

Use Case Scheduling

Rank	Use Case Name	Justification
Unspecified	resource_retriving	
Unspecified	resource_creating	
Unspecified	resource_uncaching	
Unspecified	resource_caching	
Unspecified	OnDisk_caching	
Unspecified	InMemory_caching	
Unspecified	resource_statistics_counting	
Unspecified	Creation_time_counting	
Unspecified	GetFromDisk_time_counting	
Unspecified	Resource_size_counting	
Unspecified	Memory-Disk_shifting	
Unspecified	Cache_clearing	
Unspecified	DBMS_mapping	
Unspecified	Cache_optimizing	
Unspecified	invalidating	

Requirements

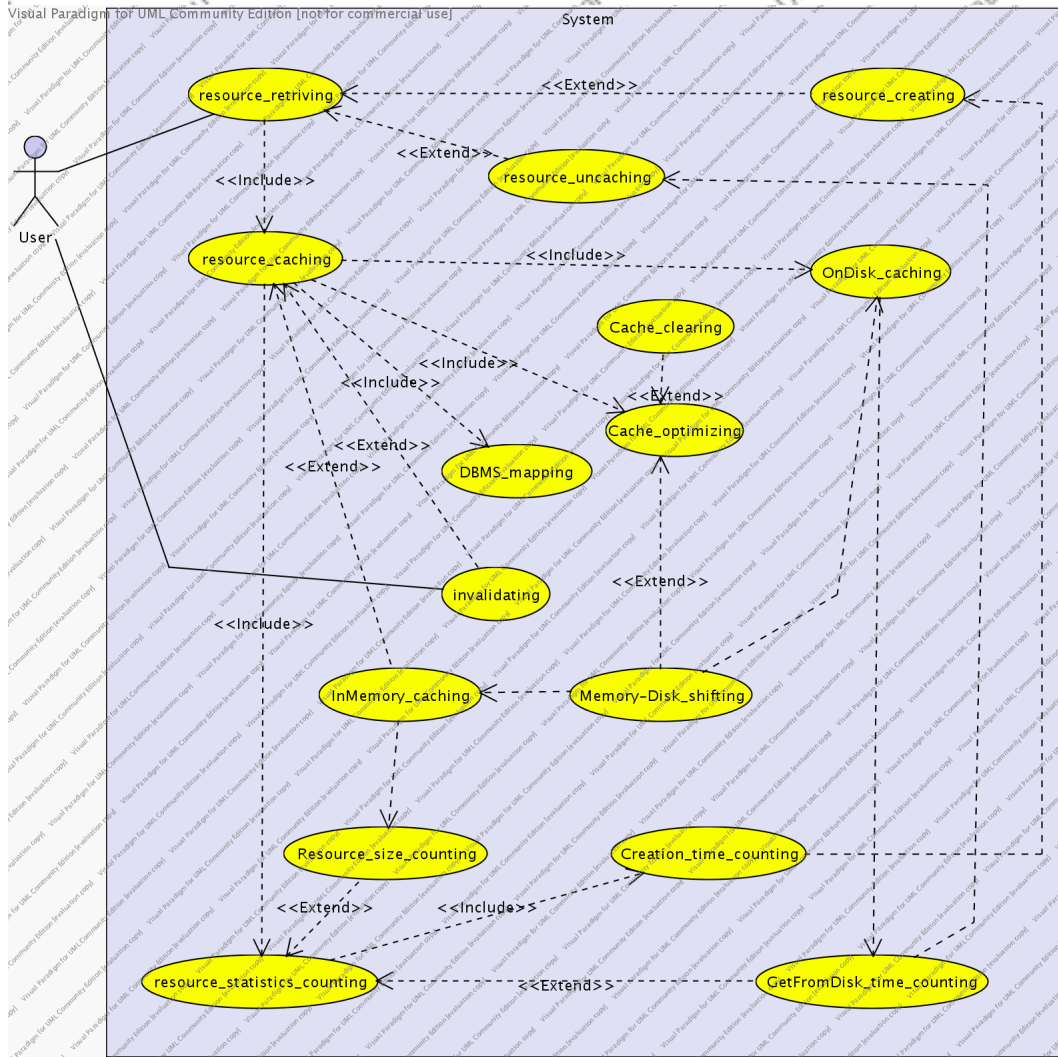


Diagram Content Summary

- [System](#)
- [resource_retriving](#)
- [resource_creating](#)
- [resource_uncaching](#)
- [resource_caching](#)
- [OnDisk_caching](#)












-  [InMemory_caching](#)
-  [resource_statistics_counting](#)
-  [Creation_time_counting](#)
-  [GetFromDisk_time_counting](#)
-  [Resource_size_counting](#)
-  [Memory-Disk_shifting](#)
-  [Cache_clearing](#)
-  [DBMS_mapping](#)
-  [Cache_optimizing](#)
-  [invalidating](#)
-  [User](#)

Diagram Content Detail

UseCase resource_retriving

Parent : [System](#)

Include:

[resource_caching](#)

Extended by

[resource_creating](#) ,

[resource_uncaching](#)

Extension Point

Name : ExtensionPoint

Name : ExtensionPoint

Communication Link [communicationlink to User](#)

Sub-diagrams

[ResourceRetrieval](#)

UseCase resource_creating

Use Case Extends Hierarchy

resource_retriving

+-resource_creating

Parent : [System](#)

Extend from

[resource_retriving](#)

Depended by:

[Creation_time_counting](#)

 **UseCase** resource_uncaching

Use Case Extends Hierarchy

```
resource_retriving
|
+-resource_uncaching
```


Parent : [System](#)

Extend from

[resource_retriving](#)

Depended by:

[GetFromDisk_time_counting](#)

 **UseCase** resource_caching

Parent : [System](#)

Include:

[OnDisk_caching](#), [resource_statistics_counting](#), [DBMS_mapping](#), [Cache_optimizing](#)

Include by:

[resource_retriving](#)

Extended by

[InMemory_caching](#),
[invalidating](#)

Extension Point

Name : ExtensionPoint

Name : ExtensionPoint

 **UseCase** OnDisk_caching

Parent : [System](#)

Include by:

[resource_caching](#)

Depend to:

[GetFromDisk_time_counting](#)

Depended by:

[Memory-Disk_shifting](#)



UseCase InMemory_caching

Use Case Extends Hierarchy

resource_caching

|

+--InMemory_caching

Parent : [System](#)

Extend from

[resource_caching](#)

Depend to:

[Resource_size_counting](#)

Depended by:

[Memory-Disk_shifting](#)



UseCase resource_statistics_counting

Parent : [System](#)

Include:

[Creation_time_counting](#)

Include by:

[resource_caching](#)

Extended by

[GetFromDisk_time_counting](#),

[Resource_size_counting](#)

Extension Point

Name : ExtensionPoint

Name : ExtensionPoint



UseCase Creation_time_counting

Parent : [System](#)

Include by:

[resource_statistics_counting](#)

Depend to:

[resource_creating](#)



UseCase GetFromDisk_time_counting

Use Case Extends Hierarchy

```
resource_statistics_counting
|
+-GetFromDisk_time_counting
```

Parent : [System](#)

Extend from

[resource_statistics_counting](#)

Depend to:

[resource_uncaching](#)

Depended by:

[OnDisk_caching](#)



UseCase Resource_size_counting

Use Case Extends Hierarchy

```
resource_statistics_counting
|
+-Resource_size_counting
```

Parent : [System](#)

Extend from

[resource_statistics_counting](#)

Depended by:

[InMemory_caching](#)



UseCase Memory-Disk_shifting

Use Case Extends Hierarchy

Cache_optimizing

|

+Memory-Disk_shifting

Parent : [System](#)

Extend from

[Cache_optimizing](#)

Depend to:

[InMemory_caching](#), [OnDisk_caching](#)



UseCase Cache_clearing

Use Case Extends Hierarchy

Cache_optimizing

|

+Cache_clearing

Parent : [System](#)

Extend from

[Cache_optimizing](#)



UseCase DBMS_mapping

Parent : [System](#)

Include by:

[resource_caching](#)



UseCase Cache_optimizing

Parent : [System](#)

Include by:

[resource_caching](#)

Extended by


[Memory-Disk_shifting](#),

[Cache_clearing](#)

Extension Point

Name : ExtensionPoint

Name : ExtensionPoint

 **UseCase** invalidating

Use Case Extends Hierarchy

resource_caching

+--invalidating

Parent : [System](#)

Extend from

[resource_caching](#)

Communication Link [communicationlink](#) to [User](#)

 **Actor** User

Communication Link [communicationlink](#) to [invalidating](#)

Communication Link [communicationlink](#) to [resource_retriving](#)

 **System** System

Children:

[resource_retriving](#), [resource_creating](#), [resource_uncaching](#), [resource_caching](#), [OnDisk_caching](#),
[InMemory_caching](#), [resource_statistics_counting](#), [Creation_time_counting](#),
[GetFromDisk_time_counting](#), [Resource_size_counting](#), [Memory-Disk_shifting](#), [Cache_clearing](#),
[DBMS_mapping](#), [Cache_optimizing](#), [invalidating](#)

Communication Link

Communication Link End From

Element : [resource_retriving](#)

Communication Link End To

Element : [User](#)

Communication Link

Communication Link End From

Element : [User](#)

Communication Link End To

Element : [invalidating](#)

ClassDiagram

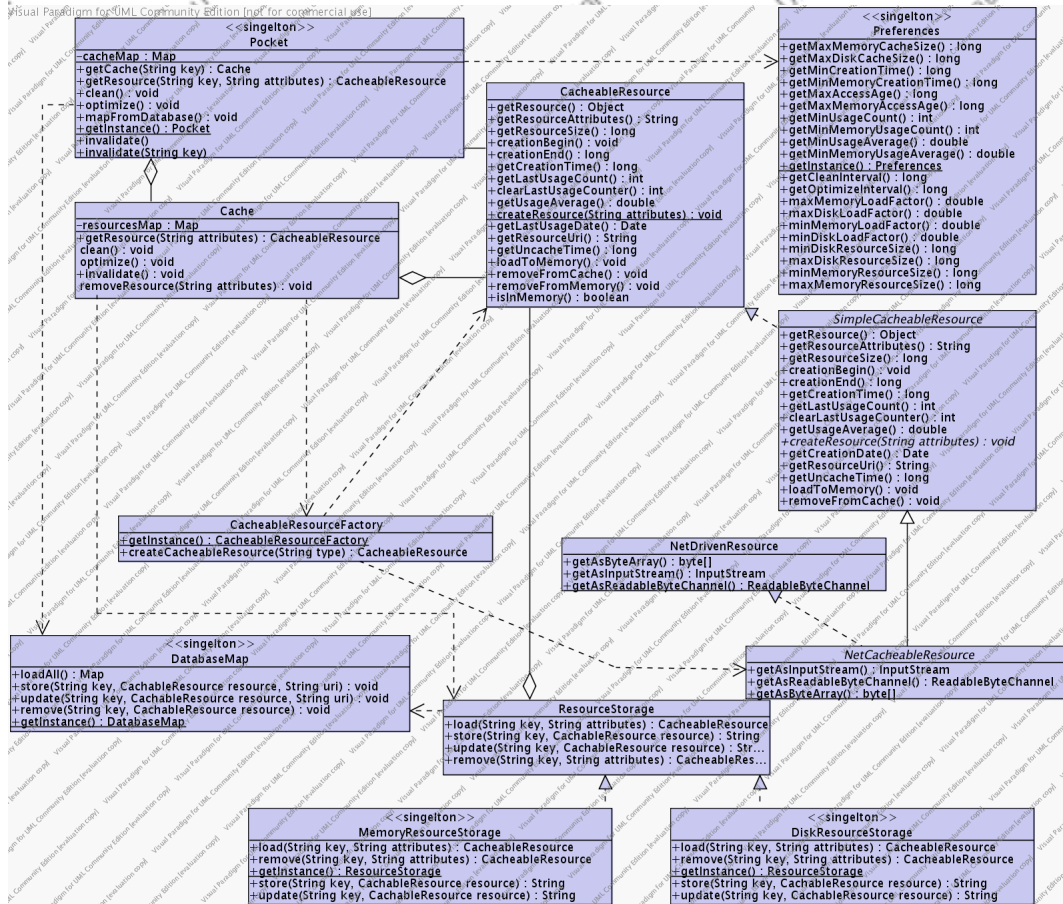


Diagram Content Summary

- [CacheableResource](#)
- [Cache](#)
- [Pocket](#)
- [ResourceStorage](#)
- [MemoryResourceStorage](#)
- [DiskResourceStorage](#)
- [DatabaseMap](#)
- [Preferences](#)
- [SimpleCacheableResource](#)


[CacheableResourceFactory](#)
[NetDrivenResource](#)
[NetCacheableResource](#)

Diagram Content Detail


Class CacheableResource

Realized by:

[SimpleCacheableResource](#)
Association [association to Cache](#)
Association [association to Pocket](#)
Association [association to ResourceStorage](#)

Depended by:

[CacheableResourceFactory](#)

Operation Summary

public Object	getResource()
public String	getResourceAttributes()
public long	getResourceSize()
public void	creationBegin()
public long	creationEnd()
public long	getCreationTime()
public int	getLastUsageCount()
public int	clearLastUsageCounter()
public double	getUsageAverage()
public void	createResource(String attributes)
public Date	getLastUsageDate()
public String	getResourceUri()
public long	getUncacheTime()
public void	loadToMemory()
public void	removeFromCache()
public void	removeFromMemory()
public boolean	isInMemory()

Operation Detail

public [Object](#) getResource()

Documentation

@return

Return Type : [Object](#)

```
public String getResourceAttributes()
```

Return Type : [String](#)

```
public long getResourceSize()
```

Return Type : [long](#)

```
public void creationBegin()
```

Return Type : [void](#)

```
public long creationEnd()
```

Return Type : [long](#)

```
public long getCreationTime()
```

Return Type : [long](#)

```
public int getLastUsageCount()
```

Return Type : [int](#)

```
public int clearLastUsageCounter()
```

Return Type : `int`

```
public double getUsageAverage()
```

Return Type : `double`

```
public void createResource( String attributes)
```

Scope : `classifier`

Parameter Detail

Name : `String` attributes

Type :

Return Type : `void`

```
public Date getLastUsageDate()
```

Return Type : `Date`

```
public String getResourceUri()
```

Return Type : `String`

```
public long getUncacheTime()
```

Return Type : `long`

```
public void loadToMemory()
```

Return Type : `void`

```
public void removeFromCache()
```

Return Type : void

```
public void removeFromMemory()
```

Return Type : void

```
public boolean isInMemory()
```

Return Type : boolean



Class Cache

Documentation

@author Sebastian Ryszard Kruk

Visibility : protected

Association [association to CacheableResource](#)

Association [association to Pocket](#)

Depend to:

[ResourceStorage](#), [CacheableResourceFactory](#)

Attribute Summary

private Map	resourcesMap
-------------	--------------

Operation Summary

public CacheableResource	getResource(String attributes)
package void	clean()
package void	optimize()
public void	invalidate()
package void	removeResource(String attributes)

Attribute Detail

```
private Map resourcesMap
```

Operation Detail

```
public CacheableResource getResource( String attributes)
```

Parameter Detail

Name : String attributes

Type :

Return Type : [CacheableResource](#)

```
package void clean()
```

Return Type : void

```
package void optimize()
```

Return Type : void

```
public void invalidate()
```

Return Type : void

```
package void removeResource( String attributes)
```

Parameter Detail

Name : String attributes

Type :

Return Type : void

**Class** Pocket**Documentation**

@author Sebastian Ryszard Kruk

Visibility : protected**Association** [association to Cache](#)**Association** [association to CacheableResource](#)**Depend to:**[DatabaseMap](#), [Preferences](#)**Attribute Summary**

private Map cacheMap

Operation Summary

public Cache	getCache(String key)
public CacheableResource	getResource(String key, String attributes)
public void	clean()
public void	optimize()
public void	mapFromDatabase()
public Pocket	getInstance()
public	invalidate()
public	invalidate(String key)

Attribute Detailprivate [Map](#) cacheMap**Operation Detail**public [Cache](#) getCache(String key)**Parameter Detail****Name** : String key

Type :

Return Type : [Cache](#)

```
public CacheableResource getResource( String key, String attributes)
```

Parameter Detail

Name : String key

Type :

Name : String attributes

Type :

Return Type : [CacheableResource](#)

```
public void clean()
```

Return Type : void

```
public void optimize()
```

Return Type : void

```
public void mapFromDatabase()
```

Return Type : void

```
public Pocket getInstance()
```

Scope : classifier

Return Type : [Pocket](#)

```
public invalidate()
```

Return Type:

```
public invalidate( String key)
```

Parameter Detail

Name : String key

Type :

Return Type :



Class ResourceStorage

Documentation

@author Sebastian Ryszard Kruk

Realized by:

[DiskResourceStorage](#), [MemoryResourceStorage](#)

Association [association to CacheableResource](#)

Depend to:

[DatabaseMap](#)

Depended by:

[Cache](#)

Operation Summary

public CacheableResource	load(String key, String attributes)
public String	store(String key, CacheableResource resource)
public String	update(String key, CacheableResource resource)
public CacheableResource	remove(String key, String attributes)

Operation Detail

```
public CacheableResource load( String key, String attributes)
```

Parameter Detail

Name : String key

Type :

Name : String attributes

Type :

Return Type : [CacheableResource](#)

```
public String store( String key, CacheableResource resource)
```

Parameter Detail

Name : String key

Type :

Name : CacheableResource resource

Type :

Return Type : String

```
public String update( String key, CacheableResource resource)
```

Parameter Detail

Name : String key

Type :

Name : CacheableResource resource

Type :

Return Type : String

```
public CacheableResource remove( String key, String attributes)
```

Parameter Detail

Name : String key

Type :

Name : String attributes

Type :

Return Type : [CacheableResource](#)



Class MemoryResourceStorage

Documentation

@author Sebastian Ryszard Kruk

Realization:

[ResourceStorage](#)

Operation Summary

public CacheableResource	load(String key, String attributes)
public CacheableResource	remove(String key, String attributes)
public ResourceStorage	getInstance()
public String	store(String key, CacheableResource resource)
public String	update(String key, CacheableResource resource)

Operation Detail

```
public CacheableResource load( String key, String attributes)
```

Parameter Detail

Name : String key

Type :

Name : String attributes

Type :

Return Type : [CacheableResource](#)

```
public CacheableResource remove( String key, String attributes)
```

Parameter Detail

Name : String key

Type :

Name : String attributes

Type :

Return Type : [CacheableResource](#)

```
public ResourceStorage getInstance()
```

Scope : classifier

Return Type : [ResourceStorage](#)

```
public String store( String key, CacheableResource resource)
```

Parameter Detail

Name : String key

Type :

Name : [CacheableResource](#) resource

Type :

Return Type : [String](#)

```
public String update( String key, CacheableResource resource)
```

Parameter Detail

Name : String key

Type :

Name : [CacheableResource](#) resource

Type :

Return Type : [String](#)



Class [DiskResourceStorage](#)

Documentation

@author Sebastian Ryszard Kruk

Realization:

[ResourceStorage](#)

Operation Summary	
public CacheableResource	load(String key, String attributes)
public CacheableResource	remove(String key, String attributes)
public ResourceStorage	getInstance()
public String	store(String key, CacheableResource resource)
public String	update(String key, CacheableResource resource)

Operation Detail

```
public CacheableResource load( String key, String attributes)
```

Parameter Detail

Name : String key

Type :

Name : String attributes

Type :

Return Type : [CacheableResource](#)

```
public CacheableResource remove( String key, String attributes)
```

Parameter Detail

Name : String key

Type :

Name : String attributes

Type :

Return Type : [CacheableResource](#)

ResourcePocket - Java caching library

```
public ResourceStorage getInstance()
```

Scope : classifier

Return Type : [ResourceStorage](#)

```
public String store( String key, CacheableResource resource)
```

Parameter Detail

Name : [String](#) key

Type :

Name : [CacheableResource](#) resource

Type :

Return Type : [String](#)

```
public String update( String key, CacheableResource resource)
```

Parameter Detail

Name : [String](#) key

Type :

Name : [CacheableResource](#) resource

Type :

Return Type : [String](#)



Class [DatabaseMap](#)

Documentation

@author Sebastian Ryszard Kruk

Depended by:

[ResourceStorage](#), [Pocket](#)

Operation Summary

ResourcePocket - Java caching library

public Map	loadAll()
public void	store(String key, CacheableResource resource, String uri)
public void	update(String key, CacheableResource resource, String uri)
public void	remove(String key, CacheableResource resource)
public DatabaseMap	getInstance()

Operation Detail

```
public Map loadAll()
```

Return Type : [Map](#)

```
public void store( String key, CacheableResource resource, String uri)
```

Parameter Detail

Name : String key

Type :

Name : CacheableResource resource

Type :

Name : String uri

Type :

Return Type : [void](#)

```
public void update( String key, CacheableResource resource, String uri)
```

Parameter Detail

Name : String key

Type :

Name : CacheableResource resource

Type :

ResourcePocket - Java caching library

Name : String uri

Type :

Return Type : void

```
public void remove( String key, CacheableResource resource)
```

Parameter Detail

Name : String key

Type :

Name : CacheableResource resource

Type :

Return Type : void

```
public DatabaseMap getInstance()
```

Scope : classifier

Return Type : [DatabaseMap](#)



Class Preferences

Documentation

@author Sebastian Ryszard Kruk

Depended by:

[Pocket](#)

Operation Summary	
public long	getMaxMemoryCacheSize()
public long	getMaxDiskCacheSize()
public long	getMinCreationTime()
public long	getMinMemoryCreationTime()
public long	getMaxAccessAge()
public long	getMaxMemoryAccessAge()
public int	getMinUsageCount()

ResourcePocket - Java caching library

public int	getMinMemoryUsageCount()
public double	getMinUsageAverage()
public double	getMinMemoryUsageAverage()
public Preferences	getInstance()
public long	getCleanInterval()
public long	getOptimizeInterval()
public double	maxMemoryLoadFactor()
public double	maxDiskLoadFactor()
public double	minMemoryLoadFactor()
public double	minDiskLoadFactor()
public long	minDiskResourceSize()
public long	maxDiskResourceSize()
public long	minMemoryResourceSize()
public long	maxMemoryResourceSize()

Operation Detail

```
public long getMaxMemoryCacheSize()
```

Return Type : long

```
public long getMaxDiskCacheSize()
```

Return Type : long

```
public long getMinCreationTime()
```

Return Type : long

```
public long getMinMemoryCreationTime()
```

Return Type : long

```
public long getMaxAccessAge()
```

Return Type : [long](#)

```
public long getMaxMemoryAccessAge()
```

Return Type : [long](#)

```
public int getMinUsageCount()
```

Return Type : [int](#)

```
public int getMinMemoryUsageCount()
```

Return Type : [int](#)

```
public double getMinUsageAverage()
```

Return Type : [double](#)

```
public double getMinMemoryUsageAverage()
```

Return Type : [double](#)

```
public Preferences getInstance()
```

Scope : classifier

Return Type : [Preferences](#)

```
public long getCleanInterval()
```

Return Type : [long](#)

```
public long getOptimizeInterval()
```

Return Type : [long](#)

```
public double maxMemoryLoadFactor()
```

Return Type : [double](#)

```
public double maxDiskLoadFactor()
```

Return Type : [double](#)

```
public double minMemoryLoadFactor()
```

Return Type : [double](#)

```
public double minDiskLoadFactor()
```

Return Type : [double](#)

```
public long minDiskResourceSize()
```

Return Type : [long](#)

```
public long maxDiskResourceSize()
```

Return Type : [long](#)

```
public long minMemoryResourceSize()
```

Return Type : long

public long maxMemoryResourceSize()

Return Type : long**Class** SimpleCacheableResource**Is Abstract :** true**Realization:**[CacheableResource](#)**Subclasses**[NetCacheableResource](#)**Operation Summary**

public Object	getResource()
public String	getResourceAttributes()
public long	getResourceSize()
public void	creationBegin()
public long	creationEnd()
public long	getCreationTime()
public int	getLastUsageCount()
public int	clearLastUsageCounter()
public double	getUsageAverage()
public void	createResource(String attributes)
public Date	getCreationDate()
public String	getResourceUri()
public long	getUncacheTime()
public void	loadToMemory()
public void	removeFromCache()

Operation Detail

public Object getResource()

Documentation

@return

Return Type : [Object](#)

```
public String getResourceAttributes()
```

Return Type : [String](#)

```
public long getResourceSize()
```

Return Type : [long](#)

```
public void creationBegin()
```

Return Type : [void](#)

```
public long creationEnd()
```

Return Type : [long](#)

```
public long getCreationTime()
```

Return Type : [long](#)

```
public int getLastUsageCount()
```

Return Type : [int](#)

```
public int clearLastUsageCounter()
```

Return Type : [int](#)

```
public double getUsageAverage()
```

Return Type : [double](#)

```
public void createResource( String attributes)
```

Is Abstract : [true](#)

Parameter Detail

Name : [String](#) attributes

Type :

Return Type : [void](#)

```
public Date getCreationDate()
```

Return Type : [Date](#)

```
public String getResourceUri()
```

Return Type : [String](#)

```
public long getUncacheTime()
```

Return Type : [long](#)

```
public void loadToMemory()
```

Return Type : [void](#)

```
public void removeFromCache()
```

Return Type: void



Class CacheableResourceFactory

Depend to:

[CacheableResource](#), [NetCacheableResource](#)

Depended by:

[Cache](#)

Operation Summary

public	getInstance()
CacheableResourceFactory	
public CacheableResource	createCacheableResource(String type)

Operation Detail

public [CacheableResourceFactory](#) getInstance()

Scope: classifier

Return Type: [CacheableResourceFactory](#)

public [CacheableResource](#) createCacheableResource(String type)

Parameter Detail

Name: String type

Type:

Return Type: [CacheableResource](#)



Class NetDrivenResource

Realized by:

[NetCacheableResource](#)

Operation Summary

ResourcePocket - Java caching library

public byte[]	getAsByteArray()
public InputStream	getAsInputStream()
public ReadableByteChannel	getAsReadableByteChannel()

Operation Detail

```
public byte\[\] getAsByteArray\(\)
```

Return Type : [byte\[\]](#)

```
public InputStream getAsInputStream\(\)
```

Return Type : [InputStream](#)

```
public ReadableByteChannel getAsReadableByteChannel\(\)
```

Return Type : [ReadableByteChannel](#)



Class [NetCacheableResource](#)

Generalization Hierarchy

[SimpleCacheableResource](#)

+-[NetCacheableResource](#)

Is Abstract : true

Realization:

[NetDrivenResource](#)

Super Class

[SimpleCacheableResource](#)

Depended by:

[CacheableResourceFactory](#)

Operation Summary	
public InputStream	getAsInputStream()
public ReadableByteChannel	getAsReadableByteChannel()
public byte[]	getAsByteArray()

Operation Detail

```
public InputStream getAsInputStream()
```

Return Type : [InputStream](#)

```
public ReadableByteChannel getAsReadableByteChannel()
```

Return Type : [ReadableByteChannel](#)

```
public byte[] getAsByteArray()
```

Return Type : [byte](#)[]

Association

Association End From

Element : [Cache](#)

Aggregation Kind : Aggregation

Association End To

Element : [CacheableResource](#)

Association

Association End From

Element : [Pocket](#)

Aggregation Kind : Aggregation

Association End To

Element : [Cache](#)

Association

Association End From

Element : [Pocket](#)

Association End To

Element : [CacheableResource](#)

Association

Association End From

Element : [ResourceStorage](#)

Aggregation Kind : Aggregation

Association End To

Element : [CacheableResource](#)

ResourceRetrieval

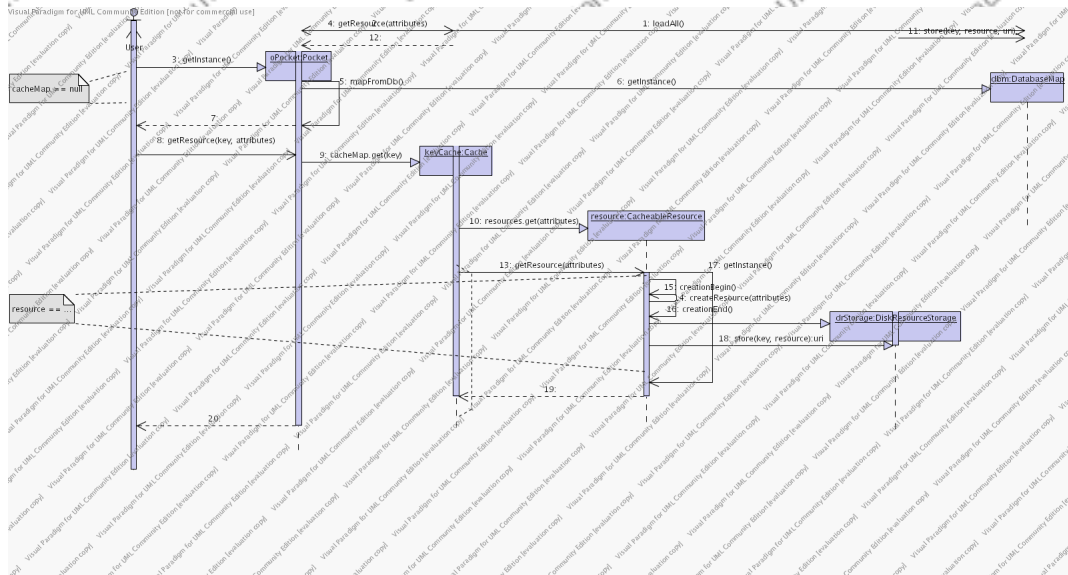


Diagram Content Summary

- [User](#)
- [oPocket](#)
- [keyCache](#)
- [resource](#)
- [dbm](#)
- [drStorage](#)

Diagram Content Detail

Actor User

Create Message to [oPocket](#)

Name : getInstance()
Sequence number: 3

Message to [oPocket](#)

Name : getResource(key, attributes)

Sequence number: 8

Message from

[oPocket](#)



Object oPocket

Base Classifier: [Pocket](#)

Message to [dbm](#)

Name : loadAll()

Sequence number: 1

Return Message to [User](#)

Name :

Sequence number: 20

Return action detail

Name : Return

Message to [keyCache](#)

Name : getResource(attributes)

Sequence number: 4

Message to [oPocket](#)

Name : mapFromDb()

Sequence number: 5

Create Message to [dbm](#)

Name : getInstance()

Sequence number: 6

Return Message to [User](#)

Name :

Sequence number: 7

Return action detail

Name : Return

Create Message to [keyCache](#)

Name : cacheMap.get(key)

Sequence number: 9

Create Message from

[User](#)

Message from

[User](#), [keyCache](#)



Object [keyCache](#)

Base Classifier: [Cache](#)

Create Message to [resource](#)

Name : resources.get(attributes)

Sequence number: 10

Return Message to [oPocket](#)

Name :

Sequence number: 12

Return action detail

Name : Return

Message to [resource](#)

Name : getResource(attributes)

Sequence number: 13

Return Message to [oPocket](#)

Name :

Sequence number: 2

Return action detail

Name : Return

Create Message from

[oPocket](#)

Message from

[oPocket](#), [resource](#)



Object [resource](#)

Base Classifier: [CacheableResource](#)

Message to [resource](#)

Name : createResource(attributes)

Sequence number: 14

Message to [resource](#)

Name : creationBegin()

Sequence number: 15

Message to [resource](#)

Name : creationEnd()

Sequence number: 16

Create Message to [drStorage](#)

Name : getInstance()

Sequence number: 17

Call Message to [drStorage](#)

Name : store(key, resource):uri

Sequence number: 18

Call action detail

Name : Call

Return Message to [keyCache](#)

Name :

Sequence number: 19

Return action detail

Name : Return

Create Message from

[keyCache](#)

Message from

[keyCache](#)



Object dbm

Base Classifier: [DatabaseMap](#)

Create Message from

[oPocket](#)

Message from

[oPocket](#), [drStorage](#)



Object [drStorage](#)

Base Classifier: [DiskResourceStorage](#)

Send Message to [dbm](#)

Name : store(key, resource, uri)

Sequence number: 11

Asynchronous : true

Send action detail

Name : Send

Create Message from

[resource](#)

Message from

[resource](#)

Activity Diagram 1

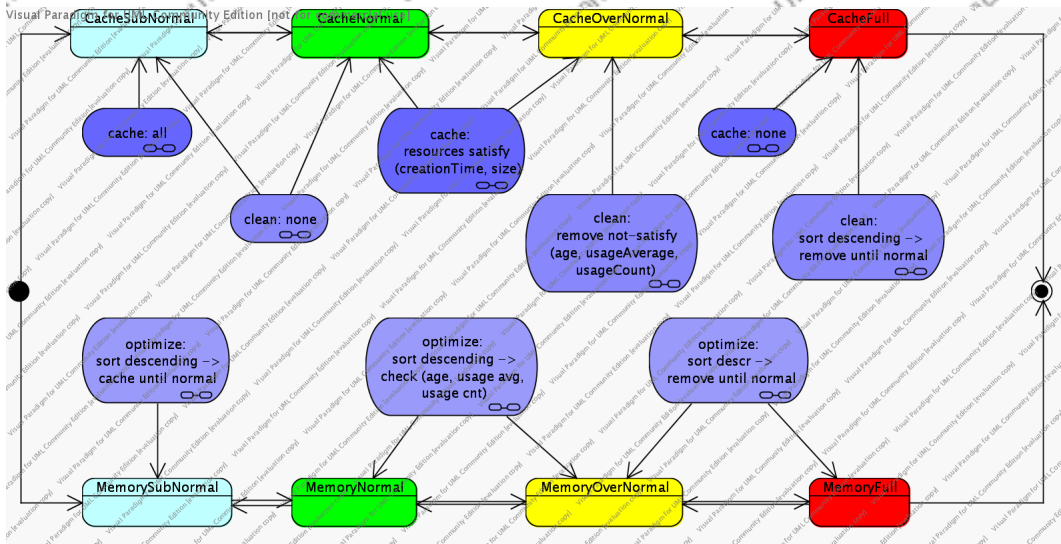


Diagram Content Summary

- InitialState
- [cache: all](#)
- [cache: resources satisfy \(creationTime, size\)](#)
- [clean: none](#)
- [clean: remove not-satisfy \(age, usageAverage, usageCount\)](#)
- [clean: sort descending -> remove until normal](#)
- [cache: none](#)
- [optimize: sort descending -> cache until normal](#)
- [optimize: sort descending -> check \(age, usage avg, usage cnt\)](#)
- [optimize: sort descr -> remove until normal](#)
- [CacheNormal](#)
- [CacheOverNormal](#)
- [CacheFull](#)
- [MemorySubNormal](#)
- [CacheSubNormal](#)
- [MemoryNormal](#)
- [MemoryOverNormal](#)
- [MemoryFull](#)

 [FinalState](#)

Diagram Content Detail

 **Initial State** InitialState

Transition link to [CacheSubNormal](#)

Transition link to [MemorySubNormal](#)

 **Sub Activity** cache: all

Transition link to [CacheSubNormal](#)

 **Sub Activity** cache: resources satisfy (creationTime, size)


Transition link to [CacheNormal](#)

Transition link to [CacheOverNormal](#)

 **Sub Activity** clean: none

Transition link to [CacheSubNormal](#)

Transition link to [CacheNormal](#)

 **Sub Activity** clean: remove not-satisfy (age, usageAverage, usageCount)

Transition link to [CacheOverNormal](#)

 **Sub Activity** clean: sort descending -> remove until normal

Transition link to [CacheFull](#)

 **Sub Activity** cache: none

Transition link to [CacheFull](#)


 **Sub Activity** optimize: sort descending -> cache until normal

Transition link to [MemorySubNormal](#)

 **Sub Activity** optimize: sort descending -> check (age, usage avg, usage cnt)

Transition link to [MemoryNormal](#)

Transition link to [MemoryOverNormal](#)

 **Sub Activity** optimize: sort descr -> remove until normal

Transition link to [MemoryOverNormal](#)

Transition link to [MemoryFull](#)

 **State** CacheNormal

Transition link to [CacheOverNormal](#)

Transition link to [CacheSubNormal](#)

Transition link from

[CacheOverNormal](#), [CacheSubNormal](#), clean: none, cache: [resources satisfy \(creationTime, size\)](#)



State CacheOverNormal

Transition link to [CacheFull](#)

Transition link to [CacheNormal](#)

Transition link from

[CacheNormal](#), [CacheFull](#), [cache: resources satisfy \(creationTime, size\), clean: remove not-satisfy \(age, usageAverage, usageCount\)](#)



State CacheFull

Transition link to [FinalState](#)

Transition link to [CacheOverNormal](#)

Transition link from

[CacheOverNormal](#), [cache: none, clean: sort descending -> remove until normal](#)



State MemorySubNormal

Transition link to [MemoryNormal](#)

Transition link from

[InitialState](#), [optimize: sort descending -> cache until normal, MemoryNormal](#)



State CacheSubNormal

Transition link to [CacheNormal](#)

Transition link from

[InitialState](#), [CacheNormal](#), [cache: all, clean: none](#)



State MemoryNormal

Transition link to [MemoryOverNormal](#)

Transition link to [MemorySubNormal](#)

Transition link from

[MemorySubNormal](#), optimize: sort descending -> check (age, usage avg, usage cnt),
[MemoryOverNormal](#)



State MemoryOverNormal

Transition link to [MemoryFull](#)

Transition link to [MemoryNormal](#)

Transition link from

[MemoryNormal](#), optimize: sort descending -> check (age, usage avg, usage cnt), optimize:
sort descr -> remove until normal, [MemoryFull](#)



State MemoryFull

Transition link to [FinalState](#)

Transition link to [MemoryOverNormal](#)

Transition link from

[MemoryOverNormal](#), optimize: sort descr -> remove until normal



Final State FinalState

Transition link from

[CacheFull](#), [MemoryFull](#)

