



Configuring Content Replication

This chapter describes how to configure demand-based content replication and content staging and replication.



Note

The Demand-Based Content Replication and the Content Staging and Replication features are part of the CSS Enhanced feature set.

The information in this chapter applies to all CSS models, except where noted.

This chapter contains the following major sections:

- [Configuring Demand-Based Content Replication](#)
- [Configuring Content Staging and Replication](#)

Configuring Demand-Based Content Replication

One of the biggest challenges for a Web site includes managing unpredictable traffic and flash crowds caused by sudden hot content. Using demand-based content replication, the CSS can track content requests and identify and replicate hot content to overflow Web servers or caches dynamically.

Demand-based content replication is traffic-based. Increases in the flow of traffic launch replication services automatically. When you configure demand-based content replication, the CSS automatically:

1. Uses hot lists to detect hot content when the URL hits or bandwidth exceeds the configured hot list threshold.
2. Modifies the content rules dynamically to provide additional services from which the hot content may be served.

The following sections describe how to configure service replication:

- [Configuring Hot Lists](#)
- [Specifying Service Type for Replication](#)
- [Configuring Max Age](#)
- [Configuring Max Content](#)
- [Configuring Max Usage](#)
- [Configuring FTP Access for Content Replication](#)
- [Creating an FTP Record](#)

Configuring Hot Lists

Use the **hotlist** command to define a hot list that lists the content most requested (hot content) during a user-defined period of time. The CSS enables you to configure hot-list attributes for content rules. Defining hot-list attributes for a content rule enables you to determine which content is heavily accessed. With this information, you can accurately determine which content should be replicated.



Note

You must configure and enable a hot list for service types replication-store and replication-cache to work.

You can configure the following hot-list attributes for specific content from config-owner-content mode:

- **hotlist** - Enables the hot list. To enable a hot list for a specific content rule, enter the **hotlist** command from the corresponding owner-content mode. For example:

```
(config-owner-content [arrowpoint-rule1])# hotlist
```

To disable a hot list, enter:

```
(config-owner-content [arrowpoint-rule1])# no hotlist
```

- **hotlist interval** - Sets the hot-list refresh interval. Enter the interval time from 1 to 60 minutes. The default is 1. For example:

```
(config-owner-content [arrowpoint-rule1])# hotlist interval 10
```

To restore the hot-list interval to the default of 1, enter:

```
(config-owner-content [arrowpoint-rule1])# no hotlist interval
```

- **hotlist size** - Sets the size of the hot list. Enter the total number of entries maintained for this rule from 1 to 100. The default is 10. For example:

```
(config-owner-content [arrowpoint-rule1])# hotlist size 20
```

To restore the hot-list size to the default of 10, enter:

```
(config-owner-content [arrowpoint-rule1])# no hotlist size
```

- **hotlist threshold** - Sets the hot-list threshold. Enter an integer from 0 to 65535 to specify the threshold above which a piece of content is considered hot. The default is 0. For example:

```
(config-owner-content [arrowpoint-rule1])# hotlist threshold 9
```

To restore the hot-list threshold default of 0, enter:

```
(config-owner-content [arrowpoint-rule1])# no hotlist threshold
```

- **hotlist type hitCount** - Sets the hot-list type to hit count, which is how many times the content was accessed. For example:

```
(config-owner-content [arrowpoint-rule1])# hotlist type hitcount
```

To restore the hot-list type to the default setting **hitCount**, enter:

```
(config-owner-content [arrowpoint-rule1])# no hotlist type
```

Specifying Service Type for Replication

Within a replication configuration, you must configure at least two servers: one local and one replication type. The CSS provides the following service types specific to replication:

- **type rep-cache-redirect** - Specifies the service is a replication cache with redirect.
- **type rep-store** - Specifies the service is a replication store, which is a local overflow service used to load balance content requests.
- **type rep-store-redirect** - Specifies the service is a replication store to which content requests are redirected. No content rules are applied to requests from this service type.

When you specify a service as **type rep-cache-redirect**, the CSS uses the service as a cache server, caching hot content and sending requests to it. Once content is cached on the replication server, the CSS creates a dynamic content rule for the hot content and a dynamic service.

The CSS deletes the hot content when the max-age time has elapsed. See the section, “[Configuring Max Age](#)” later in this chapter.

For example:

```
(config)# service serv1
(config-service[serv1])# type rep-cache-redirect
```

When you specify a service as **type rep-store**, the CSS replicates hot content on the service. Once content is replicated on the replication server, the CSS creates a dynamic content rule for the hot content automatically. The dynamic content rule inherits all the attributes of the existing rule with the following changes:

- Specifically identifies the hot content
- Changes the server type from replication-store to type local

The CSS deletes the dynamic content rule after the maximum age time elapses. See the following section, “[Configuring Max Age](#)”. The CSS lists the dynamic content rule in the **show rule** display. It is not displayed in the running- or startup-config files.



Note

A replication service type is not included in the load balancing algorithm until content is replicated on the service.

For example:

```
(config)# service serv1
(config-service[serv1])# type rep-store
```

Configuring Max Age

Use the **max age** command to define the maximum age for replicated objects on services defined as type **rep-cache-redis**, **rep-store**, or **rep-store-redis**. Enter the maximum age in minutes from 1 to 1440. The default is 120.

For example:

```
(config-service[serv1])# max age 10
```

To set the maximum age for replicated objects to its default value of 120, enter:

```
(config-service[serv1])# no max age
```

Configuring Max Content

Use the **max content** command to define the maximum pieces of content for replication on services defined as type **rep-cache-redis**, **rep-store**, or **rep-store-redis**. Enter the maximum pieces of content from 1 to 65535. The default is 100.

For example:

```
(config-service[serv1])# max content 50
```

To set the maximum content to its default value of 100, enter:

```
(config-service[serv1])# no max content
```

Configuring Max Usage

Use the **max usage** command to define the maximum disk space allowed for replication on services defined as type **rep-cache-redis**, **rep-store**, or **rep-store-redis**. Enter the disk space for a service from 1 to 1000 megabytes. The default is 1.

For example:

```
(config-service[serve1])# max usage 100
```

To set the maximum disk space to its default value of 1, enter:

```
(config-service[serve1])# no max usage
```

Configuring FTP Access for Content Replication

Use the **access ftp** command to associate an FTP access mechanism with a service for demand-based replication activities. You must use this command for each service that offers publishing services.

When you use this command to associate an FTP access mechanism to a service, the base directory of an existing FTP record becomes the tree root. To maintain coherent mapping between WWW daemons and FTP daemons, make the FTP access base directory equivalent to the WWW daemon root directory as seen by clients.

Enter the access *ftp_record* as the name of an existing FTP record. Enter the FTP record name as an unquoted text string with no spaces.



Note

To create an FTP record, use the **(config) ftp-record** command. For more information on creating an FTP record, see [“Creating an FTP Record”](#) later in this chapter.

For example:

```
(config-service[serve1])# access ftp myftprecord
```

To remove a service access mechanism, enter:

```
(config-service[serve1])# no access ftp
```



Note

Content replication does not support the WSFTP FTP application.

Creating an FTP Record

To create a File Transfer Protocol (FTP) record file to use when accessing an FTP server from the CSS, use the **ftp-record** command. The syntax for this global configuration mode command is:

```
ftp-record ftp_record ip_address or hostname username  
[“password”|encrypted-password password] base_directory
```

The variables are:

- *ftp_record* - The name for this FTP record file. Enter an unquoted text string with no spaces and a maximum length of 16 characters.
- *ip_address* or *hostname* - The IP address or host name of the FTP server you want to access.
- *username* - A valid login username on the FTP server. Enter a case-sensitive unquoted text string with no spaces and a maximum of 32 characters.
- *“password”* - The password for the login username on the FTP server. Enter a case-sensitive quoted text string with no spaces and a maximum of 16 characters.
- **encrypted-password** *encrypted_password* - The encrypted password for the valid login username on the FTP server. Enter a case-sensitive unquoted text string with no spaces and a maximum of 16 characters.
- *base_directory* - An optional base directory when using this record.

For example:

```
(config)# ftp-record ftp1 172.16.6.58 bobo "secret" /
```

To delete an FTP record file from the CSS, use the **no ftp-record** command and the ftp record name. For example:

```
(config)# no ftp-record ftp1
```

Configuring Content Staging and Replication

The CSS supports content staging and replication using Publisher and Subscriber services. Content staging and replication is a timer-based replication service. At a pre-configured day and time, the CSS takes content (for example, a file, multiple files, or complete directories) that you post to the staging publisher server and replicates the content dynamically to multiple subscriber servers based on:

- CLI commands.
- Time of day using the Command Scheduler feature.
- Detected changes to specific content on the staging server. The CSS then replicates that content to the subscriber servers or caches dynamically.

The CSS detects changes to specific content by performing an FTP-based examination of filenames, sizes, and file dates. The CSS performs this examination based on the configured publisher interval or by the **replicate** command. The subscriber knows how to interface to the publisher by virtue of the 'access ftp' associated with the publisher designated service.

You can configure the CSS to continually update content that has been replicated. For example, the CSS can replicate content associated with a breaking news story. You can post updates to the staging server and the updates will be replicated to all distributed locations automatically.

Publisher and Subscriber services are usually defined as type **local**. There is no need to change the service type.

The following sections describe how to configure publisher and subscriber services:

- [Configuring FTP Access for Publishing and Subscribing](#)
- [Configuring a Publishing Service](#)
- [Configuring a Subscriber Service](#)
- [Configuring a Content Rule for Content Staging and Replication](#)
- [Configuring Publisher Content Replication](#)
- [Displaying Content](#)

Configuring FTP Access for Publishing and Subscribing

Use the **access ftp** command to associate an access mechanism with a service for use during publishing and subscribing activities. You must use this command for each service that offers publishing services and for each service that you configure as a subscriber.

Enter the FTP record as the name of an existing FTP record. Enter the FTP record name as an unquoted text string with no spaces.



Note

When you configure content staging and replication, you must create the FTP record prior to configuring any other content staging and replication command or the feature will not work properly. To create an FTP record, use the **(config) ftp-record** command. For more information see [“Creating an FTP Record”](#) earlier in this chapter.

The syntax for this service mode command is:

```
(config-service [pubserver]) # access ftp myftprecord
```

To remove a service access mechanism, enter:

```
(config-service [pubserver]) # no access ftp
```

Configuring a Publishing Service

Use the **publisher** command to configure a service as a publishing service. A publishing service can be any type of service that applies to your applications (for example, local or proxy-cache). For a complete description of service types, refer to the *Cisco Content Services Switch Basic Configuration Guide*.

A publishing service synchronizes content among associated subscriber services. To move the content during publishing activities, you must configure an access mechanism for the publisher service. Use the **(config-service) access ftp** command defined earlier in this chapter to configure a mechanism for the publisher service.

When you define the interval to synchronize the subscriber, the interval begins at the time you issue the command. Subscribers that are unavailable for synchronization are placed in an offline state and retried until the operation is completed.

There is no limit on the size of the files that a CSS can replicate between a publisher and a subscriber. When transferring data between a publisher and a subscriber, a CSS creates a virtual pipe so that the replicated files never use the CSS disk. The CSS uses the default buffering associated with the TCP communications stack.



Note

The publisher service does not become active until it has at least one configured subscriber. You do not need to configure the publisher before configuring the subscriber, but the publisher must be configured before the subscriber can receive any content synchronization updates.

The syntax and options for this service mode command are:

- **publisher** - Configures the service as a publishing service.
- **publisher interval** *minutes* - Defines a recurrent interval in minutes to synchronize content among the subscribers. You can enter this command only after you configure this service as a publishing service. Enter the synchronization interval in minutes. Enter the number from 0 to 3600. The default is 0 which disables the interval.
- **publisher interval** *minutes trigger_filename* - Defines a recurrent interval in minutes to synchronize content among the subscribers only when the specified trigger file is modified. Specify the *trigger_filename* from 1 to 64 characters in length. You can enter this command only after you configure the service as a publishing service.

To configure publishing on a service, enter:

```
(config-service [pubserver]) # publisher
```

To remove publishing on a service, enter:

```
(config-service [pubserver]) # no publisher
```

To configure a publisher resynchronization interval, enter:

```
(config-service [pubserver]) # publisher interval 120
```

To disable the publisher resynchronization interval by setting it to its default of 0, enter:

```
(config-service [pubserver]) # no publisher interval
```

Displaying Publisher Configurations

Use the **show publisher** command to display the operational status of the publishing service and content information. The options and syntax are:

- **show publisher** - Displays information about all configured publishing services.
- **show publisher *publisher_name*** - Displays information about the specified publishing service.
- **show publisher *publisher_name* content {verbose}** - Displays information about the content for the specified publishing service. Include the verbose option to display more detailed content information.

To display information about the publishing services, enter:

```
(config-service)# show publisher
```

[Table 9-1](#) describes the fields in the **show publisher** output.

Table 9-1 *Field Descriptions for the show publisher Command*

Field	Description
State	The state of the publisher service.
Access Type	The associated access mechanism with a service for use during publishing activities. Currently, the FTP record is the only mechanism.
Access IP	The IP address for the FTP record.
Access Port	The port number for the FTP record associated with the access mechanism.
Access Username	The username for the FTP server as defined through the FTP record.
Access Base Dir	The base directory as defined through the FTP record.
Published Files	The number of files published from the publisher to the subscriber.
Published Bytes	The number of bytes published from the publisher to its subscribers.

Table 9-1 Field Descriptions for the show publisher Command (continued)

Field	Description
Subscribers	The number of subscribers configured to use the publisher.
Trigger File	The file upon modification that causes the synchronization between the publisher and the subscriber.
Publish Interval	The interval in seconds when the publisher checks for subscriber synchronization.
Next Interval	The time when the next publisher synchronization check will occur.
Managed Files	The number of files that the publisher will replicate.
Subscribers Synced	The number of synchronized subscribers.
Managed Dirs	The number of files that the publisher will replicate.
Managed Bytes	The number of bytes that the publisher is tracking.
Last Method	The last method that caused the publisher to attempt synchronization with the subscriber. The synchronization methods are: <ul style="list-style-type: none"> • cli - User initiated • interval - The configured time interval • signal - Trigger file change • retry - Retry when a publisher failed to synchronize previously • reboot - CSS reboot
Last Time	The last time when the publisher attempted to synchronize with the subscriber.

Configuring a Subscriber Service

Use the **subscriber** command to configure a service as a subscriber to a publishing service. You can define a maximum of 31 subscribers per publisher.

You must configure an access mechanism for each subscriber. Use the **(config-service) access ftp** command defined earlier in this chapter to configure an access mechanism for each subscriber.

To configure a service as a subscriber to a publishing service, enter:

```
(config-service[subserver]) # subscriber publisher
```

To unsubscribe the service from a publishing service, enter:

```
(config-service[subserver]) # no subscriber
```



Note

A subscriber's state will not be ready or will be in access failure until the publisher's state is ready.

Displaying Subscriber Configurations

Use the **show subscriber** command to display the operational status of the subscriber services. The syntax is:

- **show subscriber** - Displays information about all configured subscriber services
- **show subscriber *publisher_name*** - Displays information about all subscriber services for the specified publishing service
- **show publisher *publisher_name subscriber_name*** - Displays information about the specified subscriber service for the specified publishing service

To display information about the subscriber services, enter:

```
(config)# show subscriber
```

Table 9-2 describes the fields in the **show subscriber** output.

Table 9-2 *Field Descriptions for the show subscriber Command*

Field	Description
State	The state of the subscriber.
Access Type	The FTP access mechanism with a service for use during subscribing activities.
Access IP	The IP address for the FTP record associated with the access mechanism.
Access Port	The port number for the FTP record associated with the access mechanism.
Access Username	The username for the FTP record associated with the access mechanism.
Access Base Dir	The base directory for the FTP record associated with the access mechanism.
Subscribed Files	The number of files replicated on the subscriber.
Subscribed Bytes	The number of bytes replicated on the subscriber.
Last Method	The last method that caused the publisher to attempt synchronization with the subscriber. The synchronization methods are: <ul style="list-style-type: none"> • cli - user initiated • interval - the configured time interval • signal - trigger file change • retry - retry when a publisher failed to synchronize previously • reboot - CSS reboot
Last Time	The last time when the publisher attempted to synchronize with the subscriber.
Synchronized	Indicates whether or not the subscriber is currently synchronized with the publisher.

Configuring a Content Rule for Content Staging and Replication

When you configure content staging and replication, you must configure a URL in a content rule to define which files you want replicated. Then add the subscriber services to the content rule.



Note

If you want all files in all directories replicated, you do not need to create a content rule. Create a content rule to specify only those files you want replicated.



Note

You cannot configure a URQL with subscriber services in a content rule.

For example, to specify a URL that matches all requests for content in the *announcements* directory with *.html* extensions, enter:

```
(config-owner-content [arrowpoint-products.html])# url
"/announcements/*.html"
```

For a complete description of configuring URLs, refer to the *Cisco Content Services Switch Basic Configuration Guide*.

To add the subscriber services to the content rule, use the **add service** command. For example:

```
(config-owner-content [arrowpoint-products.html])# add service
subscriber
```

Configuring Publisher Content Replication

Use the **replicate** command to start replicating content between a publisher and all associated subscribers. You can use this command to replicate content to new subscribers or force resynchronization immediately.

Enter the *publisher_name* as the name of the existing publisher. Enter the *subscriber_name* as the name of the subscriber associated with the publisher service.

The syntax and options are:

- **replicate** *publisher_name* - Resynchronizes any changes to content between the specified publisher and its subscriber services. If the content has not changed, no resynchronization occurs.
- **replicate** *publisher_name subscriber_name* - Resynchronizes any changes to content between the specified publisher and the specified subscriber service. If the content has not changed, no resynchronization occurs.
- **replicate** *publisher_name subscriber_name force* - Resynchronizes all content between the specified publisher and the specified subscriber service whether or not content changes have occurred.

For example:

```
# replicate pubserver
```

Displaying Content

The **show content** command enables you to display content entries in the Content Service Database (CSD) of a CSS. This command is available in all modes.

The syntax is:

```
show content slot slot_number {start-index index_number}
```

The variables and option are:

- **slot** *slot_number* - Display content from the module located in a specific slot in the CSS chassis. For the CSS 11503, the available choices are 1 through 3; for the CSS 11506, the available choices are 1 through 6. If you do not specify a slot number the CSS displays the content entries from the SCM in slot 1 of the CSS chassis.
- **start-index** *index_number* - Display content entries starting at the specified *index_number* parameter. This variable defines where you want to start browsing CSS content. Starting from the specified index number, you receive up to a maximum of 64 KB of information. To see additional information, issue the **show content** command again, starting from the last index number displayed. To specify an index number, enter a number from 0 to 4095. If you do not specify a start-index the CSS displays the content entries starting from 0.

For example, to look at the content from the module in chassis slot 2, starting at index 150, enter:

```
(config)# show content slot 2 start-index 150
```

Table 9-3 describes the fields in the **show content** output.

Table 9-3 *Field Descriptions for the show content Command*

Field	Description
Pieces of Content for Slot	The chassis slot number in which the module resides.
Subslot	The module slot number in which the Session Processor resides.
Total Content	The total number of content entries.
Index	Unique index for a known piece of content in the CSD.
<address>	The IP address of the piece of content.
Protocol	The IP Protocol of the piece of content.
Port	Protocol port of the piece of content.
Best Effort	The QoS class of the piece of content. This field is not used by the CSS at this time.
Streamed	Identifies if the piece of content is streaming media (video or audio). This field is not used by the CSS at this time.
URL	The Universal Resource Locator of the piece of content.
Domain	The domain name of the piece of content.

