VfsCacheStrategy

VFS - CacheStragey

One problem with libraries like vfs is that they cache some data to avoid successive access to the real file - simply to speed up things.

This cache might get stale - no it GET stale.

Currently VFS provides three strategies to workaround it:

- manual
- on resolve (the default)
- on call

Note: It is not possible to use `VFS.getManager` to configure the cacheStrategy. You have to create your own static class to create it.

manual cache strategy

Setup

```
StandardFileSystemManager fs = new StandardFileSystemManager();
fs.setCacheStrategy(CacheStrategy.MANUAL);
fs.init();
```

using this setup you have to use `fileObject.refresh()` to refresh your object with the filesystem

on_resolve

Setup

```
StandardFileSystemManager fs = new StandardFileSystemManager();
fs.setCacheStrategy(CacheStrategy.ON_RESOLVE);
fs.init();
```

every time you call `fs.resolveFile()` the file data will be refreshed. You still can use `fileObject.refresh()` to refresh the data on demand.

on_call

Setup

```
StandardFileSystemManager fs = new StandardFileSystemManager();
fs.setCacheStrategy(CacheStrategy.ON_CALL);
fs.init();
```

Every time you call a method on the resolve file object the data will be refreshed with the filesystem. This will give you the behaviour you might expect from a local file but also might be a hughe performance loss as it will greatly increase the network load.

You can also achieve this cache strategy by wrapping the file object in an `org.apache.commons.vfs.cache.OnCallRefreshFileObject`

```
FileObject fo = VFS.getManager().resolveFile("....");
OnCallRefreshFileObject foc = new OnCallRefreshFileObject(fo);
```

The difference to the above is, that in the first case you will always get the same file object instance and thus you can synchronize on it.