

# StoreComponents

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**.

## 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).

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

## Store (aka. "main Store")

The store (role `Store.ROLE`) is used to store objects that are serializable. As the transient store, it is a mandatory component. For efficiency reasons, implementations of the main store should keep most-often used objects in memory and swap older ones to disk.

Cocoon uses the main store to cache pipeline output.

## 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.

## To sum up :

- `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.

– [SylvainWallez](#)