

Contents

- [Cache Monitor](#)
 - [Cache Monitor](#)
 - [Cache Monitor Reports](#)
 - [Cache Monitor Methods](#)
 - [General](#)
 - [getCachedClasses\(\)](#)
 - [getCacheSettings\(className\)](#)
 - [Size](#)
 - [getEstimatedSize\(className\)](#)
 - [getCalculatedSize\(className\)](#)
 - [getTotalEstimatedSize\(\)](#)
 - [getTotalCalculatedSize\(\)](#)
 - [Hits](#)
 - [getHits\(className\)](#)
 - [getTotalHits\(\)](#)
 - [Misses](#)
 - [getMisses\(className\)](#)
 - [getTotalMisses\(\)](#)
 - [Hit/Miss Ratios](#)
 - [getHitMissRatio\(className\)](#)
 - [getTotalHitMissRatio\(\)](#)
 - [Evictions](#)
 - [getEvictions\(className\)](#)
 - [getTotalEvictions\(className\)](#)
 - [Resets](#)
 - [resetHitsAndMisses\(\)](#)
 - [resetEvictions\(\)](#)

Cache Monitor

When developing with Transfer, it is generally advisable to have some cache expiry configured, in order to get the optimal balance between memory usage, and performance gain from caching TransferObjects.

The Cache Monitor is a easy way to view what is going on inside the Transfer cache at any given moment.

Cache Monitor

The [CacheMonitor.cfc](#) is available from [transfer.com.Transfer](#), via the `getCacheMonitor()` method.

For example:

```
<cfset monitor = application.transferFactory.getTransfer().getCacheMonitor() />
```

Cache Monitor Reports

Transfer comes bundled with two reports, a basic report, and a detailed report, which can be quickly and easily used to display the state of the Transfer cache.

For more details, see the [Cache Report Custom Tag](#)

Cache Monitor Methods

General

getCachedClasses()

[\(API\)](#)

This returns all the cached classes that currently reside in Transfer.

getCacheSettings(className)

[\(API\)](#)

Gets the settings for the particular class, with a struct with the following keys:

- scope
- maxobjects
- maxminutespersisted
- accessedminutestimeout

Size

These methods count how many objects are stored in a specific set inside the cache.

getEstimatedSize(className)

[\(API\)](#)

A fast lookup of how many items in the cache, simply by checking its size, which may not be exactly accurate.

getCalculatedSize(className)

[\(API\)](#)

A slow look at how many items are in cache, where a copy of the cache is taken, and inspected item by item.

getTotalEstimatedSize()

[\(API\)](#)

The estimated size total for all classes in the cache.

getTotalCalculatedSize()

[\(API\)](#)

The calculated size total for all classes in the cache.

Hits

Hits occur when an object is requested from the cache, and the cache already has available to return.

getHits(className)

[\(API\)](#)

Returns the number of hits for that class

getTotalHits()

[\(API\)](#)

Returns the total number of hits, for all cached classes.

Misses

Misses occur when an object is requested from the cache, but it does not exist.

getMisses(className)

[\(API\)](#)

Returns the number of misses for that class

getTotalMisses()

[\(API\)](#)

Returns the total number of misses, for all cached classes.

Hit/Miss Ratios

A ratio greater than 1 between hits and misses is ideal, as it means that more often than not, an object is being retrieved from the cache, rather than having to go to the database.

getHitMissRatio(className)

[\(API\)](#)

Returns the ratio of hits vs misses. Values above 1 mean that more hits are occurring more than misses.

getTotalHitMissRatio()

[\(API\)](#)

Returns the ratio of total hits vs total misses. Values above 1 mean that more hits are occurring than misses.

Evictions

Evictions are when an object is dropped from the cache, either by the JVM, or by expiry configuration settings.

getEvictions(className)

[\(API\)](#)

Returns the total number of cache evictions for this class

getTotalEvictions(className)

[\(API\)](#)

Returns the the total number of cache evictions.

Resets

In some instances, it may be useful to reset some of the cache tracking data back to 0. This is probably most useful when running taking snapshots of the cache at scheduled time intervals.

resetHitsAndMisses()

[\(API\)](#)

Resets the Hit and Miss counters back to 0

resetEvictions()

[\(API\)](#)

Resets eviction counters back to 0

 Categories:

- [Cache](#)