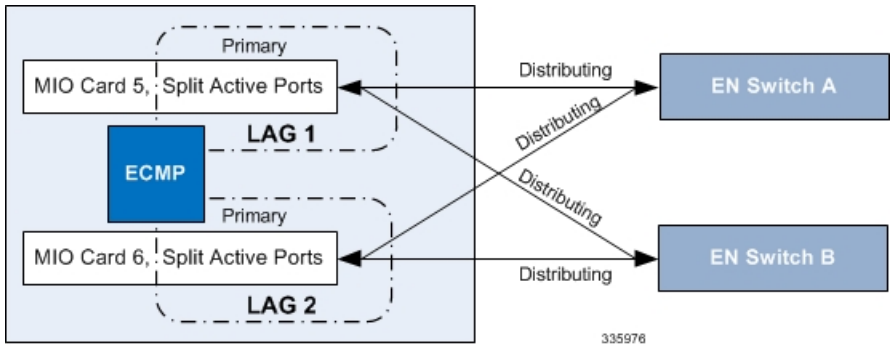
Important If you use the Ethernet Port Configuration mode **shutdown** command to shut down one of the ports on an MIO/UMIO/MIO2 card in this LAG configuration, by default the paired port on the other MIO/UMIO/MIO2 card will also be shut down. You can selectively disable an MIO/UMIO/MIO2 port in this LAG configuration using the Exec mode **port { disable | enable } ethernet** *slot/port* command.

Important With this mode of operation, automatic ASR 5500 port redundancy is lost.

To achieve redundancy you must configure a second non-redundant LAG. You can use a higher layer load balancing mechanism such as ECMP (Equal Cost Multiple Path) routing to uniformly distribute the traffic across two LAG groups.

When one MIO/UMIO/MIO2 fails, half the ports from both the LAG groups will be available for distribution of the traffic from the other MIO/UMIO/MIO2.

Figure 7: Non-Redundant LAG Configuration with ECMP



Configuring a second LAG group is not mandatory, but is the usual approach for achieving redundancy with this mode of LAG.

However, if the aggregating ports are loaded with more than 50% of their capacity and an MIO/UMIO/MIO2 failure/switchover occurs, the ASR 5500 configured port capacity is oversubscribed and an indeterminate amount of sessions are dropped and traffic lost.

Faster Data Plane Convergence

The Global Configuration mode **fast-data-plane-convergence** command enables faster recovery of existing sessions in an Active-Active LAG configuration with aggressive MicroBFD timers. This feature can be enabled with an Active-Standby LAG configuration, however, reduced switchover time cannot be guaranteed.

This feature eliminates false positive detection of failure with an external switch and false positive ICSR failover with another ASR 5500.

```
configure
  fast-data-plane-convergence
```

Important Active-Active LAG groups must be configured, along with aggressive microBFD timers (such as 150*3). During MIO card recovery BGP Sessions might flap based on the configuration. To avoid traffic loss during these events, BGP graceful restart must be configured with proper hold/keepalive and restart timers. See the description of the **bgp graceful-restart** command in the *BGP Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Link Aggregation Status

To check the status of link aggregation, use the following commands:

- **show port table**
- **show port info** *slot/port*

A single character is used to display LAG physical port status in the output of the **show port table** command. See the table below.

Table 4: LAG Port Status

Display	Description
LA+	Port is actively used for distributing (transmit nd recieve data).
LA-	Port failed to negotiate LACP.
LA~ (tilde)	Port negotiated LACP but another peer was selected.
LA*	Port is (re)negotiating LACP.
LA#	Port has been gone down because the min-link criteria is not met.

Configuring a Demux Card

You can dedicate a DPC/UDPC or DPC2/UDPC2, or MIO/UMIO/MIO2 to function as a demux card. Demux is a generic term for signal demultiplexing tasks. These are the tasks are responsible for parsing call setup (signaling packets) and distributing the calls internally. For this reason there almost as many tasks running on a demux card as there are services.

The vpnmgr tasks responsible for each context also run on the demux card. The number of vpnmgr tasks correspond to the number of contexts. A vpnmgr is responsible for IP address assignment to mobile equipment, IP routing (such as BGP, OSPF), as well as a variety of associated tasks.

Overview

Designating a DPC/UDPC or DPC2/UDPC2, or MIO/UMIO/MIO2 as a demux card frees up resources for session handling, which has the potential to increase system throughput. However, there is no increased support in total subscriber capacity due to other system resource restrictions.

This feature is disabled by default and can be enabled via the Global Configuration mode **require demux** command. It is only supported for a limited number of products. Refer to the product Administration Guide for additional information.

To support this feature session recovery must also be enabled via the Global Configuration mode **require session recovery** command.

Important After enabling demux card and session recovery, you must save the configuration and reboot the ASR 5500 to enable this feature.

Caution Enabling the Demux on MIO/UMIO/MIO2 feature changes resource allocations within the system. This directly impacts an upgrade or downgrade between StarOS versions in ICSR configurations. Contact Cisco TAC for procedural assistance prior to upgrading or downgrading your ICSR deployment.

MIO Demux Restrictions

The following restrictions apply when enabling an MIO/UMIO/MIO2 as a demux card:

- The **require demux management-card** command must be configured before any service or contexts have been created on the system. The command will not execute after a mode of operation has been selected for the chassis.
- Only the following services currently support the designation of an MIO/UMIO/MIO2 card for demux functions: ePDG (StarOS Release 21.2 and later), GGSN, HeNBGW (StarOS Release 21.2 and later, SaMOG (StarOS Release 21.2 and later), SGW, PGW, HA, SAE-GW and L2TP LNS. These services are supported only when they are deployed as consumer gateways.
- SGSN, MME, HNBGW, HeNBGW (StarOS Release 21.1 and earlier), SaMOG (StarOS Release 21.1 and earlier), PDG, PDIF, ePDG (StarOS Release 21.1 and earlier), IPSG, PDSN, HSGW, L2TP LAC, NEMO, FA, and WSG are not supported. Enterprise or corporate gateways (GGSN, HA, PGW, etc.) are also not supported.
- You should not enable demux functionality on MIO/UMIO/MIO2 for configurations that require a large number of tunnels.
- After the ASR 5500 has booted with demux functions running on an MIO/UMIO/MIO2, you cannot configure non-supported services. A maximum of eight Demux Managers are supported. Any attempt to add more than eight Demux Managers will be blocked.
- Service/products requiring a large number of VPN Managers, VRFs and/or Demux Managers must not enable demux functions on an MIO/UMIO/MIO2.
- With demux functions running on an MIO/UMIO/MIO2, the ASR 5500 supports a maximum of 10 contexts, 64 interfaces per context, and 250 VRFs per system.
- ICSR upgrades require compatible configurations and Methods of Procedure (MOPs).

Implementation of this feature assumes that CEPS (Call Events Per Second) and the number of subscribers will remain constant, and only the data rate will increase. This ensures that the CPU demand will not increase on the MIO/UMIO/MIO2.

Note If a process crash occurs in the background on a demux card, planned or unplanned migration of the card fails.

Important Contact Cisco TAC for additional assistance when assessing the impact to system configurations when enabling the Demux on MIO/UMIO/MIO2 feature.

Configuration

For releases prior to 15.0, to configure a DPC/UDPC as a demux card enter the following CLI commands:

```
configue
  require demux card
  end
```

For release 15.0+, to configure a DPC/UDPC as a demux card enter the following CLI commands:

```
configue
  require demux processing-card
  end
```

For release 18.0+, to configure a DPC/2UDPC2 as a demux card enter the following CLI commands:

```
configue
  require demux processing-card
  end
```

To configure an MIO/UMIO/MIO2 as a demux card enter the following CLI commands:

```
configue
  require demux management-card
  end
```