Configuring VIP and Virtual IP Interface Redundancy

This chapter describes how to plan for and configure Virtual IP (VIP) and Virtual IP Interface Redundancy on the CSS. Information in this chapter applies to all CSS models except where noted.

This chapter contains the following sections:

- VIP and Virtual IP Interface Redundancy Overview
- VIP and Virtual IP Interface Redundancy Quick Start
- Configuring VIP and Virtual IP Interface Redundancy
- Synchronizing a VIP Redundancy Configuration
- Displaying Redundant VIP and Virtual IP Interface Configurations
- VIP and Virtual IP Interface Redundancy Running-Config Examples
- Configuring Adaptive Session Redundancy
- Displaying Adaptive Session Redundancy Information
VIP and Virtual IP Interface Redundancy Overview

The CSS enables you to configure redundancy for:

- Virtual IP Address (VIP) Redundancy
- Virtual IP Interface Redundancy

Virtual IP Address (VIP) Redundancy

When you configure more than one CSS for processing or forwarding client requests to the same Virtual IP address (VIP), the VIP is considered redundant. A typical use of VIP redundancy would be in a configuration where a master CSS processes all client requests to a VIP using a directly-connected dedicated Web server farm. If the Web server farm becomes unavailable, the backup CSS takes over using its own dedicated Web servers.

Note

The CSS does not support VIP redundancy and CSS-to-CSS redundancy (IP redundancy) simultaneously. For information on IP redundancy, refer to Chapter 7, Configuring Redundant Content Services Switches.

To setup CSSs for VIP redundancy, you must configure a virtual router on each CSS that will participate in the redundant configuration. A virtual router is an entity within a CSS to which you associate an existing VIP. A VIP becomes redundant when you associate it with a virtual router. You may configure a maximum of 255 virtual routers for each VLAN. You can associate a virtual router only with a single VLAN.

Virtual routers providing redundancy for an IP address are considered peers. Each virtual router peer has the same identifier and runs on the same VLAN, but runs on a different CSS. Once the virtual routers are configured, the CSSs negotiate for mastership using Virtual Router Redundancy Protocol (VRRP). A virtual router in a redundant VIP configuration that is designated as:

- Master will process all client requests directed to the VIP
- Backup may be either a:
  - Backup virtual router, which forwards all client requests directed to the VIP to the master CSS
  - Shared backup virtual router, which processes all client requests it receives
A CSS can serve simultaneously as a master to one virtual router and as a backup to a different virtual router. All redundant VIP addresses will share the state of the virtual router to which it is associated.

**Figure 6-1** shows an example of a redundant VIP configuration with:

- **CSS-Boston configured as:**
  - VLAN1 IP address 192.168.8.1.
  - Master virtual router 1 for VIP 192.168.8.4.
  - Backup virtual router 2 for VIP 192.168.8.6. Because virtual router 2 is operating as a backup, it will forward all client requests it receives for VIP 192.168.8.6 to CSS-Cambridge master virtual router 2.

- **CSS-Cambridge configured as:**
  - VLAN1 IP address 192.168.8.2.
  - Backup virtual router 1 for VIP 192.168.8.4. Because virtual router 1 is operating as a backup, it will forward all client requests it receives for VIP 192.168.8.4 to CSS-Boston master virtual router 1.
  - Master virtual router 2 for VIP 192.168.8.6.

**Figure 6-1 Master and Backup Virtual Router Redundant VIP Configuration Example**

![Master and Backup Virtual Router Redundant VIP Configuration Example](image-url)
Figure 6-2 shows an example of a redundant VIP configuration with:

- CSS-Boston configured as:
  - Master virtual router 1 for VIP 192.168.8.4.
  - Shared backup virtual router 2 for VIP 192.168.8.6. Because virtual router 2 is operating as a shared backup, it will process all client requests it receives for VIP 192.168.8.6.

- CSS-Cambridge configured as:
  - Shared backup virtual router 1 for VIP 192.168.8.4. Because virtual router 1 is operating as a shared backup, it will process all client requests it receives for VIP 192.168.8.4.
  - Master virtual router 2 for VIP 192.168.8.6.

Figure 6-2  Master CSS and Shared Backup CSS Redundant VIP Configuration Example
Virtual IP Interface Redundancy

Virtual interface redundancy is a form of IP address redundancy that applies only to IP interfaces (not VIPs). A typical interface IP address on a CSS defines the interface in use on a particular VLAN. In this type of configuration, the CSS designated as master maintains control over the interface IP address.

The typical use for virtual interface redundancy is in a configuration where servers are positioned behind a Layer 2 switch and CSSs with the redundant virtual interface are positioned in front of the Layer 2 switch. The servers would be configured with a default route pointing to the redundant virtual interface IP address.

A CSS designated as master of a virtual interface sends out gratuitous ARPs for the virtual interface’s IP address. This enables the Layer 2 switch to learn where to forward packets that are directed to the virtual interface. This allows a server’s default route to always point to the CSS designated as the master of the virtual interface.

Figure 6-3 shows an example of a virtual interface redundancy configuration with a master CSS and a backup CSS.

- **Note**: Interface redundancy does not support *shared* backup.

---

**Figure 6-3 Virtual Interface Redundancy Configuration Example using a Master and a Backup CSS**
VIP and Virtual IP Interface Redundancy Quick Start

Table 6-1 provides a quick overview of the steps required to configure VIP and virtual interface redundancy for each CSS in the redundant configuration. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI command, refer to the sections following Table 6-1.

<table>
<thead>
<tr>
<th>Task and Command Example</th>
<th></th>
</tr>
</thead>
</table>
| 1. Enter into config mode. | # config  
(config)# |
| 2. Enter circuit mode for the desired circuit VLAN. | (config)# circuit VLAN1  
(config-circuit[VLAN1])# |
| 3. Configure a circuit IP address. | (config-circuit[VLAN1])# ip address 192.168.8.1/24  
(config-circuit-ip[VLAN1-192.168.8.1])# |
| 4. Configure the virtual router. Optionally, you may assign a priority different from the default of 100. Include the preempt keyword when configuring the master virtual router. The master virtual router must have the highest priority among its peers. | (config-circuit-ip[VLAN1-192.168.8.1])# ip virtual-router 1 priority 230 preempt |
| 5. Configure the redundant virtual IP interface on the virtual router. | (config-circuit-ip[VLAN1-192.168.8.1])# ip redundant-interface 1 192.168.8.6 |
Table 6-1  VIP Redundancy and Virtual IP Interface Redundancy Configuration Quick Start (continued)

<table>
<thead>
<tr>
<th>Task and Command Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>6. Configure the redundancy VIP on the virtual router. If you defined the content rule VIP using the range option, you must configure an identical range for the redundant VIP.</td>
</tr>
<tr>
<td>(config-circuit-ip[_vlan1-192.168.8.1])# ip redundant-vip 1 192.168.8.10 range 10</td>
</tr>
<tr>
<td>If you want the backup virtual router to process client requests, you must configure it as a shared backup.</td>
</tr>
<tr>
<td>(config-circuit-ip[_vlan1-192.168.8.1])# ip redundant-vip 1 192.168.8.10 shared</td>
</tr>
<tr>
<td>7. Configure the critical service for the virtual router.</td>
</tr>
<tr>
<td>(config-circuit-ip[_vlan1-192.168.8.1])# ip critical-service 1 serv1</td>
</tr>
<tr>
<td>8. Display the configuration (optional).</td>
</tr>
<tr>
<td>(config)# show virtual-routers</td>
</tr>
</tbody>
</table>

Because this chapter is dedicated to configuring VIP and virtual interface redundancy, it contains only those circuit IP commands that pertain to this feature. For a complete description of all circuit IP commands, refer to the Cisco Content Services Switch Administration Guide, Chapter 2, Configuring Interfaces and Circuits.
Configuring VIP and Virtual IP Interface Redundancy

The following sections describe how to configure VIP and virtual IP interface redundancy. You must configure each CSS in a redundant configuration.

- Configuring a Circuit IP Interface
- Configuring an IP Virtual Router
- Configuring an IP Redundant Interface
- Configuring an IP Redundant VIP
- Configuring IP Critical Services
- Synchronizing a VIP Redundancy Configuration

Configuring a Circuit IP Interface

Before you can configure VIP and virtual interface redundancy, you must configure a circuit IP interface and assign it an IP address. To enter a specific circuit configuration mode, enter the `circuit` command and VLAN as shown in the following example:

```
(config)# circuit VLAN1
(config-circuit[VLAN1])#
```

**Note**

When you use the `circuit` command, enter the word “VLAN” in uppercase letters and do not include a space between VLAN and the VLAN number (for example, VLAN1).

To assign an IP address to a circuit, use the `ip address` command from the specific circuit mode. Enter the IP address and a subnet mask in CIDR bitcount notation or a mask in dot-decimal notation. The subnet mask range is 8 to 32. For example, to configure an IP address and subnet mask for VLAN1, enter:

```
(config-circuit[VLAN1])# ip address 192.168.8.1/24
```

When you specify an IP address, the mode changes to the specific circuit-ip-VLAN-IP address as shown:

```
(config-circuit-ip[VLAN1-192.168.8.1])#
```
Configuring an IP Virtual Router

Use the `ip virtual-router` command to create a virtual router on a CSS and configure its identifier and priority used when negotiating control of associated VIPs. You must configure the virtual router before you can configure redundant VIPs.

A virtual router’s role as a master or backup is determined during negotiations between all virtual routers with the same ID and on the same VLAN.

The syntax and options for the IP interface command are:

```
   ip virtual-router vrid {priority number} {preempt}
```

The variables and options are:

- `vrid` - The virtual router identifier (VRID). Enter an integer between 1 and 255. You can configure 255 virtual routers per VLAN. Virtual routers are considered peers when they have the same VRID and are on the same VLAN.

- `priority number` - The optional priority of the virtual router with its peer. The default priority value is 100. Enter an integer between 1 and 255. A virtual router with the highest priority usually becomes master. However, a higher priority virtual router will not assume mastership from a lower priority master unless you include the `preempt` option. When a virtual router is the master, it handles the traffic directed to its associated VIPs. To set a virtual router so that it will always be master, set its priority to 255 and configure it with the `preempt` option.

- `preempt` - The optional keyword that allows a higher priority virtual router to assert mastership over a lower priority virtual router. By default, a virtual router does not become master if the current master has a lower priority. For example, if a CSS with a virtual router that has a low priority boots before other CSSs, that virtual router becomes the master. When another CSS with a virtual router that has a higher priority boots, it will not take the mastership from the first router unless you specify the `preempt` option on the higher priority virtual router. This option does not have an effect if the priority of the two virtual routers is identical. You can use this option with or without the `priority` option. You can configure only one virtual router as the master of a particular VIP.

⚠️ **Caution**

Never configure the `preempt` option on the same virtual router on both CSSs. Such a configuration may result in both CSSs becoming master, which will cause network problems.
Because a virtual router’s priority is dependent on the state of the critical services, the priority field status in the `show virtual router` display may be different than the priority you configured. The priority may be different when you:

- Assign a priority of 255 to a virtual router and that virtual router gains mastership, the CSS automatically reconfigures that virtual router’s priority to 254. This action ensures that you can assign a different virtual router a priority of 255.

- Configure critical services. The critical service types are:
  - **scripted** - the priority changes to 0 when one service in the scripted group goes down
  - **redundancy uplink** - the priority changes to 0 when all of the services in the uplink group go down
  - **local** - the priority changes to 0 when all of the services in the local group go down. Local services include all services other than scripted and uplink.

For information on configuring critical services, refer to “Configuring IP Critical Services” later in this chapter.

For example:

```
(config-circuit-ip[VLAN1-192.168.8.1])# ip virtual-router 1
 priority 1 preempt
```

To remove the virtual router from the CSS, enter:

```
(config-circuit-ip[VLAN1-192.168.8.1])# no ip virtual-router 1
```

### Configuring an IP Redundant Interface

Use the **ip redundant-interface** command to configure a redundant virtual interface address used for a backend server’s default route. Servers use the IP address of the virtual interface as a default route to guarantee packets will be sent to the CSS containing the master virtual router. You may assign a redundant interface with the same virtual router of a VIP that has a rule that refers to the server. This ensures that the master for a VIP is also the CSS that is master for the redundant virtual interface.

The syntax for this IP mode command is:

```
ip redundant-interface vrid ip_address
```
The variables are:

- `vrid` - The ID for a previously configured virtual router.
- `ip_address` - The address for the redundant interface. Enter an IP address in dotted-decimal notation (for example, 192.168.8.6).

**Note** You cannot use an IP address that already exists for a VIP, redundant VIP, source group, service, log host, or IP interface address on a circuit. If you do, the following error message appears: Address conflicts with local I/F, VIP, service, or sourcegroup.

For example:
```
(config-circuit-ip[VLAN1-192.168.8.1])# ip redundant-interface 1 192.168.8.6
```

To remove an interface from a virtual router, enter:
```
(config-circuit-ip[VLAN1-192.168.8.1])# no ip redundant-interface 1 192.168.8.6
```

**Note** The CSS does not support a traceroute of a redundant IP interface.

### Configuring an IP Redundant VIP

Use the `ip redundant-vip` command to associate an existing VIP to a virtual router and if required, configure the virtual router as a shared backup. A shared backup virtual router processes client requests. A redundant VIP configuration can consist of only two CSSs.

**Note** Before you use this command, the VIP must be configured in a minimum of one content rule. Additionally, if you defined the content rule VIP using the range option, you must configure an identical range for the redundant VIP.

The syntax for this IP mode command is:

```
ip redundant-vip vrid vip_address {range number} {shared}
```
The variables and options are:

- **vrid** - The ID for an existing virtual router.
- **vip_address** - The address for the redundant VIP. This address must be already configured in a content rule. Enter an IP address in dotted-decimal notation (for example, 192.168.8.10).
- **range number** - The optional keyword and variable if an IP address range is specified in the content rule. You cannot specify a range that differs from the content rule. Also, you cannot specify address ranges to overlap. Enter a number from 0 to 65535. The default is 1.
- **shared** - The optional keyword to enable shared VIP redundancy. When you use this option, the master and backup virtual routers share the processing of traffic directed to the VIP, so the backup does not forward packets to the master. Each VIP should be configured identically on both CSSs.

For example:

```
(config-circuit-ip[VLAN1-192.168.8.1])# ip redundant-vip 1 192.168.8.10 range 10 shared
```

To remove a VIP from a virtual router, enter:

```
(config-circuit-ip[VLAN1-192.168.8.1])# no ip redundant-vip 1 192.168.8.10
```

### Configuring IP Critical Services

Use the **ip critical-service** command to associate a critical service with a virtual router. When a critical service goes down, the associated virtual router will also go down. There are three types of critical services that you can configure:

- A scripted critical service, as defined by the (config-service) keepalive type **script** command or the (config-service) keepalive type **named** command, that is constantly scanning for service and network availability. The keepalive sets the service to a down state whenever network or service availability is a problem. The virtual router goes down if any associated scripted service goes down.

- A redundancy uplink critical service, as defined by the (config-service) type **redundancy-up** command. The virtual router goes down when all associated redundancy uplink services go down regardless of any configured keepalive type. Refer to Chapter 7, Configuring Redundant Content Services Switches, in the section “Configuring Multiple Redundant Uplink Services”.
You cannot add redundant uplink services to a content rule.

- Local critical services for any service other than scripted or redundancy uplink, such as a Web service. The virtual router goes down when all associated local critical services go down.

The syntax and options for the `ip critical-service` command are:

```
ip critical-service vrid service_name
```

The variables are:
- `vrid` - The ID for an existing virtual router.
- `service_name` - The name of the service. To see a list of services, enter `ip critical-service vrid ?`.

For example:

```
(config-circuit-ip[VLAN1-192.168.8.1])# ip critical-service 1 serv1
```

To remove a critical service from a virtual router, enter:

```
(config-circuit-ip[VLAN1-192.168.8.1])# no ip critical-service 1 serv1
```

The `show service` command displays the current service type only. It does, however, display the keepalive type, so you can determine from it the behavior of a configured critical service. To display critical service-specific information, use the `show critical-services` command. See “Displaying IP Critical Services” later in this chapter.

SNMP values returned for services show the current service type only. To determine the critical service behavior of a particular service, you need to consult the service keepalive type. For more information on SNMP, refer to the Cisco Content Services Switch Administration Guide.
Synchronizing a VIP Redundancy Configuration

To ensure that your backup CSS can perform the same tasks as your master CSS in the event of a master CSS failure, the running-config on the backup must be identical (with some modifications) to the running-config on the master. To automate this configuration synchronization process, you can run a script (commit_VipRedundConfig) on the master CSS that copies the master CSS running-config to the backup CSS running-config.

There are two types of configuration synchronization:

- **Complete** - On CSSs that have an identical chassis (the same CSS model), produces a running-config on the backup CSS that exactly matches the running-config on the master CSS.

- **Partial** (default) - On CSSs with incompatible configurations, synchronizes all parameter values in the configuration except the interface and circuit configurations. For example, the master is a CSS 11506 and the backup is a CSS 11503. The script maintains the current backup interface and circuit configurations automatically.

### Script Functions

The configuration synchronization script performs all the necessary steps to update the backup CSS with the master's running configuration. The script:

- Saves the master running-config to the startup-config
- Archives the startup-config
- Copies the startup-config to a temporary file (tmp.cfg)
- Calls a function that converts the master VRRP/APP IP addresses to the backup VRRP/APP IP addresses in tmp.cfg
- Changes all VRID priorities on the master to 254
- Changes all VRID priorities in the backup's new config to 1
- Removes all preempt configs from all VRIDs in the backup's new config
• Uses `rcmd` to:
  – Copy tmp.cfg to a temp file on the backup (newconfig)
  – Check newconfig and copy it to the startup-config
  – Clear the backup CSS’s running-config and script play newconfig

The script performs some verifications before executing the above steps. It checks to see if the local switch is a backup for any VRIDs and asks you if you want to continue, thereby changing the state on the two CSSs. The script also checks the backup to see if it is the master for any VRIDs. If the state is Interface (IF) Down, the script asks you if you want to continue without synchronizing those VRIDs on interfaces that are Down.

**Before You Begin**

Before you run the configuration synchronization script, ensure that you have configured VIP/interface redundancy and the Application Peering Protocol (APP). For details on configuring VIP/interface redundancy, see “Configuring VIP and Virtual IP Interface Redundancy” earlier in this chapter. For details on configuring APP, refer to Chapter 1, Configuring the CSS Domain Name Service in the section “Configuring Application Peering Protocol”.

The synchronization script does not support the following configurations:

• Active/active shared VIP
• Any configuration where some independent VIP addresses are a master while other VIP addresses are a backup

**Running the Configuration Synchronization Script**

To run the configuration synchronization script, use the `script play commit_vip_redundancy` command. The syntax is:

```plaintext
script play commit_vip_redundancy "arguments"
```

You can also run the configuration synchronization script using the predefined alias that comes with all CSSs, as follows:

```plaintext
commit_VipRedundConfig "arguments"
```
The arguments for the commit_redundancy script are:

- **ip address** - The IP addresses of the master and backup APP sessions. This is the only required argument for this script. Use the following syntax when entering the addresses:

  "local master IP address remote backup IP address"

For details on automating the entry of the IP address, see "Setting the BACKUP_VIPR_IP Variable" later in this section.

- **-a (All)** - Synchronizes the configuration completely. Use this argument only when the master and backup CSSs have identical chassis. This argument synchronizes the entire configuration and the interface mode.

- **-d (Debug)** - Debug switch for the commit_redundancy script, which displays the current task being performed as the script progresses. Debug messages display even when you specify the -s argument.

⚠️ **Caution**

Before you use the -f argument to remove a config sync lock file, ensure that no one else is running the config sync script on the CSS. Otherwise, if you remove the lock file and then run the script again while the script is in use, the resulting configurations may have some discrepancies.

- **-f** - After an abnormal script termination, removes the lock file so that you can run the script again. This argument overrides all other specified arguments and the script exits immediately after removing the lock file. For details on the lock file, see "Setting the BACKUP_VIPR_IP Variable" later in this section.

- **-nv (No Verify)** - Informs the script not to verify that the configuration synchronization was successful.

⚠️ **Note**

The script verifies the configuration synchronization by default.

- **-s (Silent)** - Suppresses script progress messages and displays only the result of running the script: Commit Successful or Commit Failed. The -d argument overrides the -s argument.
You can specify the script arguments in any order.

For example, on the master CSS, run the following script, which uses the defaults of verify on and partial synchronization, plus the IP addresses set as variables and the script alias name:

```
CSS11503# commit_VipRedundConfig
```

The following output appears:

```
CSS11503# commit_VipRedundConfig
Verifying app and redundancy configs ...  
Checking vip redundancy state ...  
Working \  
Verifying running-config copy success ...  
Commit successful!
```

In this example, the script:

- Performs a partial configuration synchronization (default)
- Verifies that the configuration synchronization was successful (default)

For more information on scripts, refer to Chapter 11, Using the CSS Scripting Language.

**Config Sync Lock File**

When you run the script, the software creates a lock file (vipr_config_sync_lock) in the script directory so that you cannot run the script from another session on the CSS. If the lock file exists and you run the script, the following message appears:

```
The script is in use by another session.
```

If the script terminates abnormally, the software does not remove the lock file. The next time you run the script, the above message appears. If you are certain that the script is not in use by another session, use the -f argument to remove the lock file. When you run the script with this argument, the following message appears and the script exits:

```
VIPR Config Sync lock file removed.
```

Now you can run the script again.
Setting the BACKUP_VIPR_IP Variable

To eliminate the need to specify IP addresses each time you run the configuration synchronization script, you can set the value of two variables (BACKUP_VIPR_IP and MASTER_VIPR_IP) to IP addresses and save them in your user profile. Once you set the variables and save them in your user profile, the variables will always be available after you log in to the CSS.

To set the variables, enter:

```
# set BACKUP_VIPR_IP "ip_address" {session}
# set MASTER_VIPR_IP "ip_address" {session}
```

where, `ip_address` is the IP address of the backup or master CSS.

To save the variable in your user profile, enter:

```
# copy profile user-profile
```

Now you can run the configuration synchronization script without typing an IP address.
Logging Configuration Synchronization Script Result Messages

You can specify that script result messages (script success or failure messages) be sent to the current logging device automatically each time you run the configuration synchronization script. To log the script result messages, enable logging on NETMAN with level info-6 or debug-7 by entering:

```
(config)# logging subsystem netman level info-6
```

**Note**

Log messages are generated with or without the `-s` (silent) argument specified. See “Running the Configuration Synchronization Script” earlier in this chapter.

For example, if the APP session to the backup CSS is not running, the CSS generates the following log message:

```
vipr config sync: app session is DOWN
```

For ease of tracking, each log message contains the string “vipr config sync”.
Displaying Redundant VIP and Virtual IP Interface Configurations

The CSS provides show commands to enable you to display redundant VIP and virtual interface configurations. The following sections describe the commands and provide examples of the screen displays and tables describing the screen fields.

- Displaying IP Critical Services
- Displaying Redundant Interfaces
- Displaying Redundant VIPs
- Displaying Virtual Router Configurations

Displaying IP Critical Services

Use the `show critical-services` command to display a list of all critical services configured on the CSS. You may provide an interface IP address option to display only the critical services present on a particular interface. You may also include a VRID to display only the critical service information for a particular virtual router.

The syntax for this command is:

```
show critical-services {ip_address {vrid}}
```

The optional variables are:

- `ip_address` - The address for the redundant interface. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1).
- `vrid` - The ID for an existing virtual router.

For example, to view all critical services on the CSS, enter:

```
# show critical-services
```
Table 6-2 describes the fields.

**Table 6-2  Field Descriptions for the Show Critical Services Command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Address</td>
<td>The IP interface address associated with the virtual router.</td>
</tr>
<tr>
<td>VRID</td>
<td>The assigned identifier associated with the virtual router.</td>
</tr>
<tr>
<td>Service Name</td>
<td>The name of the critical service.</td>
</tr>
<tr>
<td>Service Type</td>
<td>The type of critical service. Possible critical service types are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Scripted</strong> - A service whose state depends upon a running script or a named keepalive.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Redundancy-up</strong> - A service whose state depends upon the state of an ICMP keepalive on a router.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Local</strong> - Every type of service other than a scripted service or a redundancy uplink service. Typically, this is a Web server.</td>
</tr>
</tbody>
</table>

**Displaying Redundant Interfaces**

Use the `show redundant-interfaces` command to display a list of all redundant virtual IP interfaces configured on the CSS. You may provide an interface IP address option to display only the virtual interfaces present on a particular interface. You may also include a VRID to display only the virtual interface information for a particular virtual router.

The syntax for this command is:

```
show redundant-interfaces {ip_address {vrid}}
```
The optional variables are:

- *ip_address* - The address for the redundant interface. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1).
- *vrid* - The ID for an existing virtual router.

For example, to view all redundant interfaces on the CSS, enter:

```
(config) # show redundant-interfaces
```

Table 6-3 describes the fields.

**Table 6-3  Field Descriptions for the Show Redundant Interface Command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Address</td>
<td>The IP interface address associated with the redundant virtual interface.</td>
</tr>
<tr>
<td>VRID</td>
<td>The assigned identifier associated with the virtual router.</td>
</tr>
<tr>
<td>Redundant Address</td>
<td>The IP address of the redundant virtual interface.</td>
</tr>
<tr>
<td>Range</td>
<td>Not applicable. This field is always set to 1.</td>
</tr>
<tr>
<td>State</td>
<td>Current state of the virtual router. Possible states are:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Master</strong> - The virtual router is master.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Backup</strong> - The virtual router is backup.</td>
</tr>
<tr>
<td></td>
<td>- <strong>No Service</strong> - One or more critical services associated with the virtual router is down.</td>
</tr>
<tr>
<td></td>
<td>- <strong>IF Down</strong> - The IP interface associated with the virtual router is down.</td>
</tr>
<tr>
<td>Master IP</td>
<td>The IP address of the master virtual router.</td>
</tr>
<tr>
<td>State Changes</td>
<td>The number of times the redundant virtual interface state has changed.</td>
</tr>
<tr>
<td>Last Change</td>
<td>The date and time of the redundant virtual interface state last state change.</td>
</tr>
</tbody>
</table>
Displaying Redundant VIPs

Use the `show redundant-vips` command to display a list of all redundant VIPs configured on the CSS. You could provide an interface IP address option to display only the VIPs present on a particular interface. You can also include a VRID to display only the VIP information for a particular virtual router.

The syntax for this command is:

```
show redundant-vips {ip_address {vrid}}
```

The optional variables are:

- `ip_address` - The address for the redundant interface. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1).
- `vrid` - The ID for an existing virtual router.

For example, to view all redundant VIPs on the CSS, enter:

```
(config)# show redundant-vips
```

Table 6-4 describes the fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Address</td>
<td>The IP interface address associated with the redundant VIP.</td>
</tr>
<tr>
<td>VRID</td>
<td>The assigned identifier associated with the virtual router.</td>
</tr>
<tr>
<td>Redundant Address</td>
<td>The IP address of the VIP.</td>
</tr>
<tr>
<td>Range</td>
<td>The range associated with the VIP.</td>
</tr>
</tbody>
</table>
Displaying Redundant VIP and Virtual IP Interface Configurations

Use the `show virtual-routers` command to display a list of all virtual routers configured on the CSS. You may provide an interface IP address option to display only the virtual routers present on a particular interface. You may also include a VRID to display only the information for a particular virtual router.

The syntax for this command is:

```
show virtual-routers {ip_address {vrid}}
```

The optional variables are:

- `ip_address` - The address for the redundant interface. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1).
- `vrid` - The ID for an existing virtual router.

For example, to view all virtual routers on the CSS, enter:

```
(config)# show virtual-routers
```
Table 6-5 describes the fields.

**Table 6-5  Field Descriptions for the show virtual-routers Command**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Address</td>
<td>The IP interface address associated with the virtual router.</td>
</tr>
<tr>
<td>VRID</td>
<td>The assigned identifier associated with the virtual router.</td>
</tr>
<tr>
<td>Priority</td>
<td>The priority currently being advertised by the virtual router. Because the priority is dependent on the state of the critical services, the priority may be different than the one configured.</td>
</tr>
<tr>
<td>Config. Priority</td>
<td>The configured priority.</td>
</tr>
<tr>
<td>State</td>
<td>Current state of the virtual router. Possible states are:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Master</strong> - The virtual router is master.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Backup</strong> - The virtual router is backup.</td>
</tr>
<tr>
<td></td>
<td>- <strong>No Service</strong> - One or more critical services associated with the virtual router is down.</td>
</tr>
<tr>
<td></td>
<td>- <strong>IF Down</strong> - The IP interface associated with the virtual router is down.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Idle</strong> - The virtual router does not have any virtual interfaces or VIPs associated with it.</td>
</tr>
<tr>
<td>Master IP</td>
<td>The IP address of the master virtual router.</td>
</tr>
<tr>
<td>State Changes</td>
<td>The number of times the virtual router state has changed.</td>
</tr>
<tr>
<td>Last Change</td>
<td>The data and time of the virtual router last state change.</td>
</tr>
<tr>
<td>Preempt</td>
<td>True if preemption is enabled for the virtual router; false otherwise.</td>
</tr>
<tr>
<td>Critical-Services</td>
<td>The names of the critical services.</td>
</tr>
</tbody>
</table>
Table 6-5  Field Descriptions for the show virtual-routers Command (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>The current condition of the critical service. Possible states are: Alive, Down, or Suspended.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of critical service. Possible types are: Scripted, RedundancyUp, or Local.</td>
</tr>
</tbody>
</table>

VIP and Virtual IP Interface Redundancy Running-Config Examples

The following running-config examples show VIP redundancy configured on two CSSs.

CSS-Boston Running-Config

```
!******************************************************************************* GLOBAL *******************************************************************************
!******************************************************************************* INTERFACE *******************************************************************************
interface e1
  bridge vlan 1
!******************************************************************************* CIRCUIT  *******************************************************************************
circuit VLAN1
  ip address 192.168.8.1 255.255.255.0
  ip virtual-router 1 priority 230 preempt
  ip redundant-vip 1 192.168.8.10 shared
  ip critical-service 1 serv1
!******************************************************************************* SERVICE  *******************************************************************************
service serv1
  ip address 20.1.1.1
  active
!******************************************************************************* OWNER  *******************************************************************************
owner arrow
content L5_1
  protocol tcp
  vip address 192.168.8.10
  port 80
  url "/*
  add service serv1
  active
```
CSS-Cambridge Running-Config

!********************** GLOBAL **********************
!****************** INTERFACE **********************
interface e1
  bridge vlan 1
!******************* CIRCUIT *******************
circuit VLAN1
  ip address 192.168.8.2 255.255.255.0
  ip virtual-router 1 priority 200
  ip redundant-vip 1 192.168.8.10 shared
  ip critical-service 1 serv2
!**************** SERVICE *****************
service serv2
  ip address 20.2.2.2
  active

!*************************** OWNER ***************************
owner arrow
content L5_1
  protocol tcp
  vip address 192.168.8.10
  port 80
  url "/**"
  add service serv2
  active
Chapter 6    Configuring VIP and Virtual IP Interface Redundancy

Configuring Adaptive Session Redundancy

Configure Adaptive Session Redundancy (ASR) on 11500 series CSS peers in an active-backup VIP redundancy and virtual IP interface redundancy environment to provide stateful failover of existing flows. ASR ensures that, if the master CSS fails, the backup CSS has the necessary flow state information to continue any active flows (including TCP and UDP) without interruption when the backup CSS assumes mastership. “Adaptive” means that you can configure ASR on a per content rule basis.

Use ASR for:

- Mission-critical enterprise applications.
- Long-lived flows such as FTP and HTTP file transfers.
- E-commerce applications, such as online stock trading or banking where users must remain connected to a service for the duration of a transaction even if the master CSS fails.

In an ASR configuration, CSSs replicate flows that are:

- Fully-resolved (the master CSS has received a SYN/ACK from a server)
- Set up using content rules, services, and source groups that you specify as redundant

Note
For implicit or explicit Layer 5 rules, where there is delayed binding, binding is not complete until the CSS processes the SYN/ACK from the server. If a failover occurs in the middle of a spanned content request, the master CSS will not receive the SYN/ACK from the server and the flow will not be replicated on the backup CSS. No data is lost and users can simply refresh their browsers to restart the connection.

Note
During an FTP failover, the control channel and/or the data channel need to share information with the backup CSS. If the current state information has not been fully transferred across the ISC link to the backup CSS, then the flow may be lost.
Stateful Failover

Active flows that match a redundant content rule, service, or source group on the master CSS are replicated as dormant flows on the backup CSS peer. A dormant flow contains all the flow state information necessary for the backup CSS to take over the flow if the master CSS fails, including the flow ID assigned by the session processor (SP) that created the flow. If the master CSS fails, the dormant flows on the backup CSS become active when the backup CSS assumes mastership of the VIP. In turn, the active flows on the former master CSS transition to a dormant state to fully back up the active flows on the new master CSS.

A master CSS maps a newly activated TCP flow after it receives the first packet for the flow. If it can resolve a single route back to the source address, a CSS attempts to map a UDP flow when it activates the flow. Otherwise, the CSS maps the UDP flow after it receives the first packet for the flow.
Inter-Switch Communications

In an ASR configuration, CSS peers share redundant flow state information over a maximum of two private Inter-Switch Communications (ISC) links after booting. ISC is a messaging service used by CSSs to exchange flow state information. Only one ISC link is active at a time. The other ISC link (if configured) remains in backup mode until needed.

To determine if an ISC link is up, a CSS uses a mechanism called LifeTick. LifeTick sends an asynchronous message that contains information about the selected path. If the CSS does not receive a LifeTick message within one second, the CSS considers the ISC link to be down. If a second link is configured, the CSS uses that link for ISC.

The ISC links use the Gigabit Ethernet ports or the Fast Ethernet ports on the CSS session processors (SPs) to send ISC messages containing the flow state information. Once you configure the ISC ports, you cannot use those same ports for non-ISC traffic.

Note

You must connect the ISC ports directly to the two CSSs. You cannot use L2 devices on the ISC links between the two CSSs. Also, the ISC links must be dedicated to passing only ISC traffic.

For new flows, CSSs exchange flow states in real time over the ISC links. For existing flows, CSSs exchange flow states at boot-up time and at VIP redundancy failover.

Redundant Indexes

ASR uses unique global redundant indexes to keep track of content rules, services, and source groups configured on the redundant CSS peers. Set up the redundant indexes in rules, services, and groups using the `redundant-index` command. You must then configure identical redundant content rules, services, and source groups on CSS peers in the ASR configuration.

Each redundant index that you configure on a rule, service, or group must be unique among all rules, services, or groups configured on a redundant pair of CSSs. For example, if you configure a rule with a redundant index of 1 on a pair of CSSs, you cannot configure an index of 1 on another rule. However, you could configure an index of 1 on a group or service if that value has not already been used on a group or a service.
If you run traffic to a configuration that contains discrepancies between the redundant indexes on the two CSSs, the CPU utilization for each processor on the CSS may climb to an abnormal level (at 2000 flows/second, approximately 50 percent utilization for each processor). If you set the logging level to notice-5 or higher, the SCM utilization may peak at approximately 90 percent because each connection generates a redundant index mismatch log entry. For example: AUG 7 14:12:15 3/1 1124272 SLR-5: Rejected. Redundant global rule index (7) not found.

**Configuration Requirements and Restrictions**

The following requirements and restrictions apply to both CSS peers in an ASR configuration:

- Configure VIP/virtual IP interface redundancy on both CSS peers. For details, see “Configuring VIP and Virtual IP Interface Redundancy” earlier in this chapter.
- Configure a redundant VIP in a redundant content rule or source group. In order to activate a redundant content rule or source group, you must associate the rule or group with a redundant VIP.
- Ensure that VIP ranges specified in redundant content rules and source groups are the same as the VIPs associated with virtual routers for VIP redundancy. If the redundant content rule or source group VIPs are a superset, ASR is supported only for the VIPs that are associated with the virtual routers. For the remaining VIPs, the behavior is undefined when a failover occurs, because it is unclear whether those VIPs are mastered on the new master CSS or not.
- You cannot configure VIP wildcard or double-wildcard caching rules because they do not require a VIP. For information on wildcard cache rules, refer to the *Cisco Content Services Switch Basic Configuration Guide*, Chapter 7, Configuring Caching.
- Configure ISC on both CSSs. This allows the CSSs to share flow state information.
Configure a maximum of two ISC ports on a CSS. Multiple ports must reside on the same module in the CSS 11503 or CSS 11506 or on the same CSS 11501. Also, the ports must be of the same type (Gigabit Ethernet or Fast Ethernet) in both CSSs. Ensure that the ISC ports are not configured in any VLANs. If necessary, remove the designated ports from all VLANs before configuring ISC. For details on disabling an interface port from a VLAN, refer to the *Content Services Switch Administration Guide*.

You must connect the ISC ports directly to the two CSSs. You cannot use L2 devices on the ISC links between the two CSSs. Also, the ISC links must be dedicated to passing only ISC traffic.

- If you configure any ISC ports on an SCM, you can have only one SCM installed in the CSS 11506.
- The CSS 11501 does not support redundant GE Inter-Switch Communications links for ASR because the switch includes only a single GBIC port.
- Ensure that any service configured with connection limits, marked as redundant, and used by at least one redundant content rule is used only by other content rules that are also redundant. If this is not true, there could be redundant and nonredundant flows connected to the service with connection limits. In case of a failover, no information is available for the nonredundant flows on the backup CSS. Until the server cleans up the nonredundant flow connections, they continue to contribute to the connection limit on the service without the backup CSS having any knowledge of how many such connections exist. Making all flows redundant by imposing the above restrictions eliminates this problem.
- When you configure critical services, be sure to change the default keepalive settings to the following recommended settings for ASR. For example, enter:

```bash
service CriticalService
  ip address 192.168.2.1
  keepalive frequency 2
  keepalive maxfailure 2
  keepalive retryperiod 2
  active
```

**Note** The above keepalive values are a recommended starting point. Some scripted keepalives may take longer than two seconds to run. You may need to adjust your keepalive values so that the CSS detects a failure before your application times out.
- Configure as redundant any source groups that you specify in ACL clauses. It is helpful to configure ACLs similarly on the master and backup CSSs. This ensures that the CSSs can share the portmap state during flow setup time, and, at failover time, a CSS finds the same ACL and source group configured on the peer. Otherwise, when a flow fails over to the backup, it is possible that the flow may match on a different ACL clause that has no source group configured or a different source group (possibly a nonredundant one).

Source groups selected by ACL-checking always take precedence over other source group matches for a flow. Therefore, if the master and backup CSSs have different ACL definitions, when a flow fails over to the backup and the source group selected on the master is not found on the backup, the CSS rejects the flow. Also, if the flow matches on a different source group through an ACL, that source group takes precedence over the redundant source group that was sent from the master.

- Configure as redundant any preferred service that you configured in an ACL clause.

- Configure mutually exclusive portmap ranges on the redundant peers using the `global-portmap` command to avoid potential network port collisions. Keeping the portmap ranges mutually exclusive on the redundant peer also eliminates the need to dynamically update the global portmap database on the backup CSS. For more information on portmapping, refer to the Cisco Content Services Switch Administration Guide, Chapter 6, Configuring User Profiles and CSS Parameters.

- Do not configure ASR and stateless redundancy failover on the same CSS. Such a configuration is not supported. For details on stateless redundancy failover, refer to Chapter 7, Configuring Redundant Content Services Switches, in the section “Configuring Stateless Redundancy Failover”.

- ASR does not support NAT Peering. For details on NAT Peering, refer to the Content Services Switch Basic Configuration Guide.
Adaptive Session Redundancy Quick Start

Table 6-6 provides a quick overview of the steps required to configure ASR for each CSS in the redundant configuration. Each step includes the CLI command or a reference to the procedure required to complete the task. For a complete description of each feature and all the options associated with the CLI command, refer to the sections following Table 6-6.

<table>
<thead>
<tr>
<th>Task and Command Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enter config mode.</td>
</tr>
<tr>
<td># config</td>
</tr>
<tr>
<td>(config)#</td>
</tr>
<tr>
<td>2. Configure active/backup VIP/virtual IP interface redundancy. See “Configuring VIP and Virtual IP Interface Redundancy” earlier in this chapter.</td>
</tr>
<tr>
<td>(config)# interface 1/1</td>
</tr>
<tr>
<td>(config-if[ 1/1])# isc-port-one</td>
</tr>
<tr>
<td>(config)# interface 1/2</td>
</tr>
<tr>
<td>(config-if[ 1/2])# isc-port-two</td>
</tr>
<tr>
<td>3. Configure a maximum of two directly connected (no intervening L2 devices) ISC links on Gigabit Ethernet or Fast Ethernet ports between the two redundant CSSs. See “Configuring Inter-Switch Communications” later in this chapter.</td>
</tr>
<tr>
<td>4. Configure services that are targets of redundant content rules. For more information on services, refer to the Cisco Content Services Switch Basic Configuration Guide, Chapter 1, Configuring Services.</td>
</tr>
<tr>
<td>(config)# service server1</td>
</tr>
<tr>
<td>(config-service[server1])# ip address 192.168.100.100</td>
</tr>
<tr>
<td>(config-service[server1])# redundant-index 1</td>
</tr>
<tr>
<td>(config-service[server1])# active</td>
</tr>
</tbody>
</table>
Chapter 6  Configuring VIP and Virtual IP Interface Redundancy

Configuring Adaptive Session Redundancy

5. Configure redundant content rules and add the redundant services. For more information on content rules, refer to the Cisco Content Services Switch Basic Configuration Guide, Chapter 3, Configuring Content Rules.

```
(config)# owner arrowpoint
(config-owner[arrowpoint])# content rule1
(config-owner-content[arrowpoint-rule1])# vip address 192.168.1.1
(config-owner-content[arrowpoint-rule1])# protocol tcp
(config-owner-content[arrowpoint-rule1])# port 80
(config-owner-content[arrowpoint-rule1])# url "/redundant.html"
(config-owner-content[arrowpoint-rule1])# add service server1
(config-owner-content[arrowpoint-rule1])# redundant-index 5
(config-owner-content[arrowpoint-rule1])# active
```

6. Configure redundant source groups and add the redundant services. For more information on source groups, refer to the Cisco Content Services Switch Basic Configuration Guide, Chapter 5, Configuring Source Groups, ACLs, EQLs, URQLs, NQLs, and DQLs.

```
(config)# group group1
(config-group[group1])# vip address 192.168.10.10
(config-group[group1])# add service server1
(config-group[group1])# redundant-index 4
(config-group[group1])# active
```

7. Configure global portmapping (port translation) with mutually exclusive port ranges on the CSS peers to avoid potential port collisions. For more information on CSS portmapping, refer to the Cisco Content Services Switch Administration Guide, Chapter 6, Configuring User Profiles and CSS Parameters.

For example, on one CSS peer, enter:

```
(config)# global-portmap base-port 3000 range 30000
```

On the other CSS peer, enter:

```
(config)# global-portmap base-port 33100 range 30000
```

8. Configure the same redundant services, content rules, and source groups on the other CSS peer (synchronize the configurations).
Configuring Inter-Switch Communications

Inter-Switch Communications (ISC) is a messaging service that 11500 series CSS peers use to exchange flow state information in an ASR configuration. If the master CSS fails, the backup CSS already has the flow state information necessary to continue the current flows without interruption. Using ISC, CSSs exchange state information:

- For existing flows at boot-up time and at VIP redundancy failover
- For new flows in real time (after the CSS receives a SYN/ACK from the server)

Use the `isc-port-one` and `isc-port-two` commands in interface configuration mode to enable ISC between two 11500 series CSSs in an ASR configuration. You can configure a maximum of two ISC ports on each CSS. The two ports must be of the same type (Gigabit Ethernet or Fast Ethernet) and must be on the same module in the CSS 11503 or CSS 11506 or on the same CSS 11501.

The CSS 11501 does not support redundant GE Inter-Switch Communications links for ASR because the switch includes on a single GBIC port.

You must connect the ISC ports directly to the two CSSs. You cannot use L2 devices on the ISC links between the two CSSs. Also, the ISC links must be dedicated to passing only ISC traffic.

For example, to enable both ISC ports on a CSS 11506, enter:

```
(config)# interface 1/1
(config-IF[ 1/1])# isc-port-one
(config-IF[ 1/1])# interface 1/2
(config-IF[ 1/2])# isc-port-two
```

To disable both ISC ports on a CSS 11506, enter:

```
(config)# interface 1/1
(config-IF[ 1/1])# no isc-port-one
(config-IF[ 1/1])# interface 1/2
(config-IF[ 1/2])# no isc-port-two
```
Configuring Redundant Services

Use the `redundant-index` command to configure the global service index for a redundant service. A CSS uses the global service index to keep track of redundant services and associated flow state information.

The syntax for this service configuration mode command is:

```
redundant-index index
```

The variable `index` is a unique number you assign to a redundant service. Enter a unique integer from 0 to 32767, where a value of 0 disables ASR for a service. The default is 0, but it does not appear in the running-config even if you configure it explicitly.

For example:

```
(config-service[server1])# redundant-index 5
```

To disable ASR for a service, enter:

```
(config-service[server1])# no redundant-index
```

**Note**

If you issue the `no redundant-index` command on an active redundant service for live redundancy peers, the command automatically suspends the service. Flows already mapped by a CSS are not affected. However, if a failover occurs during the life of an active flow that matches on such a suspended service, the backup CSS cannot map the flow because it cannot find the service with the same global index as that on the original master.

For more information on configuring services, refer to the *Cisco Content Services Switch Basic Configuration Guide*, Chapter 1, Configuring Services.

Configuring Redundant Content Rules

Use the `redundant-index` command to configure the global content index for a redundant content rule. A CSS uses the global content index to keep track of redundant content rules and associated flow state information.

The syntax for this content configuration mode command is:

```
redundant-index index
```
The variable *index* is a unique number you assign to a redundant content rule. Enter a unique integer from 0 to 32767, where a value of 0 disables ASR on a content rule. The default is 0, but it does not appear in the running-config even if you configure it explicitly.

For example:

```
(config-owner-content[arrowpoint-rule1]# redundant-index 1
```

To disable ASR on a content rule, enter:

```
(config-owner-content[arrowpoint-rule1]# no redundant-index
```

---

**Note**

If you issue the `no redundant-index` command on an active redundant content rule for live redundancy peers, the command automatically suspends the content rule. Flows already mapped by a CSS are not affected. However, if a failover occurs during the life of an active flow that matches on such a suspended content rule, the backup CSS cannot map the flow because it cannot find the content rule with the same global index as that on the original master.

For more information on configuring content rules, refer to the *Cisco Content Services Switch Basic Configuration Guide*, Chapter 3, Configuring Content Rules.

### Configuring Redundant Source Groups

Use the `redundant-index` command to configure the global source group index for a redundant source group. A CSS uses the global source group index to keep track of redundant content rules and associated flow state information.

The syntax for this group configuration mode command is:

```
redundant-index index
```

The variable *index* is a unique number you assign to a redundant source group. Enter a unique integer from 0 to 32767, where a value of 0 disables ASR for a source group. The default is 0, but it does not appear in the running-config even if you configure it explicitly.
For example, to enable ASR for a source group:

```
(config-group[group1])# redundant-index 4
```

To disable ASR for a source group, enter:

```
(config-group[group1])# no redundant-index
```

**Note**

If you issue the `no redundant-index` command on an active redundant source group on live redundancy peers, the command automatically suspends the source group. Flows already mapped by a CSS are not affected. However, if a failover occurs during the life of an active flow that matches on such a suspended source group, the backup CSS cannot map the flow because it cannot find the source group with the same global index as that on the original master.

For more information on configuring source groups, refer to the *Cisco Content Services Switch Basic Configuration Guide*, Chapter 5, Configuring Source Groups, ACLs, EQLs, URQLs, NQLs, and DQLs.

## Synchronizing Adaptive Session Redundancy Configurations

You must synchronize configurations on both CSS peers to ensure that the ASR-specific configurations on the master CSS and the backup CSS are the same. This is critical to the proper functioning of ASR.

For ASR, you must manually configure on each peer:

- ISC
- Redundant content rules
- Redundant services
- Redundant source groups
Displaying Adaptive Session Redundancy Information

Use the following commands to display information for:

- Inter-Switch Communications
- Dormant flows used for ASR
- ASR status and global redundant indexes

Displaying Inter-Switch Communications Ports

Use the `show isc-ports` command to display the ports configured for ISC on an 11500 series CSS.

Table 6-7 describes the fields in the `show isc-ports` output.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Inter-Switch Communications Configuration | Lists the CSS ports (in slot/port format) configured for ISC port one and ISC port two. If ISC is not configured, the command displays the following messages:  
  Inter-Switch Port One is not configured.  
  Inter-Switch Port Two is not configured. |
| Inter-Switch Communications Status    | Indicates whether ISC is Up or Down and, if Up, on which CSS port ISC is currently active. |
Displaying Dormant Flow Information

Use the `show dormant flows` command to display information about the current dormant flows on the backup CSS in an ASR configuration. Dormant flows are flows on the backup CSS that become active if the master CSS fails and the backup CSS assumes mastership.

The syntax for this command is:

```
show dormant flows {source_address {destination_address}}
```

The optional variables for this command are:

- `source_address` - Displays dormant flows for the specified source IP address. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).
- `destination_address` - Displays dormant flows for the specified destination IP address. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).

Table 6-8 describes the fields in the `show dormant flows` output.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Src Address</td>
<td>The source address for the flow.</td>
</tr>
<tr>
<td>SPort</td>
<td>The source port for the flow.</td>
</tr>
<tr>
<td>Dst Address</td>
<td>The destination address for the flow.</td>
</tr>
<tr>
<td>DPort</td>
<td>The destination port for the flow.</td>
</tr>
<tr>
<td>NAT Dst Address</td>
<td>The network address translation (NAT) destination address.</td>
</tr>
<tr>
<td>Prt In</td>
<td>Not applicable. A dormant flow does not have a port associated with it.</td>
</tr>
<tr>
<td>OutPort</td>
<td>Not applicable. A dormant flow does not have a port associated with it.</td>
</tr>
</tbody>
</table>
Use the `flow statistics dormant` command to display summary information about redundant dormant flows.

Table 6-9 describes the field in the `flow statistics dormant` output.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Dormant Flows</td>
<td>The total number of inactive redundant flows mapped on the backup CSS from active redundant flows on the master CSS. The dormant flows contain all the flow state information necessary for the backup CSS to master the flows if the master CSS fails. If the master CSS fails, the backup CSS becomes the master CSS and the dormant flows become active flows.</td>
</tr>
</tbody>
</table>

**Displaying ASR Information for Content Rules, Services, and Source Groups**

The following sections describe how to display:

- ASR status and global index values
- Summary ASR information

**Displaying ASR Status and Global Index Values**

Use the `show rule`, `show service`, and `show group` commands to display information about ASR status and global redundant indexes. The relevant fields in the output of these commands are:

- **Session Redundancy** - The state of ASR for the content rule, service, or source group. Possible values are: Enabled or Disabled
- **Redundancy Global Index** - The unique global index value for ASR configured for the content rule, service, or source group using the `redundant-index` command.

For full details on the `show rule`, `show service`, and `show group` commands, refer to the *Cisco Content Services Switch Basic Configuration Guide*. 
Displaying Summary ASR Information

Use the `show session-redundant` command to display summary ASR information about redundant content rules, services, and source groups.

The syntax for this global configuration mode command is:

```
show session-redundant [rule|service|group|all]
```

The optional keywords are:

- `rule` - Displays summary ASR information for redundant content rules.
- `service` - Displays summary ASR information for redundant services.
- `group` - Displays summary ASR information for redundant source groups.
- `all` - Displays summary ASR information for content rules, services, and source groups.

For example, to view summary ASR information for redundant content rules, enter:

```
(config)# show session-redundant rule
```

Table 6-10 describes the fields.

### Table 6-10  Field Descriptions for the `show session-redundant` Command

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Session Redundant Content Rules</strong></td>
<td></td>
</tr>
<tr>
<td>Content Rule</td>
<td>The redundant content rule name.</td>
</tr>
<tr>
<td>Content Rule State</td>
<td>The current state of the redundant content rule. Possible states are: Active or Suspend.</td>
</tr>
<tr>
<td>VIP Address</td>
<td>The virtual IP address of the redundant content rule in dotted decimal notation.</td>
</tr>
<tr>
<td>Redundancy Global Index</td>
<td>The ASR global index configured for the redundant content rule.</td>
</tr>
<tr>
<td>Redundancy State</td>
<td>The state of the CSS peer: Master, Backup, or Suspend.</td>
</tr>
<tr>
<td>Rule Redundant Services 1</td>
<td>The name of the redundant service and its global index value configured on the rule.</td>
</tr>
<tr>
<td><strong>Session Redundant Services</strong></td>
<td></td>
</tr>
</tbody>
</table>
Table 6-10  Field Descriptions for the show session-redundant Command (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>The name of the redundant service.</td>
</tr>
<tr>
<td>Service State</td>
<td>The current state of the redundant service. Possible states are: Alive, Dying, or Down.</td>
</tr>
<tr>
<td>IP Address</td>
<td>The virtual IP address of the redundant service in dotted-decimal notation.</td>
</tr>
<tr>
<td>Redundancy Global Index</td>
<td>The ASR global index configured for the redundant service.</td>
</tr>
</tbody>
</table>

Session Redundant Source Groups

| Source Group                       | The name of the redundant source group.                                     |
| Source Group State                 | The current state of the redundant source group. Possible states are: Active or Suspend. |
| VIP Address                        | The virtual IP address of the redundant source group.                       |
| Redundancy Global Index            | The ASR global index configured for the redundant source group.             |

Group Redundant Services

| Source Services                    | The redundant source services configured in this redundant source group, their keepalive state, and global index. If no source services are configured in this source group, the value is NONE. |
| Destination Services              | The redundant destination services configured in this redundant source group and their keepalive state. If no destination services are configured in this source group, the value is NONE. |