Cocoon Source Resolving (2.1 legacy document)

Table of contents

Table of contents

Warning:
This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

1. The Store Components in Cocoon

To keep cached data, Cocoon uses components implementing the *org.apache.excalibur.Store* interface. Cocoon uses two implementations of this interface, the "transient store" and the "store", and optionally a third one, the "persistent store".

1.1. The Transient store

The transient store (role *Store.TRANSIENT_STORE* is used for objects that are not serializable, or whose storage doesn't make sense across server restart. The transient store lives on its own and has no relation with other stores. This is a mandatory component within Cocoon (i.e. used by Cocoon's code). Transient store can drop stored components if system is running low on memory.

Cocoon uses the transient store to cache XSLT style sheets, XSP logicsheets, etc.

1.2. The Store (aka "main Store")

The store (role *Store.ROLE*) can hold not serializable objects also. It is a mandatory component, as the transient store. If memory is scarce, the store can drop non serializable objects, and swaps serializable objects to disk. For efficiency reasons, implementations of the store should swap out (or drop) least often used objects. On shutdown, store can write out all objects residing in memory to the disk.

Cocoon uses the main store to cache pipeline output.

1.3. Persistent store (optional)

Some store (*Store.ROLE*) implementations (but not all) may actually be just an in-memory cache that swap objects by calling the persistent store (*Store.PERSISTENT_STORE*) when needed. So the persistent store is an *optional* component which is used only by MRUMemoryStore, and nowhere else in the code.

Two examples to illustrate this:

- when using JISP, we had this mechanism : the store was a MRUMemoryStore swapping to the persistent store which was a JISPStore.
- JCS has its own in-memory front end to its own persistent storage. In this configuration Store.ROLE will be a JCSStore and Store.PERSISTENT_STORE will have no implementation, because we don't need it.

2. Summary

- Store.TRANSIENT_STORE : used by Cocoon to store non-serializable objects
- Store.ROLE : used by Cocoon to store serializable objects
- Store.PERSISTENT_STORE : optional component that may be used by in-memory implementations of Store.ROLE to delegate persistent storage.

1. Comments

add your comments