

For best results, use the following resources to troubleshoot an integrated cache issue on a NetScaler appliance:

- The relevant trace files
- The ns.conf file
- The RFC 2616 document
- A copy of the object, if possible

In addition to the above resources, the following tools expedite troubleshooting:

- The iehhttpheaders or a similar utility
- The Wireshark application customized for the NetScaler trace files

Troubleshooting Integrated Caching issues

The following are effective steps to troubleshoot the objects that are not cached:

1. Verify that the Integrated Caching feature is enabled. Run the following command to verify the feature is enabled:

```
show ns feature
```

Output

```
Feature status:  
Web Logging: OFF  
Surge Protection: OFF  
Load Balancing: ON  
Content Switching: ON  
Cache Redirection: OFF  
Sure Connect: OFF  
Compression Control: OFF  
Priority Queuing: OFF  
SSL Offloading: OFF  
Global Server Load Balancing: OFF  
Http DoS Protection: OFF  
Dynamic Routing: OFF  
Content Filtering: OFF  
Integrated Caching: ON  
SSL VPN: OFF  
OSPF Routing: OFF  
RIP Routing: OFF  
BGP Routing: OFF  
Done
```

The entry in bold (for reference) in the above output indicates that the integrated caching feature is enabled. If the feature is not enabled, run the following command to enable it:

```
enable ns feature IC
```

2. Make sure that sufficient memory is available on the NetScaler appliance. Depending on the size of the object to be cached, memory available to store the cacheable object might be insufficient. You can set the memory limit for the integrated cache either globally or for individual content groups.

```
show cache parameter
```

Output

```
Integrated cache global configuration:  
Memory usage limit: 256 MBytes Via header  
NS-CACHE-6.1: 101  
Verify cached object using: HOSTNAME_AND_IP  
Max POST body size to accumulate: 2768  
Current outstanding prefetches: 0  
Max outstanding prefetches: 4294967294  
Treat NOCACHE policies as BYPASS policies: YES  
Done
```

The entry highlighted in bold (for reference) in the preceding output indicates the amount of memory allocated to the integrated cache globally. Run the following command to verify the memory allocated to an individual content group:

```
show cache contentGroup <Content_Group_Name>
```

Output

```
show cache contentgroup content1  
  
Name: content1  
Heuristic expiry time parameter: 10 percent  
Weak relative expiry time - Positive responses: 3600 secs  
Weak relative expiry time - Negative responses: 600 secs  
Hit parameters: NONE  
Invalidation Parameters: NONE  
Invalidation restricted to host: NO  
Ignore parameter value case: NO  
Match Request Cookies: NO  
Poll Every Time: NO  
Ignore reload request: YES  
Remove Response Cookies: YES  
Prefetch: YES
```

```
Prefetch period: heuristic
Current outstanding prefetches: 0
Max outstanding prefetches: 4294967294
Flashcache: NO
Expire at last byte: NO
Insert Via header: YES
Insert Age header: YES
Insert ETag header: YES
Cache-control header: NONE
Quick abort size: 4194303 KBytes (MAXIMUM)
Minimum Response Size: 0 KBytes
Maximum Response Size: 80 KBytes
Memory usage: 0 Bytes
Memory usage limit: 64 MBytes
Ignore caching headers in request: YES
Non-304 hits: 0
304 hits: 0
Cached objects: 0
Number of times expired/flushed: 1
MinHits configured: 0
Always evaluate policies: NO
Pinned: NO
Done
```

The entry highlighted in boldface (for reference) in the above output indicates the amount of memory allocated to the content group. Verify that cacheable object is small enough to be stored within the configured memory limits.

Complete the following procedure to make space for the object to be cached:

- Run the following command to flush the cache for the content group to which the object belongs:

```
flush cache contentGroup <Content_Group_Name>
```

- Verify that the object is cached. If the object is cached successfully, increase the memory allocated for the content group. Otherwise, run the following command to flush the cache globally:

```
flush cache contentGroup ALL
```

- Verify that the object is cached. If the object is cached successfully, consider increasing the global memory limit. If the object is still not cached, something else is causing the failure to cache the object.

The memory allocated to the integrated cache depends on the NetScaler appliance model. You can allocate approximately half the available memory to the integrated cache. Similarly, the maximum amount of memory you can allocate for a content group cannot be more than the memory allocated for global cache.

To increase the global memory limit for the integrated cache, run the following command:

```
set cache parameter -memLimit <Integer>
```

To increase the memory limit for a content group, run the following command:

```
set cache contentgroup <contentgroup name> -memLimit <Integer>
```

3. Verify that the cache policy is bound to an appropriate bind point, an appropriate priority is set for the policy, and an `appropriateprecedeDefRuleswitch` is configured.

You must activate a caching policy by binding it globally. To verify that the policy is active, run the following command:

```
show cache global
```

Example

```
Name: red_pol State: ACTIVE Priority: 1
Rule: URL CONTAINS red
Action: NOCACHE
Precede default HTTP rules: YES
Hits : 10
Done
```

In the output, verify the following settings:

- **The policy is bound.** The output should contain all the active cache policies. If the cache policy for the object to be cached is not listed in the output, the policy is not yet bound. Run the following command to bind the policy globally:

```
bind cache global <Policy_Name> -priority <Integer> [-precedeDefRules YES|NO]
```

- **The policy is Active.** If the policy is bound, verify that the state of the policy is displayed as Active. The entry indicating that the policy in the preceding output is active is the first highlighted entry in the sample output of the `show cache global` command, above. The policy is active if it is bound globally and an appropriate priority is set. Otherwise, the status of the policy is showed as Passive.

- **An appropriate priority is assigned to the policy.** The first highlighted entry in the sample output above displays the priority of the policy. If the priority is not set, you can use the `bind` command to set the priority of the policy. Note that the higher the priority number, the lower the priority. The priority assigned to the policy enables the NetScaler appliance to determine the order in which the policy should be evaluated.

If evaluation of a particular policy fails, increase the priority of the policy so that it is evaluated before other policies. Caching policies, due to their high granularity, can be very complicated to configure. Therefore, two policies might be contradictory. As a result, only the higher-priority policy takes effect.

- **The precedeDefRules switch setting is correct.** The second highlighted entry in the sample output of the show cache global command, above, indicates the precedeDefRules switch setting. This setting enables the NetScaler appliance to determine whether the policy should be evaluated before the default built-in policies, which implement the standard HTTP caching behavior, such as basing caching decisions on HTTP header fields (for example, the If-Modified-Since and no-cache fields). You can set this switch when binding the policy.

For certain types of HTTP(S) transactions, you might have to make sure that the policy precedes default HTTP rules, to force objects to be cached. Especially if requests include header fields, such as If-Modified-Since, or responses contain the No-Cache header field, you might have to make sure that the cache policy overrides the default in order for objects from these transactions to be cached. Force the policy to override default HTTP rules by rebinding the cache policy with the -precedeDefRules YES switch.

1. Verify the size of the object to be cached. You can configure a content group with minimum, which by default is 0 KB, and maximum, which by default is 80 KB, response sizes for the objects to be cached. The object does not get cached if its size is not within the configured range. Additionally, verify that the cache expiry times are set to an appropriate value. For example, check for a very small time limit, such as one second.

Run the following command from the command line interface of the appliance to display the size limits and expiry times for a specific content group:

```
show cache contentGroup <Content_Group_Name>
```

Following is an example of this command's output:

```
show cache contentGroup content1
```

Example

```
Name: content1
Heuristic expiry time parameter: 10 percent
Weak relative expiry time - Positive responses: 3600 secs
Weak relative expiry time - Negative responses: 600 secs
Hit parameters: NONE
Invalidation Parameters: NONE
Invalidation restricted to host: NO
Ignore parameter value case: NO
Match Request Cookies: NO
Poll Every Time: NO
Ignore reload request: YES
Remove Response Cookies: YES
```

```
Prefetch: YES
Prefetch period: heuristic
Current outstanding prefetches: 0
Max outstanding prefetches: 4294967294
Flashcache: NO
Expire at last byte: NO
Insert Via header: YES
Insert Age header: YES
Insert ETag header: YES
Cache-control header: NONE
Quick abort size: 4194303 KBytes (MAXIMUM)
Minimum Response Size: 0 KBytes
Maximum Response Size: 80 KBytes
Memory usage: 0 Bytes
Memory usage limit: UNLIMITED
Ignore caching headers in request: YES
Non-304 hits: 0
304 hits: 0
Cached objects: 0
Number of times expired/flushed: 0
MinHits configured: 0
Always evaluate policies: NO
Pinned: NO
Done
```

In addition to the above steps for troubleshooting integrated caching issues, you can consider using the following troubleshooting techniques:

Depending on the configuration of a policy, there are virtually an unlimited number of reasons for the policy not getting evaluated. If you have completed the preceding steps to troubleshoot the issue, consider completing the following procedure to troubleshoot the issue further:

1. Flush the cache.
2. Verify the value of the hit parameter for the policy by running the following command:

```
show cache global
```

Example configuration:

```
Name: home_pol_1 State: ACTIVE Priority: 99
Rule: URL CONTAINS home
Action: NOCACHE
Precede default HTTP rules: NO
Hits: 2
```

3. Send an HTTP request for the related object from a Web browser.
4. Run the show cache global command again and verify that the value for the hit parameter has incremented.

Depending on the policy receiving hits or not, you can determine whether the issue is due to the policy have not been configured correctly or to a more global cache setting.