



I

Understanding RAM Caching

- Introducing the RAM Cache feature
- Configuring the RAM Cache
- Using iRules with the RAM Cache
- Configuring bigdb settings for RAM Cache

Introducing the RAM Cache feature

This document describes how to configure the properties of the RAM Cache feature on the BIG-IP system. A **RAM cache** is a cache of HTTP objects stored in the BIG-IP system's RAM that are reused by subsequent connections to reduce the amount of load on the back-end servers.

Getting started with RAM Caching

There are several concepts to consider before you configure the RAM Cache on the BIG-IP system.

- When to use the RAM Cache
- What items you can cache
- The cache mechanism

When to use the RAM Cache

The RAM Cache feature provides the ability to reduce the traffic load to back-end servers. This ability is useful if an object on a site is under high demand, if the site has a large quantity of static content, or if the objects on the site are compressed.

◆ **High demand objects**

This feature is useful if a site has periods of high demand for specific content. With RAM Cache configured, the content server only has to serve the content to the BIG-IP system once per expiration period.

◆ **Static content**

This feature is also useful if a site consists of a large quantity of static content such as CSS, javascript, or images and logos.

◆ **Content compression**

For compressible data, the RAM Cache can store data for clients that can accept compressed data. When used in conjunction with the compression feature on the BIG-IP system, the RAM Cache takes stress off of the BIG-IP system and the content servers.

The items you can cache

The RAM Cache feature is fully compliant with the cache specifications described in RFC 2616, *Hypertext Transfer Protocol -- HTTP/1.1*. This means you can configure RAM Cache to cache the following content types:

- 200, 203, 206, 300, 301, and 410 responses
- Responses to GET methods by default.
- Other HTTP methods for URIs specified in the URI Include list or specified in an iRule.
- Content based on the User-Agent and Accept-Encoding values. The RAM Cache holds different content for Vary headers.

The items that the RAM Cache does not cache are:

- Private data specified by cache control headers
- By default, the RAM Cache does not cache HEAD, PUT, DELETE, TRACE, and CONNECT methods.

Understanding the RAM Cache mechanism

The default RAM Cache configuration caches only the HTTP GET methods. You can use the RAM Cache to cache both the GET and other methods, including non-HTTP methods, by specifying a URI in the URI Include list or writing an iRule. The cache uses the storage mechanism for cached content described in Table 1.1.

Action	Cached Content
Removed	All cookie headers are removed.
Modified	Hop by hop headers are modified when served. These headers include: Connection, Keep-Alive, and Transfer Encoding.
Added	A Date header is added that includes the current time on the BIG-IP system. An Age header is added that reflects the amount of time the item has been in the cache. Note that this setting is on by default in the profile. You can disable this setting by turning the Insert Age Header attribute off in the profile.
Unchanged	All other headers are stored as they are received.

Table 1.1 Cache storage mechanism

Clearing items from the cache

The RAM Cache removes the least frequently used items in the cache. This prevents stale items from taking up room in the cache when newer items are selected for caching. The cache also uses a scoring system to remove stale items after a period of time. When a cached item reaches its age limit, it is expired from the cache. You can use the HTTP profile attributes to control the size of each cache instance and how quickly items are expired from the cache. For more information about these attributes, see the following section, *Configuring the RAM Cache*.

Configuring the RAM Cache

To configure the RAM Cache, you need to configure the HTTP profile. These attributes provide the ability to turn on the cache and fine tune it for a specific implementation. In addition to configuring the RAM Cache objects in the HTTP profile, you may want to configure `bigdb` variables or `iRules`. This section includes information on understanding the RAM Cache attributes in the HTTP profile, and procedures for:

- Configuring the RAM Cache attributes in the Configuration utility
- Using **bigpipe** utility to configure the RAM Cache
- Understanding additional cache control commands

Understanding the RAM Cache attributes in the HTTP profile

The default items stored by the cache are HTTP GET responses. However, you can specify URIs in the URI list if you want to cache POST and GET methods for a particular URI.

The RAM Cache attributes are available in the HTTP profile. Table 1.2 lists and describes the RAM Cache settings for the HTTP profile type.

Setting	Description	Default Value
RAM cache	This setting specifies whether the RAM Cache feature is Enabled or Disabled.	Disabled
Maximum Cache Size	This setting specifies the maximum size for the RAM Cache. When the cache reaches the maximum size, the system starts removing the oldest entries.	100 megabytes
Maximum Entries	This setting specifies the maximum number of entries that can be in the RAM Cache. The default is 0 , which means that the system does not limit the maximum entries.	0
Maximum Age	This setting specifies how long the system considers the cached content to be valid.	3600 seconds
Minimum Object Size	This setting specifies the smallest object that the system considers eligible for caching.	500 bytes
Maximum Object Size	This setting specifies the largest object that the system considers eligible for caching.	50000 bytes

Table 1.2 RAM Cache settings in the HTTP profile

Setting	Description	Default Value
URI Pinning	This setting specifies whether the system retains or excludes certain URIs in the RAM Cache. The pinning process forces the system either to cache URIs that typically are ineligible for caching, or to not cache URIs that typically are eligible for caching.	Not Configured
URI List	<p>This setting specifies the Uniform Resource Identifiers (URIs) that the system either includes in or excludes from caching.</p> <p>Include List Lists the URIs that are typically ineligible for caching, but the system caches them. When you add URIs to the include list, the GET methods are cached for the URI and other methods, including non-HTTP methods.</p> <p>Exclude List Lists the URIs that are typically eligible for caching, but the system does not cache them.</p>	Not Configured
Ignore Headers	<p>This setting specifies how the system processes client-side Cache-Control headers when RAM caching is enabled.</p> <p>All Specifies that the system disregards all Cache-Control headers.</p> <p>Cache-Control:max-age Specifies that the system ignores only the Cache-Control:max-age header.</p> <p>None Specifies that the system honors all Cache-Control headers.</p>	All
Insert Age Header	This setting specifies, when enabled, that the system inserts Date and Age headers in the cached entry. The Date header contains the current date and time on the BIG-IP system. The Age header contains the length of time the content has been in the cache.	Enabled
Aging Rate	This setting specifies how quickly the system ages a cache entry. The aging rate ranges from 0 (slowest aging) to 10 (fastest aging).	9

Table 1.2 RAM Cache settings in the HTTP profile

To edit or create an HTTP profile in the Configuration utility

1. On the Main tab, expand **Local Traffic** and click **Profiles**.
The Profiles screen opens.
2. From the Services menu, choose HTTP.
This displays a list of any existing HTTP profiles.
3. You can edit an existing profile, or create a new one:
 - **Edit an existing profile**
In the Name column, click the name of the profile that you want to view.
 - **Create a new profile**
To create a new profile, click the **Create** button.
4. Configure the RAM Cache by configuring the appropriate HTTP profile attributes. For detailed information about each attribute, refer to Table 1.2, *RAM Cache settings in the HTTP profile*, or to view the information in the Configuration utility, click the Help tab.

To create a RAM Cache configuration from the command line

You can use the **bigpipe** command line utility to configure the RAM Cache attributes in the HTTP profile. For details about the RAM Cache syntax, see the HTTP man page. To access this man page, type the following command from the command prompt:

```
man profile_http
```

Understanding additional cache control commands

In addition to the RAM cache profile settings, you can view or remove items from the cache using **bigpipe** commands. These commands provide the ability to retrieve information about the contents of the cache and to evict items from the cache.

You can add the ability to get **<n>** hottest items in the cache, where **<n>** is the number of hot items you want to display.

```
b profile http ramcache max <n> dump
```

If you want to retrieve detailed information about the items in the cache, such as hits, age, and size on each cached entry, type the following command:

```
b profile http ramcache dump
```

To remove a particular cached entry, you can specify the string of the URI you want to remove using the following command:

```
b profile http ramcache uri <string> reset
```

You can purge all the cached items for a profile using the following command:

```
b profile http <name> ramcache reset
```

If you want to purge all items from all cache profiles, type the following command:

```
b profile http all ramcache reset
```

Using iRules with the RAM Cache

You can use specific rule events and commands with the RAM Cache. This provides the ability to disable or enable whether or not an entry should be cached. A rule takes precedence over a cache control header or other default configuration settings. Table 1.3 describes the specific RAM Cache commands.

Rule Command	Description
CACHE_REQUEST	Triggered when the system receives a request for a cached object.
CACHE_RESPONSE	Triggered just before the system sends the cache response.

Table 1.3 Rule commands for RAM Cache

In addition to the rule commands CACHE_REQUEST and CACHE_RESPONSE, there are several rules you can use with the RAM Cache feature. These rules are described in Table 1.4.

Rule	Description
CACHE::disable	Disables caching for this request. This rule can be called from HTTP_RESPONSE and HTTP_REQUEST.
CACHE::enable	Forces the document to be cached. This rule can be called from HTTP_RESPONSE and HTTP_REQUEST.
CACHE::expire	Forces the document to be re-validated from the server.
CACHE::hits	Returns the document cache hits.
CACHE::age	Returns the age of documents in the cache in seconds.

Table 1.4 Rule events for use with RAM Cache

Rule	Description
CACHE::headers	Returns the HTTP headers of the cache response. This rule can only be called from the CACHE_RESPONSE event.
CACHE::payload	Returns the HTTP payload of the cache response. This rule can only be called from the CACHE_RESPONSE event.

Table 1.4 Rule events for use with RAM Cache

An example RAM Cache rule

The rule in Figure 1.1 is an example of a rule that caches the HTTP POST methods. You can create rules to cache POST methods and non-HTTP methods such as WebDAV methods.

```
class post_uris {
    "/cgi-bin/webbservice/stats"
    "/cgi-bin/webbservice/readonly"
}
rule cache_posts {
    when CACHE_REQUEST {
        if {[CACHE::hits] == 0} && ([HTTP::method] == "POST")
        && ([matchclass [HTTP::uri] equals $::post_uris]) {
            CACHE::enable
        }
    }
}
```

Figure 1.1 A rule example for caching POSTs

For more information about writing rules, refer to *Chapter 13, Writing Rules*, in the *Configuration Guide for Local Traffic Management*.

Configuring bigdb settings for RAM Cache

You can fine tune the RAM Cache implementation by changing settings in the bigdb database. We recommend that you change these settings only while under the direction of an F5 support representative. The following bigdb setting is available for the RAM Cache feature:

Ramcache.MaxMemoryPercent

This setting determines the percentage of system memory available to all RAM Cache instances. This means that the RAM Cache size is represented by this statement:

ramcacheprofile1 + ramcacheprofile2 + ramcacheprofile3 must not equal more than Total Memory times the Ramcache.MaxMemoryPercent divided by 100

◆ Note

Changes to this setting take effect only after restarting the system services, including the TMM.