Core - Feature # 60221

Status:	New	**Priority:**	Could have
Author:	Bernhard Kraft	**Category:**	File Abstraction Layer (FAL)
Created:	2014-07-10	**Assigned To:**	
Updated:	2015-06-15	**Due date:**	
PHP Version:			
Complexity:			
Sprint Focus:			
Subject:	Read FAL resources in chunks		

Description

Usually file layers like the Linux file system support a basic common set of IO operations.
If you ever have to implement a file system in linux or you are going to implement a java class implementing the file interface you will notice that you will have to implement the following methods:

open()
close()
read()
seek()
tell()
write()

... of course there are a few others like "size()" and for handling access control.

currently FAL only supports open/close as making an instance of the "File" object could get seen as a call to open().

This issue suggests to add additional methods to File() which allows those basic operations.

For the LocalDriver a fopen()/filehandle in the constructor and wrappers for fread/fwrite/fseek/ftell would be sufficient.

Driver for S3 Amazon could for example manage internal position pointers and use the "Range" header to retrieve requested chunks from the storage (http://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectGET.html)

History

#1 - 2014-07-10 01:37 - Markus Klein

Does this method from the File class fulfill your needs?

```
/**
 * Returns a path to a local version of this file to process it locally (e.g. with some system tool).
 * If the file is normally located on a remote storages, this creates a local copy.
 * If the file is already on the local system, this only makes a new copy if $writable is set to TRUE.
 *
 * @param boolean $writable Set this to FALSE if you only want to do read operations on the file.
 *
 * @throws \RuntimeException
 * @return string
 */
public function getForLocalProcessing($writable = TRUE) {
    if ($this->deleted) {
        throw new \RuntimeException('File has been deleted.', 1329821486);
    }
    return $this->getStorage()->getFileForLocalProcessing($this, $writable);
}
```

If you set $writeable = FALSE you will get the absolute path to the file (if the LocalDriver is used).
You can do with this path whatever you like.

#2 - 2014-07-10 13:52 - Bernhard Kraft

This issue is marked as feature and could have. So its rather an idea/suggestion than a pressing requirement!

I don't like to handle FileObjects using different drivers specifically. Usually I try to write code which works on any driver.

The reason for this issue is just a proposal. For example when working with large files (CSV, etc.) this could be useful.

#3 - 2014-12-23 19:42 - Mathias Schreiber

- Target version changed from 7.0 to 7.1 (Cleanup)

#4 - 2015-06-15 17:08 - Benjamin Mack

- Target version changed from 7.1 (Cleanup) to 7.4 (Backend)