Nimbusキャッシュ仕様書

2016年8月8日

目次

[1. 概要 1](#_Toc459362962)

[2. インタフェース設計 3](#_Toc459362963)

[2.1. Cache 3](#_Toc459362964)

[2.2. CacheMap 5](#_Toc459362965)

[2.3. OverflowController 7](#_Toc459362966)

[2.4. OverflowValidator 8](#_Toc459362967)

[2.5. OverflowAlgorithm 9](#_Toc459362968)

[2.6. OverflowAction 10](#_Toc459362969)

[3. 実装設計 11](#_Toc459362970)

[3.1. Cache 11](#_Toc459362971)

[3.1.1. jp.ossc.nimbus.service.cache.MemoryCacheService 11](#_Toc459362972)

[3.1.2. jp.ossc.nimbus.service.cache.FileCacheService 12](#_Toc459362973)

[3.1.3. jp.ossc.nimbus.service.cache.SerializedMemoryCacheService 14](#_Toc459362974)

[3.2. CacheMap 16](#_Toc459362975)

[3.2.1. jp.ossc.nimbus.service.cache.MemoryCacheMapService 16](#_Toc459362976)

[3.2.2. jp.ossc.nimbus.service.cache.FileCacheMapService 17](#_Toc459362977)

[3.2.3. jp.ossc.nimbus.service.cache.SerializedMemoryCacheMapService 19](#_Toc459362978)

[3.3. OverflowController 21](#_Toc459362979)

[3.3.1. jp.ossc.nimbus.service.cache.DefaultOverflowControllerService 21](#_Toc459362980)

[3.4. OverflowValidator 23](#_Toc459362981)

[3.4.1. jp.ossc.nimbus.service.cache.CacheSizeOverflowValidatorService 23](#_Toc459362982)

[3.4.2. jp.ossc.nimbus.service.cache.MemorySizeOverflowValidatorService 23](#_Toc459362983)

[3.4.3. jp.ossc.nimbus.service.cache.CalculateMemorySizeOverflowValidatorService 24](#_Toc459362984)

[3.4.4. jp.ossc.nimbus.service.cache.TimeExpierOverflowValidatorService 25](#_Toc459362985)

[3.5. OverflowAlgorithm 27](#_Toc459362986)

[3.5.1. jp.ossc.nimbus.service.cache.FIFOOverflowAlgorithmService 27](#_Toc459362987)

[3.5.2. jp.ossc.nimbus.service.cache.LFUOverflowAlgorithmService 27](#_Toc459362988)

[3.5.3. jp.ossc.nimbus.service.cache.LIFOOverflowAlgorithmService 28](#_Toc459362989)

[3.5.4. jp.ossc.nimbus.service.cache.LRUOverflowAlgorithmService 28](#_Toc459362990)

[3.6. OverflowAction 29](#_Toc459362991)

[3.6.1. jp.ossc.nimbus.service.cache.RemoveOverflowActionService 29](#_Toc459362992)

[3.6.2. jp.ossc.nimbus.service.cache.StoreCacheOverflowActionService 29](#_Toc459362993)

[3.6.3. jp.ossc.nimbus.service.cache.SoftReferenceOverflowActionService 30](#_Toc459362994)

[3.6.4. jp.ossc.nimbus.service.cache.ContextSaveOverflowActionService 31](#_Toc459362995)

# 概要

キャッシュとは、データをメモリ中にキャッシュし、データ参照を高速に行う機能である。但し、メモリ中にキャッシュされたデータによって、メモリが圧迫される事が考えられるため、一定条件を満たすと、キャッシュから追い出す「あふれ制御」を行う事で、キャッシュデータによるメモリ使用量を制限する事ができる。

この「あふれ制御」は、どのような条件を満たすと、いくつのデータをあふれさせるかを決める「あふれ検証」、どのデータをあふれさせるかを決める「あふれアルゴリズム」、どのようにあふれさせるかを決める「あふれ動作」の組み合わせで決まる。

あふれ

キャッシュ

あふれ制御

あふれ動作

あふれ検証

あふれアルゴリズム

キャッシュのインタフェースとしては、データのみをキャッシュするCacheインタフェースと、キーとデータを紐づけてキャッシュするCacheMapインタフェースを提供する。

その両方に、あふれ制御を行うOverflowControllerインタフェースの実装クラスを設定する事で、あふれを行う。

また、あふれ制御の、各要素である、あふれ検証はOverflowValidatorインタフェース、

あふれアルゴリズムはOverflowAlgorithmインタフェース、あふれ動作はOverflowActionインタフェースで実装し、OverflowControllerに設定する事で各動作を決定する。JavaDocは、以下を参照。

<http://nimbus.osdn.jp/reports/apidocs/jp/ossc/nimbus/service/cache/package-summary.html>

サンプルコードは、以下のzipファイル内のcacheを参照。

<https://osdn.jp/projects/nimbus/downloads/64236/nimbus-sample-1.2.3.zip>

# インタフェース設計

## Cache

データをキャッシュする機能を提供するインタフェースである。

基本的なAPIは、データの集合管理を行うコレクションAPIであるjava.util.Setと、ほぼ互換性のあるインタフェースである。異なるのは、データをキャッシュするメソッドadd(Object)を呼び出すと、戻り値として、CachedReferenceというキャッシュしたデータを引き出すための、キャッシュ参照が取得できるところである。

キャッシュしたデータは、あふれ制御により、その実体がどこかに退避される可能性があるため、それ自身の参照は失われる可能性がある。そのため、実体であるデータを参照して引き出すための、代わりとなる参照オブジェクトが必要で、それがCachedReferenceである。

control (CachedReference)

CachedReference

**Cache**

add(Object)

OverflowController

get()

以下に、ソースコードの例を示す。

import jp.ossc.nimbus.core.\*;

import jp.ossc.nimbus.service.cache.\*;

Cache cache = (Cache)ServiceManagerFactory.getServiceObject(“Cache”);

CachedReference ref = cache.add(“value”);

String value = (String)ref.get();

以下に、Cacheインタフェースの実装クラスの一覧を示す。

|  |  |  |
| --- | --- | --- |
| # | 実装クラス名 | 概要 |
| 1 | jp.ossc.nimbus.service.cache.MemoryCacheService | オブジェクトをメモリ中にキャッシュする実装クラスである。 |
| 2 | jp.ossc.nimbus.service.cache.FileCacheService | オブジェクトを直列化してファイルにキャッシュする実装クラスである。 |
| 3 | jp.ossc.nimbus.service.cache.SerializedMemoryCacheService | オブジェクトを直列化してメモリにキャッシュする実装クラスである。 |

## CacheMap

データをキーに紐づけてキャッシュする機能を提供するインタフェースである。

基本的なAPIは、キーと値のマッピングで集合管理を行うコレクションAPIであるjava.util.Mapと、互換性のあるインタフェースである。異なるのは、キャッシュ参照を取得するメソッドgetCachedReference(Object)で、戻り値として、KeyCachedReferenceというキャッシュしたデータを引き出すための、キャッシュ参照が取得できるところである。

キャッシュしたデータは、あふれ制御により、その実体がどこかに退避される可能性があるため、それ自身の参照は失われる可能性がある。そのため、実体であるデータを参照して引き出すための、代わりとなる参照オブジェクトが必要で、それがKeyCachedReferenceである。

put(key, value)

control (CachedReference)

**CacheMap**

OverflowController

getCachedReference(key)

KeyCachedReference

get()

import jp.ossc.nimbus.core.\*;

import jp.ossc.nimbus.service.cache.\*;

CacheMap cache = (CacheMap)ServiceManagerFactory.getServiceObject(“Cache”);

Object old = cache.put(“key”, “value”);

KeyCachedReference ref = cache.getCachedReference(“key”);

String value = (String)ref.get();

以下に、CacheMapインタフェースの実装クラスの一覧を示す。

|  |  |  |
| --- | --- | --- |
| # | 実装クラス名 | 概要 |
| 1 | jp.ossc.nimbus.service.cache.MemoryCacheMapService | オブジェクトをメモリ中にキャッシュする実装クラスである。 |
| 2 | jp.ossc.nimbus.service.cache.FileCacheMapService | オブジェクトを直列化してファイルにキャッシュする実装クラスである。 |
| 3 | jp.ossc.nimbus.service.cache.SerializedMemoryCacheMapService | オブジェクトを直列化してメモリにキャッシュする実装クラスである。 |

## OverflowController

キャッシュに設定する事で、キャッシュしたオブジェクトのあふれ制御を行う機能を提供するインタフェースである。

あふれ制御は、あふれ検証OverflowValidator、あふれアルゴリズムOverflowAlgorithm、あふれ動作OverflowActionの３つの機能要素から構成されており、これらの組み合わせで、あふれ制御の動作を決定する。

あふれ制御では、OverflowValidator#validate()を呼び出し、キャッシュしたデータを、いくつあふれさせるかを問い合わせる。それが、0以上の場合は、OverflowAlgorithm#overflow(int)を呼び出し、あふれさせるキャッシュ参照を特定する。特定したあふれ対象のキャッシュ参照を、OverflowAction#action(OverflowValidator, OverflowAlgorithm, CachedReference)に渡して、あふれさせる。

CacheMap

Cache

add()

put()

control()

KeyCachedReference

CachedReference

**OverflowController**

action()

overflow()

validate()

OverflowAction

OverflowAlgorithm

OverflowValidator

以下に、OverflowControllerインタフェースの実装クラスの一覧を示す。

|  |  |  |
| --- | --- | --- |
| # | 実装クラス名 | 概要 |
| 1 | jp.ossc.nimbus.service.cache.DefaultOverflowControllerService | デフォルト実装クラスである。 |

## OverflowValidator

あふれさせるキャッシュ参照の数を決定する機能を提供するインタフェースである。

OverflowControllerから、validate()が呼び出されると、キャッシュ参照をいくつあふれさせるかを返す。あふれさせる必要がない場合は、0を返す。

OverflowController

validate()

**OverflowValidator**

以下に、OverflowValidatorインタフェースの実装クラスの一覧を示す。

|  |  |  |
| --- | --- | --- |
| # | 実装クラス名 | 概要 |
| 1 | jp.ossc.nimbus.service.cache.CacheSizeOverflowValidatorService | キャッシュされている数が閾値を超えるとあふれさせる実装クラスである。 |
| 2 | jp.ossc.nimbus.service.cache.MemorySizeOverflowValidatorService | ヒープメモリの使用量が閾値を超えるとあふれさせる実装クラスである。 |
| 3 | jp.ossc.nimbus.service.cache.CalculateMemorySizeOverflowValidatorService | 理論的に計算したキャッシュデータによるメモリの使用量が閾値を超えるとあふれさせる実装クラスである。 |
| 4 | jp.ossc.nimbus.service.cache.TimeExpierOverflowValidatorService | キャッシュされてからの経過時間が閾値を超えるとあふれさせる実装クラスである。 |

## OverflowAlgorithm

あふれさせるキャッシュ参照を決定する機能を提供するインタフェースである。

OverflowControllerから、overflow()が呼び出されると、あふれさせるキャッシュ参照を返す。

OverflowController

CachedReference

overflow()

**OverflowAlgoritm**

以下に、OverflowAlgorithmインタフェースの実装クラスの一覧を示す。

|  |  |  |
| --- | --- | --- |
| # | 実装クラス名 | 概要 |
| 1 | jp.ossc.nimbus.service.cache.FIFOOverflowAlgorithmService | First In First Outアルゴリズムで、あふれさせるキャッシュ参照を決定する実装クラスである。 |
| 2 | jp.ossc.nimbus.service.cache.LFUOverflowAlgorithmService | Least Frequency Usedアルゴリズムで、あふれさせるキャッシュ参照を決定する実装クラスである。 |
| 3 | jp.ossc.nimbus.service.cache.LIFOOverflowAlgorithmService | Last In First Outアルゴリズムで、あふれさせるキャッシュ参照を決定する実装クラスである。 |
| 4 | jp.ossc.nimbus.service.cache.LRUOverflowAlgorithmService | Least Recently Usedアルゴリズムで、あふれさせるキャッシュ参照を決定する実装クラスである。 |

## OverflowAction

キャッシュ参照をあふれさせる動作を提供するインタフェースである。

OverflowControllerから、action()が呼び出されると、キャッシュ参照をあふれさせる。

OverflowController

action()

**OverflowAction**

以下に、OverflowActionインタフェースの実装クラスの一覧を示す。

|  |  |  |
| --- | --- | --- |
| # | 実装クラス名 | 概要 |
| 1 | jp.ossc.nimbus.service.cache.RemoveOverflowActionService | キャッシュ参照をキャッシュから削除する実装クラスである。 |
| 2 | jp.ossc.nimbus.service.cache.StoreCacheOverflowActionService | キャッシュ参照をメモリ以外にキャッシュする２次キャッシュに退避する実装クラスである。 |
| 3 | jp.ossc.nimbus.service.cache.SoftReferenceOverflowActionService | キャッシュ参照を強参照からソフト参照に変更し、ソフト参照がガベージされた場合はメモリ以外にキャッシュする２次キャッシュに退避する実装クラスである。 |
| 4 | jp.ossc.nimbus.service.cache.ContextSaveOverflowActionService | Contextの永続化機能を使って、あふれ対象のキーを保存し、キャッシュから削除する実装クラスである。 |

# 実装設計

インタフェース設計で示した各インタフェースのうち、機能性のあるインタフェースの実装クラスの機能を説明する。

また、インタフェース設計に現れなかった、クラスの機能も説明する。

## Cache

### jp.ossc.nimbus.service.cache.MemoryCacheService

データをメモリ中にキャッシュするCacheインタフェースの実装クラスである。また、PersistableCacheインタフェースも実装しており、メモリ以外にキャッシュするCacheインタフェース実装クラスを設定する事で、データを永続化する事ができる。

save()

load()

control (CachedReference)

**MemoryCacheService**

add(Object)

OverflowController

Cache

DefaultCachedReference

get()

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | OverflowControllerServiceNames | ServiceName[] | あふれ制御を行うOverflowControllerインタフェースを実装したサービスのサービス名の配列を設定する。 |
| 2 | ClearOnStop | boolean | サービスの停止時にキャッシュをクリアするかどうかを設定する。  デフォルトは、false。 |
| 3 | ClearOnDestroy | boolean | サービスの破棄時にキャッシュをクリアするかどうかを設定する。  デフォルトは、true。 |
| 4 | PersistCacheServiceName | ServiceName | 永続化先となるCacheサービスのサービス名を設定する。 |
| 5 | LoadOnStart | boolean | サービスの開始時に、永続化先となるCacheからキャッシュエントリを取得して、このキャッシュに読み込みを行うかどうかを設定する。  デフォルトは、trueで読み込みを行う。 |
| 6 | SaveOnStop | boolean | サービスの停止時に、永続化先となるCacheにキャッシュエントリを保存するかどうかを設定する。  デフォルトは、trueで保存する。 |

### jp.ossc.nimbus.service.cache.FileCacheService

データを直列化してファイルとしてキャッシュするCacheインタフェースの実装クラスである。

control (CachedReference)

**FileCacheService**

add(Object)

OverflowController

FileCachedReference

ファイル

get()

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | OverflowControllerServiceNames | ServiceName[] | あふれ制御を行うOverflowControllerインタフェースを実装したサービスのサービス名の配列を設定する。 |
| 2 | ClearOnStop | boolean | サービスの停止時にキャッシュをクリアするかどうかを設定する。  デフォルトは、false。 |
| 3 | ClearOnDestroy | boolean | サービスの破棄時にキャッシュをクリアするかどうかを設定する。  デフォルトは、true。 |
| 4 | OutputDirectory | String | キャッシュしたオブジェクトを直列化してファイルとして出力する際の出力先ディレクトリを設定する。  出力先ディレクトリが指定されていない場合は、JVMのテンポラリディレクトリを使用する。 |
| 5 | OutputPrefix | String | キャッシュしたオブジェクトをシリアライズしてファイルとして出力する際のファイル名のプレフィクスを設定する。  この出力ファイルプレフィクスが指定されていない場合は、キャッシュするオブジェクトのtoString()が使用される。 |
| 6 | OutputSuffix | String | キャッシュしたオブジェクトを直列化してファイルとして出力する際のファイル名のサフィックスを設定する。  この出力ファイルサフィックスが指定されていない場合は、".obj"が使用される。 |
| 7 | LoadOnStart | boolean | キャッシュしたオブジェクトを直列化してファイルとして出力したキャッシュファイルをサービスの開始時にロードするかどうかを設定する。  デフォルトは、falseで、サービスの開始時にロードしない。 |
| 8 | DeleteOnExitWithJVM | boolean | キャッシュしたオブジェクトを直列化してファイルとして出力したキャッシュファイルをJVMの終了時に削除するかどうかを設定する。  デフォルトは、trueで、JVM終了時に削除する。 |
| 9 | CheckFileOnLoad | boolean | キャッシュファイルのロード時に、ファイルがロードできる事を検証するかどうかを設定する。  デフォルトは、falseで、検証しない。 |
| 10 | DeleteOnCheckFileError | boolean | ファイルがロードできる事を検証した結果失敗した場合に、該当ファイルを削除するかどうかを設定する。  デフォルトは、falseで、削除しない。 |
| 11 | ExternalizerServiceName | ServiceName | ファイルに直列化する際に直列化を行うjp.ossc.nimbus.service.io.Externalizerサービスのサービス名を設定する。 |

### jp.ossc.nimbus.service.cache.SerializedMemoryCacheService

データを直列化してバイト配列としてメモリ中にキャッシュするCacheインタフェースの実装クラスである。また、PersistableCacheインタフェースも実装しており、メモリ以外にキャッシュするCacheインタフェース実装クラスを設定する事で、データを永続化する事ができる。

save()

load()

control (CachedReference)

**SerializedMemoryCacheService**

add(Object)

OverflowController

Cache

SerializeCachedReference

get()

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | OverflowControllerServiceNames | ServiceName[] | あふれ制御を行うOverflowControllerインタフェースを実装したサービスのサービス名の配列を設定する。 |
| 2 | ClearOnStop | boolean | サービスの停止時にキャッシュをクリアするかどうかを設定する。  デフォルトは、false。 |
| 3 | ClearOnDestroy | boolean | サービスの破棄時にキャッシュをクリアするかどうかを設定する。  デフォルトは、true。 |
| 4 | PersistCacheServiceName | ServiceName | 永続化先となるCacheサービスのサービス名を設定する。 |
| 5 | LoadOnStart | boolean | サービスの開始時に、永続化先となるCacheからキャッシュエントリを取得して、このキャッシュに読み込みを行うかどうかを設定する。  デフォルトは、trueで読み込みを行う。 |
| 6 | SaveOnStop | boolean | サービスの停止時に、永続化先となるCacheにキャッシュエントリを保存するかどうかを設定する。  デフォルトは、trueで保存する。 |
| 7 | ExternalizerServiceName | ServiceName | 直列化する際に直列化を行うjp.ossc.nimbus.service.io.Externalizerサービスのサービス名を設定する。 |

## CacheMap

### jp.ossc.nimbus.service.cache.MemoryCacheMapService

データをキーと紐づけてメモリ中にキャッシュするCacheMapインタフェースの実装クラスである。また、PersistableCacheインタフェースも実装しており、メモリ以外にキャッシュするCacheMapインタフェース実装クラスを設定する事で、データを永続化する事ができる。

save()

load()

put(key, value)

control (CachedReference)

**MemoryCacheMapService**

OverflowController

CacheMap

getCachedReference(key)

DefaultKeyCachedReference

get()

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | OverflowControllerServiceNames | ServiceName[] | あふれ制御を行うOverflowControllerインタフェースを実装したサービスのサービス名の配列を設定する。 |
| 2 | ClearOnStop | boolean | サービスの停止時にキャッシュをクリアするかどうかを設定する。  デフォルトは、false。 |
| 3 | ClearOnDestroy | boolean | サービスの破棄時にキャッシュをクリアするかどうかを設定する。  デフォルトは、true。 |
| 4 | PersistCacheMapServiceName | ServiceName | 永続化先となるCacheMapサービスのサービス名を設定する。 |
| 5 | LoadOnStart | boolean | サービスの開始時に、永続化先となるCacheからキャッシュエントリを取得して、このキャッシュに読み込みを行うかどうかを設定する。  デフォルトは、trueで読み込みを行う。 |
| 6 | SaveOnStop | boolean | サービスの停止時に、永続化先となるCacheにキャッシュエントリを保存するかどうかを設定する。  デフォルトは、trueで保存する。 |

### jp.ossc.nimbus.service.cache.FileCacheMapService

データとキーを紐づけて、データを直列化してファイルとしてキャッシュするCacheMapインタフェースの実装クラスである。

put(key, value)

control (CachedReference)

**FileCacheMapService**

OverflowController

getCachedReference(key)

FileKeyCachedReference

ファイル

get()

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | OverflowControllerServiceNames | ServiceName[] | あふれ制御を行うOverflowControllerインタフェースを実装したサービスのサービス名の配列を設定する。 |
| 2 | ClearOnStop | boolean | サービスの停止時にキャッシュをクリアするかどうかを設定する。  デフォルトは、false。 |
| 3 | ClearOnDestroy | boolean | サービスの破棄時にキャッシュをクリアするかどうかを設定する。  デフォルトは、true。 |
| 4 | OutputDirectory | String | キャッシュしたオブジェクトを直列化してファイルとして出力する際の出力先ディレクトリを設定する。  出力先ディレクトリが指定されていない場合は、JVMのテンポラリディレクトリを使用する。 |
| 5 | OutputPrefix | String | キャッシュしたオブジェクトをシリアライズしてファイルとして出力する際のファイル名のプレフィクスを設定する。  この出力ファイルプレフィクスが指定されていない場合は、キャッシュするオブジェクトのtoString()が使用される。 |
| 6 | OutputSuffix | String | キャッシュしたオブジェクトを直列化してファイルとして出力する際のファイル名のサフィックスを設定する。  この出力ファイルサフィックスが指定されていない場合は、".obj"が使用される。 |
| 7 | LoadOnStart | boolean | キャッシュしたオブジェクトを直列化してファイルとして出力したキャッシュファイルをサービスの開始時にロードするかどうかを設定する。  デフォルトは、falseで、サービスの開始時にロードしない。 |
| 8 | DeleteOnExitWithJVM | boolean | キャッシュしたオブジェクトを直列化してファイルとして出力したキャッシュファイルをJVMの終了時に削除するかどうかを設定する。  デフォルトは、trueで、JVM終了時に削除する。 |
| 9 | CheckFileOnLoad | boolean | キャッシュファイルのロード時に、ファイルがロードできる事を検証するかどうかを設定する。  デフォルトは、falseで、検証しない。 |
| 10 | DeleteOnCheckFileError | boolean | ファイルがロードできる事を検証した結果失敗した場合に、該当ファイルを削除するかどうかを設定する。  デフォルトは、falseで、削除しない。 |
| 11 | ExternalizerServiceName | ServiceName | ファイルに直列化する際に直列化を行うjp.ossc.nimbus.service.io.Externalizerサービスのサービス名を設定する。 |

### jp.ossc.nimbus.service.cache.SerializedMemoryCacheMapService

キーとデータを紐づけて、データを直列化してバイト配列としてメモリ中にキャッシュするCacheMapインタフェースの実装クラスである。また、PersistableCacheインタフェースも実装しており、メモリ以外にキャッシュするCacheMapインタフェース実装クラスを設定する事で、データを永続化する事ができる。

save()

load()

put(key, value)

control (CachedReference)

OverflowController

**SerializedMemoryCacheMapService**

CacheMap

getCachedReference(key)

DefaultKeyCachedReference

get()

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | OverflowControllerServiceNames | ServiceName[] | あふれ制御を行うOverflowControllerインタフェースを実装したサービスのサービス名の配列を設定する。 |
| 2 | ClearOnStop | boolean | サービスの停止時にキャッシュをクリアするかどうかを設定する。  デフォルトは、false。 |
| 3 | ClearOnDestroy | boolean | サービスの破棄時にキャッシュをクリアするかどうかを設定する。  デフォルトは、true。 |
| 4 | PersistCacheServiceName | ServiceName | 永続化先となるCacheサービスのサービス名を設定する。 |
| 5 | LoadOnStart | boolean | サービスの開始時に、永続化先となるCacheからキャッシュエントリを取得して、このキャッシュに読み込みを行うかどうかを設定する。  デフォルトは、trueで読み込みを行う。 |
| 6 | SaveOnStop | boolean | サービスの停止時に、永続化先となるCacheにキャッシュエントリを保存するかどうかを設定する。  デフォルトは、trueで保存する。 |
| 7 | ExternalizerServiceName | ServiceName | 直列化する際に直列化を行うjp.ossc.nimbus.service.io.Externalizerサービスのサービス名を設定する。 |

## OverflowController

### jp.ossc.nimbus.service.cache.DefaultOverflowControllerService

OverflowControllerインタフェースのデフォルト実装クラスである。

CacheMap

Cache

add()

put()

control()

KeyCachedReference

CachedReference

**DefaultOverflowControllerService**

action()

overflow()

validate()

OverflowAction

OverflowAlgorithm

OverflowValidator

Cacheや、CacheMapサービスに設定する事で、キャッシュされたデータのあふれ制御を行う。あふれ制御のタイミングは、設定によって、以下のタイミングで行える。

1. キャッシュしたタイミングで同期的にあふれ制御する。
2. キャッシュしたタイミングで非同期にあふれ制御する。
3. 定期的にあふれ制御する。

但し、（３）は、（１）、（２）と併用する事が可能である。

あふれ制御の要素である「あふれ検証」、「あふれアルゴリズム」、「あふれ動作」は、それぞれOverflowValidator、OverflowAlgorithm、OverflowActionに委譲する。

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | OverflowValidatorServiceName | ServiceName | あふれ検証を行うOverflowValidatorサービスのサービス名を設定する。  設定しない場合は、あふれ制御が行われない。 |
| 2 | OverflowAlgorithmServiceName | ServiceName | あふれ検証結果に従ってあふれるキャッシュオブジェクトを決定するOverflowAlgorithmサービスのサービス名を設定する。 |
| 3 | OverflowActionServiceName | ServiceName | あふれアルゴリズムによって決定されたあふれキャッシュオブジェクトをあふれさせるOverflowActionサービスのサービス名を設定する。  設定しない場合には、RemoveOverflowActionServiceが使用される。 |
| 4 | QueueServiceName | ServiceName | あふれ制御の要求を別スレッドで処理するために、一旦キューに溜めるためのQueueサービスのサービス名を設定する。  設定しない場合には、同期的にあふれ制御が行われる。 |
| 5 | PeriodicOverflowIntervalTime | long | 定期的にあふれ制御を行う時間間隔[ms]を設定する。  デフォルトは0で、定期的なあふれ制御は行わない。 |
| 6 | OverflowByAdding | boolean | キャッシュ参照が追加されるたびにあふれ制御を行うかどうかを設定する。  デフォルトは、trueで、追加のたびにあふれ制御を行う。 |
| 7 | ValidateByOverflow | boolean | あふれ制御を行うたびにあふれ検証を行うかどうかを設定する。  デフォルトは、trueで、あふれ制御を行うたびにあふれ検証を行う。 |
| 8 | OverflowNewAdding | boolean | 新規追加されるキャッシュ参照をあふれ対象に加えるかどうかを設定する。  デフォルトは、falseで、新規追加されるキャッシュ参照はあふれ対象に加えない。 |

## OverflowValidator

### jp.ossc.nimbus.service.cache.CacheSizeOverflowValidatorService

キャッシュされたデータの件数が、閾値を超えるとあふれさせるOverflowValidatorインタフェースの実装クラスである。

OverflowController

validate()

**CacheSizeOverflowValidatorService**

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | MaxSize | int | 保持するキャッシュの最大数を設定する。  有効な値の範囲は、0以上。デフォルトは0で、溢れない。 |
| 2 | OverflowThreshold | int | あふれ閾値を設定する。  あふれが発生する場合に、この閾値まであふれさせる。  デフォルトは0で、最大数を越えた分だけあふれさせる。 |

### jp.ossc.nimbus.service.cache.MemorySizeOverflowValidatorService

ヒープメモリの使用量が、閾値を超えるとあふれさせるOverflowValidatorインタフェースの実装クラスである。

OverflowController

validate()

**MemorySizeOverflowValidatorService**

高負荷ヒープメモリと高負荷ヒープメモリの２つの閾値を持ち、以下の計算式で、あふれ件数を計算する。

キャッシュされている件数×（使用ヒープメモリ‐高負荷ヒープメモリ）÷（最大ヒープメモリ‐高負荷ヒープメモリ）

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | MaxHeapMemorySize | String | ヒープメモリの最大サイズを設定する。  有効な値の範囲は、0以上。デフォルトはRuntime#maxMemory()で取得できる値。そのメソッドがサポートされていない場合は、64MByte。  単位を指定しない場合は、バイト単位。単位を指定する場合は、"K"の場合はキロバイト単位。"M"の場合はメガバイト単位。"G"の場合はギガバイト単位。 |
| 2 | HighHeapMemorySize | String | ヒープメモリの高負荷サイズを設定する。  有効な値の範囲は、0以上。デフォルトはRuntime#maxMemory()で取得できる値/2。そのメソッドがサポートされていない場合は、32MByte。  単位を指定しない場合は、バイト単位。単位を指定する場合は、"K"の場合はキロバイト単位。"M"の場合はメガバイト単位。"G"の場合はギガバイト単位。 |

### jp.ossc.nimbus.service.cache.CalculateMemorySizeOverflowValidatorService

キャッシュしたデータによるメモリの理論使用量が、閾値を超えるとあふれさせるOverflowValidatorインタフェースの実装クラスである。

OverflowController

validate()

**CalculateMemorySizeOverflowValidatorService**

最大メモリを閾値として持ち、以下の計算式で、あふれ件数を計算する。

（理論使用メモリ - 最大メモリ）÷（理論使用メモリ÷キャッシュされている件数）

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | MaxMemorySize | String | 使用メモリの最大サイズを設定する。  有効な値の範囲は、0以上。デフォルトはRuntime#maxMemory()で取得できる値 / 2。そのメソッドがサポートされていない場合は、32MByte。  単位を指定しない場合は、バイト単位。単位を指定する場合は、"K"の場合はキロバイト単位。"M"の場合はメガバイト単位。"G"の場合はギガバイト単位。 |
| 2 | CalculateProperty | boolean | オブジェクトに宣言されているgetterメソッドで取得できるprimitive型以外のオブジェクトのサイズを計算して加算するかどうかを設定する。  デフォルトはfalseで加算しない。 |
| 3 | CalculateOnValidate | boolean | あふれ検証時にメモリ使用量を計算し直すかどうかを設定する。  デフォルトはfalseで、キャッシュされた時点でメモリ使用量を計算する。  その場合、キャッシュ後に使用量が変わっても計算されない。 |

### jp.ossc.nimbus.service.cache.TimeExpierOverflowValidatorService

キャッシュした時刻から一定時間経過するとあふれさせるOverflowValidatorインタフェースの実装クラスである。

OverflowController

validate()

**TimeExpierOverflowValidatorService**

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | ExpierTerm | long | キャッシュの有効期間[ms]を設定する。  有効な値の範囲は、0以上。デフォルトは0で、溢れない。 |

## OverflowAlgorithm

### jp.ossc.nimbus.service.cache.FIFOOverflowAlgorithmService

キャッシュされたデータをFIFO（First In First Out）のアルゴリズムで、あふれさせるOverflowAlgorithmインタフェースの実装クラスである。

OverflowController

CachedReference

overflow()

**FIFOOverflowAlgorithmService**

FIFOアルゴリズムとは、先に入ったものから、あふれていくアルゴリズムである。

### jp.ossc.nimbus.service.cache.LFUOverflowAlgorithmService

キャッシュされたデータをLFU（Least Frequency Used）のアルゴリズムで、あふれさせるOverflowAlgorithmインタフェースの実装クラスである。

OverflowController

CachedReference

overflow()

**LFUOverflowAlgorithmService**

LFUアルゴリズムとは、最も使われていないものから、あふれていくアルゴリズムである。ここで言う、「最も使われていない」とは、最も参照回数が少ないことを表す。

また、使った時刻が古い場合には、参照回数を少なく見積もって、あふれやすくする機能もある。

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | CachedRatioCompare | boolean | 最後に参照された時刻が、古ければ古いほど、参照回数を弱めて評価するようにするかどうかを設定する。  デフォルトは、falseで、純粋な参照回数を評価する。 |
| 2 | RatioUnitTime | long | CachedRatioCompare属性がtrueの場合に、どのくらいの時間経過[ms]を基準に参照回数を弱めていくかを設定する。  デフォルトは、1秒。 |

### jp.ossc.nimbus.service.cache.LIFOOverflowAlgorithmService

キャッシュされたデータをLIFO（Last In First Out）のアルゴリズムで、あふれさせるOverflowAlgorithmインタフェースの実装クラスである。

OverflowController

CachedReference

overflow()

**LIFOOverflowAlgorithmService**

LIFOアルゴリズムとは、最後に入ったものから、あふれていくアルゴリズムである。

### jp.ossc.nimbus.service.cache.LRUOverflowAlgorithmService

キャッシュされたデータをLRU（Least Recently Used）のアルゴリズムで、あふれさせるOverflowAlgorithmインタフェースの実装クラスである。

OverflowController

CachedReference

overflow()

**LRUOverflowAlgorithmService**

LRUアルゴリズムとは、最も近く使われていないものから、あふれていくアルゴリズムである。ここで言う、「最も近く使われていない」とは、最後に使われた時刻が最も古いことを表す。

## OverflowAction

### jp.ossc.nimbus.service.cache.RemoveOverflowActionService

キャッシュされたデータを、キャッシュから削除するOverflowActionインタフェースの実装クラスである。

OverflowController

action()

**RemoveOverflowActionService**

### jp.ossc.nimbus.service.cache.StoreCacheOverflowActionService

キャッシュされたデータを、２次キャッシュに退避するOverflowActionインタフェースの実装クラスである。

OverflowController

action()

**StoreCacheOverflowActionService**

add() or put()

Cache or CacheMap

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | CacheServiceName | ServiceName | 退避先のCacheサービスのサービス名を設定する。 |
| 2 | CacheMapServiceName | ServiceName | 退避先のCacheMapサービスのサービス名を設定する。 |

### jp.ossc.nimbus.service.cache.SoftReferenceOverflowActionService

キャッシュされたデータを、強参照からソフト参照に変えて、同時に永続化先のキャッシュに退避するOverflowActionインタフェースの実装クラスである。ここで言う、ソフト参照とは、ガベージコレクションが発生すると、ガベージされる可能性がある参照である。ソフト参照がガベージされた時に、メモリの使用状況が変わるため、再度あふれ制御を行う。また、ガベージされた後に、キャッシュされたデータをCachedReferenceから取得しようとすると、永続化先のキャッシュからデータを取得して、元のキャッシュにデータを復元して返す。

OverflowController

action()

**SoftReferenceOverflowActionService**

add() or put()

Cache or CacheMap

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | PersistCacheServiceName | ServiceName | あふれたキャッシュオブジェクトをソフト参照にすると同時に永続化するための永続化Cacheサービスのサービス名を設定する。 |
| 2 | PersistCacheMapServiceName | ServiceName | あふれたキャッシュオブジェクトをソフト参照にすると同時に永続化するための永続化CacheMapサービスのサービス名を設定する。 |

### jp.ossc.nimbus.service.cache.ContextSaveOverflowActionService

CacheMapにキャッシュされたデータを、jp.ossc.nimbus.service.context.Context#save(Object)を使って永続化し、メモリ中から破棄するOverflowActionインタフェースの実装クラスである。

put(key,value)

CacheMap

control()

OverflowController

action()

**ContextSaveOverflowActionService**

save(key)

Context

以下にサービスの属性一覧を示す。

|  |  |  |  |
| --- | --- | --- | --- |
| # | 属性名 | 型 | 説明 |
| 1 | ContextServiceName | ServiceName | 永続化先のContextサービスのサービス名を設定する。 |