

My application seems to be using a lot of memory since I started using Transfer

By default transfer will cache indefinitely every object it creates. However, it will give memory back if the JVM needs it. See the documentation on [Managing the Cache](#).

If you see a big instantaneous memory hit, check your composition relationships. When loading a TransferObject with a *OneToMany* or a *ManyToOne* relationship, Transfer will normally load all of its children as well (and their children etc...).

If this is a problem, you can control it with the ["lazy" configuration attribute](#), or, if the composition is *OneToMany*, you can change the relationship to *ManyToOne*.

You can also configure multiple aspects of the cache to limit the number of TransferObjects that are cached, how long they are cached for, and what scope they are cached in. Which configuration options you use is greatly influenced by your application design, and it may not need to be configured at all. More details on the caching options can be found in the [Managing the Cache](#) and ["objectCache"](#) sections of the Transfer file configuration documentation.

For example, if I want to limit a TransferObject of class 'user.User' to only be cached within the request of the page, I could set it up like this, within my *objectCache* element in my configuration file.

```
<cache class="user.User">  
  <scope type="request"/>  
</cache>
```

Categories: • [Cache](#)