GFAL JAVA API

Diego Scardaci
INFN – Catania
Catania, JRA1 All-Hands meeting
7-9 March 2007

Information Society and Media

GFAL Java API

Grid File Acces Library (GFAL):

- Client POSIX (like) I/O library for directly data access.

Aim:

- Provide GFAL API to Java Developers.

Why Java?

- Reduce developing time;
- More Java developers are “available”;
- More applications are “pure Java” applications.

GFAL Java API Classes

- **GFalFile**: implements methods to manage remote files (stored on a SE).
- **GFalDirectory**: implements methods to manage SE remote directory.
- **GFalUtilities**: provides some important utility methods.

GFalFile: createFile

- **public int createFile(String name, int mode, boolean isSurl, boolean isLargeFile) throws GFalFileException**
  - name = the name of the storage element where the file will be created (e.g. aliserv6.ct.infn.it) or a SURL (e.g. srm://aliserv6.ct.infn.it/dpm/ct.infn.it/home/gilda/generated/2006-06-22/filetest);
  - mode = new file access permission (e.g. 644).
  - isSurl = “false” if the first parameter is a name of a storage element (SURL automatically generated); “true” if the first parameter is a a SURL.
  - isLargeFile = “true” if you want to create a large file (until 2^{63} bytes).

GFalFile: openFile

- **public int openFile(String fileName, int flags, int mode, boolean isLargeFile) throws GFalFileException**
  - name = the file name: lfn, guid, SURL o TURL.
  - flags = value is built by OR'ing the bits of the following constants:

    READONLY WRITEONLY CREAT LARGEFILE (if you're using the "CREAT" flag, the name parameter cannot be neither an LFN nor a GUID).
  - mode = access permission of the new file (e.g. 644).
  - isLargeFile = true if you want to open a large file.

GFalFile: other methods

public byte[] readFile(int size)	Reads size bytes from the file.
public int writeFile(byte[] buffer)	Writes buffer data in the file.
public long lseekFile(long offset, int seekMode, boolean isLargeFile)	Positions/repositions to offset the file associated with this object.
public int closeFile()	Closes the file opened by openFile method.
public int lfcRegisterFile(java.lang.String logical FileName)	Registers the file associated with this object in the lfc catalog. You have to close the file before to use this method.
public java.lang.String getSurl()	Returns the SURL of the file associated with this object .
public java.lang.String getLFN()	Returns the LFN of the file associated with this object .

GFalDirectory

public int openDir(String dirName)	Open a remote directory.
public String[] readDir()	Read the directory associated with this object.
public int closeDir()	Close the directory associated with this object.
public static int makeDir(String dirName, int mode)	Creates a new directory with permission bits taken from mode.
public static int rmDir(String dirName)	Removes a directory if it is empty.

GFalUtilities

public int accessFile(java.lang.String fileName, int mode)	Checks the existence or the accessibility of the file/directory path according to the bit pattern in amode using the real user ID.
public int chmodFile(java.lang.String fileName, int mode)	Change access mode of a file/directory.
public long[] statFile(java.lang.String fileName, boolean largeFile)	Gets information about a file or directory. This routine returns an array of long containing the following information: mode, nlink, uid, gid and size.
public long[] lstatFile(java.lang.String fileName, boolean largeFile)	lstatFile is identical to statFile except for symbolic links. In this case, the link itself is statted and not followed.
public int renameFile(java.lang.String oldName, java.lang.String newName)	Rename a file or a directory.
public int deleteFile(java.lang.String fileName)	Remove a file entry.

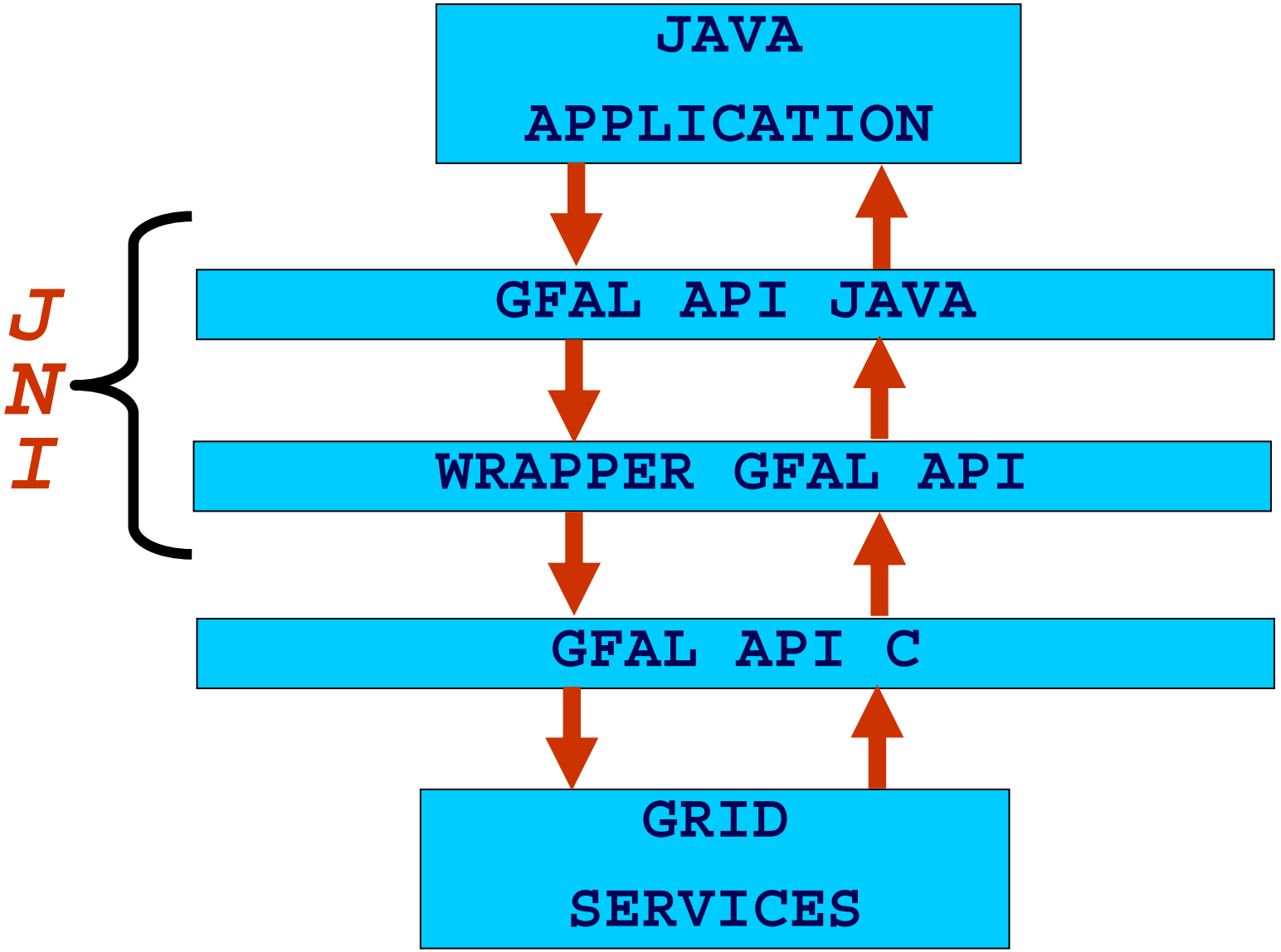
Environment Configuration

- **GFAL library: gfal.jar, libGFalFile.so e libgfal.so**
- **Environment Variables:**
  - LCG_RFIO_TYPE=<SE type> (e.g. LCG_RFIO_TYPE=dpm)
  - LCG_GFAL_VO=<your VO>
  - LCG_GFAL_INFOSYS=<your BDII>
  - LCG_CATALOG_TYPE=lfc
  - LFC_HOST=<LFC_HOSTNAME> (e.g. LFC_HOST=lfc-gilda.ct.infn.it)
  - LD_LIBRARY_PATH=<path libGFalFIle.so>:$LD_LIBRARY_PATH
  - CLASSPATH=<path gfal.jar>:$CLASSPATH
- **Needed Library: lcg_util, globus_ftp_client_gcc32, globus_gass_copy_gcc32**
- **Checks your proxy before to use the GFAL API!**

GFAL JAVA API: An example

```
import it.infn.catania.gfal.*;
public class testGFal {
   public static void main (String args[]) {
     String fileName = "aliserv6.ct.infn.it";
     try {
       GFalFile gFalFile= new GFalFile();
       gFalFile.createFile(fileName,644,false,false);

       byte[] dati = new byte[1024];
       gFalFile.writeFile(dati);

       gFalFile.closeFile();

       gFalFile.lfcRegisterFile("lfn:/grid/gilda/test/1.dat");

       gFalFile.openFile(gFalFile.getLFN(),GFalFile.READONLY,
       644,false);
       …
     }
     catch (GFalFileException exc) {
       exc.printStackTrace();
     }
  …
```

GFAL JAVA API



GFAL JAVA API: GFalFile Class

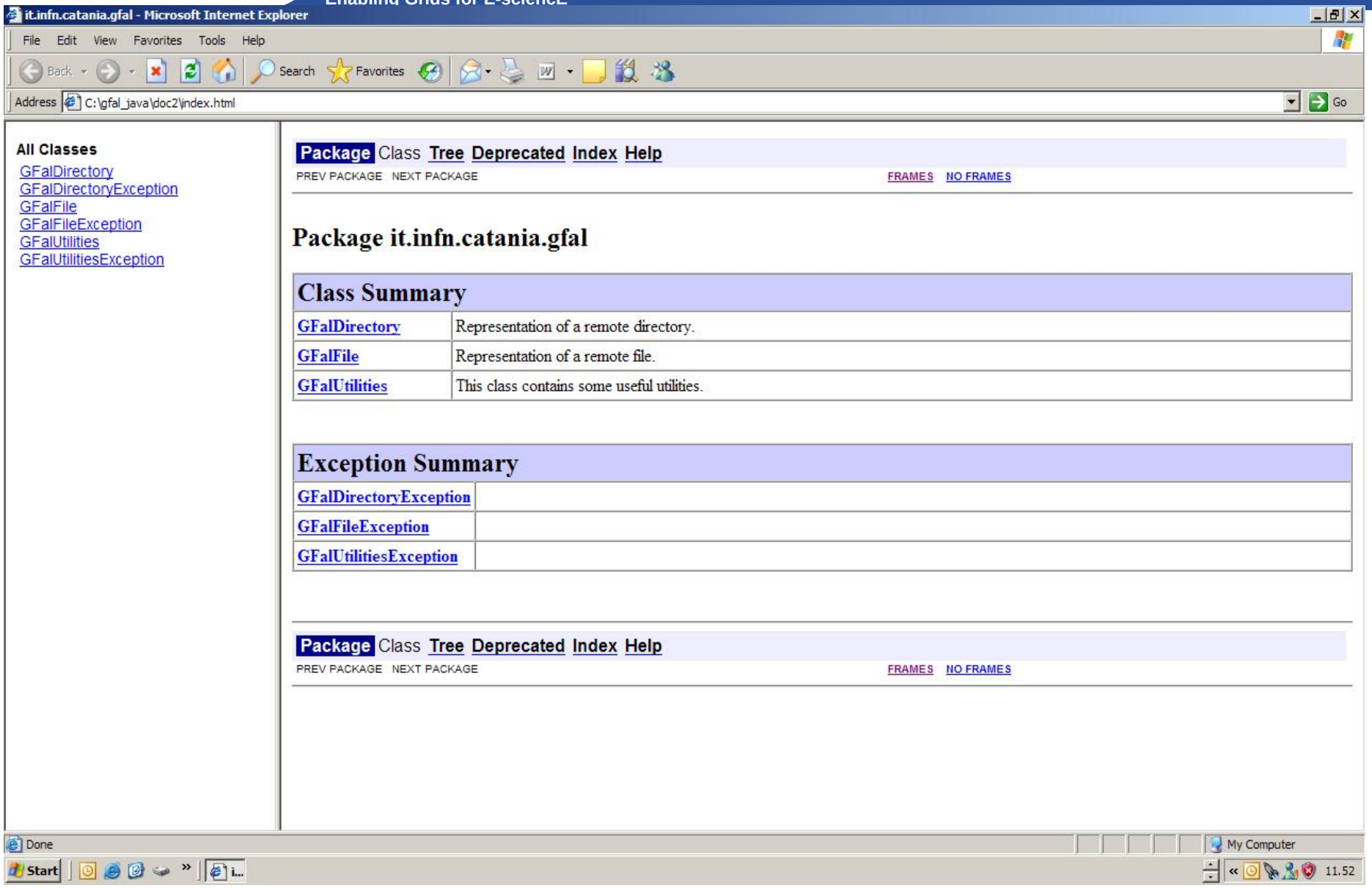
Java Class:

```
package it.infn.catania.gfal;
…
public class GFalFile
{
   static {
                System.loadLibrary("GFalFile");
   }
   …
   private native int openGFalFile(String fileName, int mode, int
   perm, boolean isLargeFile);
   private native int closeGFalFile(int fd);
   …
   public int openFile(String fileName, int flags, int mode, boolean
   isLargeFile) throws GFalFileException {
        …
        int ret = openGFalFile(fileName, flags, mode, isLargeFile);
        …
   }
}
```

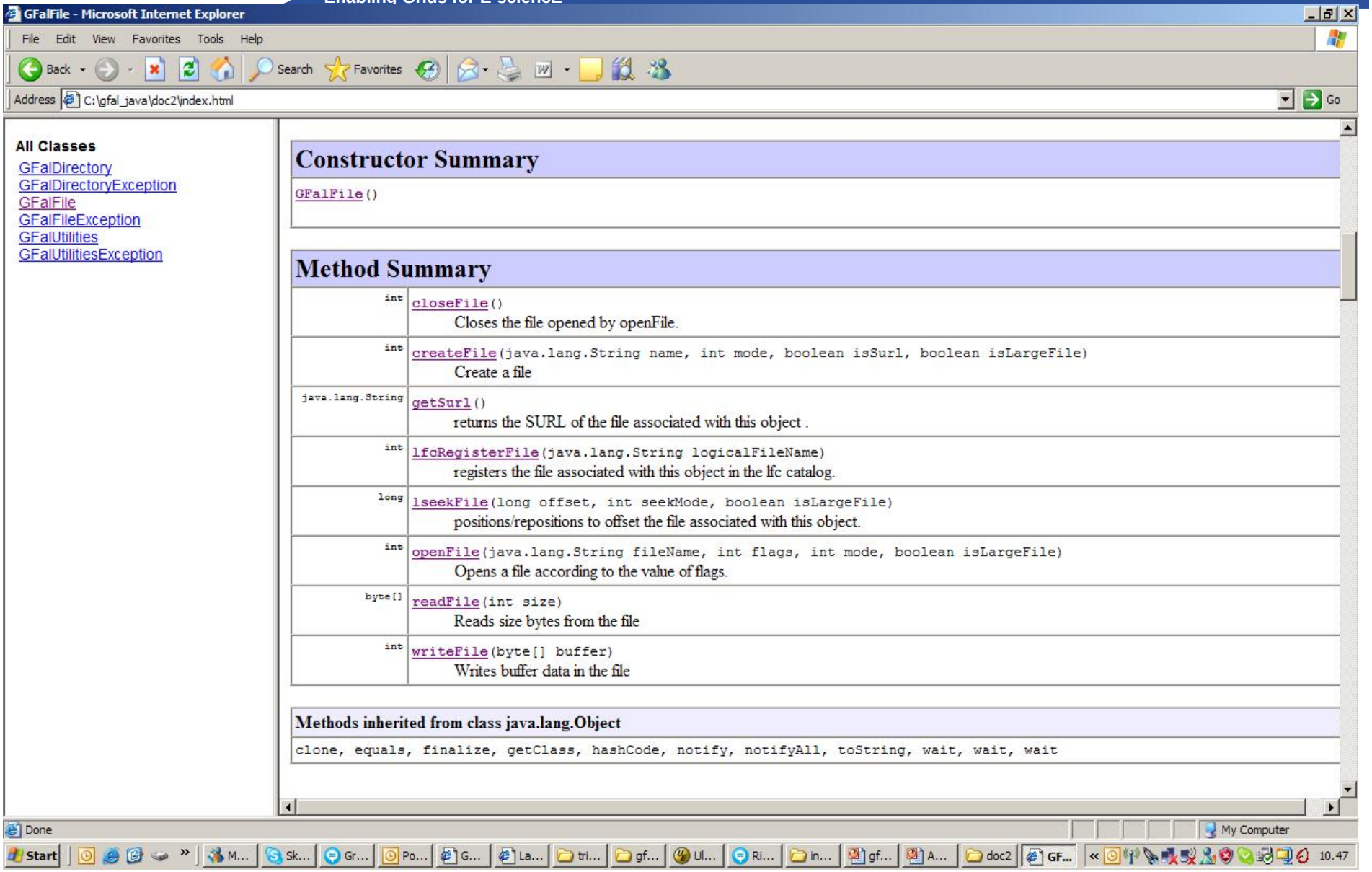
C Wrapper (libGFalFile.so):

```
#include <gfal_api.h>
#include <jni.h>
#include "it_infn_catania_gfal_GFalFile.h"
…
JNIEXPORT jint JNICALL
   Java_it_infn_catania_gfal_GFalFile_openGFalFile (JNIEnv *env,
   jobject obj, jstring fileName, jint mode, jint permission,
   jboolean largeFile)
{
   int fd;
   const char *str = (*env)->GetStringUTFChars(env, fileName, 0);
   if(largeFile)
        fd = gfal_open64(str, mode, permission);
   else
        fd = gfal_open(str, mode, permission);
   (*env)->ReleaseStringUTFChars(env, fileName, str);
   …//check errors
   return fd;
}
```

GFAL API: Javadoc



GFAL API: Javadoc



References

- **GFAL C API:**

http://grid-deployment.web.cern.ch/grid-deployment/gis/GFAL/GFALindex.html

- **GFAL Java API wiki page:**

https://grid.ct.infn.it/twiki/bin/view/GILDA/APIGFAL

- **GFal Java API Javadoc:**

https://grid.ct.infn.it/twiki/GFAL/

Any questions ?