

Esercitazione su Globus Toolkit 2:  
Librerie Java – 3ª Parte

Paolo Trunfio  
trunfio@deis.unical.it

1

## Staging dell'eseguibile

```
public class JobRun4 {
    public static void main(String [] args) {
        String contact = "jupiter.deis.unical.it";
        try {
            GassServer gassServer = new GassServer();
            String gassUrl= gassServer.getURL();
            String rsl = "&(rsl_substitution = (GLOBUSRUN_GASS_URL "+gassUrl+"))"+
                "(executable = $(GLOBUSRUN_GASS_URL)//bin/date)" +
                "(stdout="+gassUrl+"/dev/stdout)(stderr="+gassUrl+"/dev/stderr)";
            // esegue lo staging dell'eseguibile locale "/bin/date", e redirige l'output su video
            // equivale a: "globus-job-run jupiter.deis.unical.it -s /bin/date"
            GramJob job = new GramJob(rsl);
            job.addListener(new SimpleGramJobListener());
            job.request(contact);
        } catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

2

## Staging del file di output

```
public class JobRun5 {
    public static void main(String [] args) {
        String contact = "jupiter.deis.unical.it";
        try {
            GassServer gassServer = new GassServer();
            String gassUrl= gassServer.getURL();
            String rsl = "&(rsl_substitution = (GLOBUSRUN_GASS_URL "+gassUrl+"))"+
                "(executable = /bin/date)"+
                "(stdout=$(GLOBUSRUN_GASS_URL)//home/paolo/localOutput)";

            // equivale a:
            // "globus-job-run -stdout -s /home/paolo/localOutput jupiter.deis.unical.it /bin/date"
            GramJob job = new GramJob(rsl);
            job.addListener(new SimpleGramJobListener());
            job.request(contact);
        } catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

3

## Test di autenticazione con un server GridFTP

```
import org.globus.ftp.*;
public class GridFTPAuthTest {
    public static void main(String args[]) {
        String host = "jupiter.deis.unical.it";
        int port = 2811;
        try {
            GridFTPClient client = new GridFTPClient(host, port);
            client.authenticate(null);
            client.close();
            System.out.println("Authentication OK");
        }
        catch (Exception e) {
            e.printStackTrace();
            System.out.println("Authentication FAILED");
        }
    }
}
```

4

## Listing di una directory remota con GridFTP

```
import org.globus.ftp.*;
import org.globus.io.urlcopy.*;
import org.globus.util.GlobusURL;
import java.util.Vector;

public class ListDirectory {

    public static void main(String args[]) {
        String host = "jupiter.deis.unical.it";
        int port = 2811;
        try {
            GridFTPClient client = new GridFTPClient(host, port);
            client.authenticate(null);
            client.setDataChannelAuthentication(DataChannelAuthentication.NONE);
            client.setType(Session.TYPE_ASCII);
            String directory = "/home/paolo";
```

5

## Listing di una directory remota con GridFTP (cont.)

```
    // String directory = client.getCurrentDir(); /* home directory */
    client.changeDir(directory);
    Vector files = client.list();
    while (!files.isEmpty()) {
        FileInfo fileInfo = (FileInfo)files.remove(0);
        System.out.println(fileInfo.toString());
    }
    client.close();
}
catch (Exception e) {
    e.printStackTrace();
}
}
```

6

## La classe org.globus.ftp.FileInfo

**Dimensione del file in byte:**

- `public long getSize()`

**Nome del file:**

- `public String getName()`

**Data del file:**

- `public String getDate()`

**Modification time del file:**

- `public String getTime()`

**Restituisce true se è un file:**

- `public boolean isFile()`

**Restituisce true se è una directory:**

- `public boolean isDirectory()`

**Restituisce true se un soft link:**

- `public boolean isSoftLink()`

**Restituisce true se è un device:**

- `public boolean isDevice()`

7

## La classe org.globus.ftp.FileInfo (cont.)

**Informazioni sui diritti di accesso del file:**

- `public int getMode()`
- `public String getModeAsString()`
- `public boolean userCanRead()`
- `public boolean userCanWrite()`
- `public boolean userCanExecute()`
- `public boolean groupCanRead()`
- `public boolean groupCanWrite()`
- `public boolean groupCanExecute()`
- `public boolean allCanRead()`
- `public boolean allCanWrite()`
- `public boolean allCanExecute()`

**Restituisce una stringa contenente le principali informazioni sul file:**

- `public String toString()`

8

## Riferimenti

✓ **Java CoG Kit 1.1 home page:**

**<http://www-unix.globus.org/cog/java/1.1>**

- **Binary distribution**
- **Source distribution**
- **API documentation**

✓ **Manual user:**

**<http://www-unix.globus.org/cog/manual-user.pdf>**