

# How parameters, needed by castor, are retrieved in Castor v2.1

## 1) INTRODUCTION

In the following document it is described how the different parameters needed by the Stager are retrieved in the Castor v2.1.

It has been extended the TURL used and other modifications have been done to avoid not thread safe situations.

After the explanation of how the new TURL is, I'll analyse how the parameters are chosen in the different components.

The first analyse will be more from the user point of view and after I'll describe the implementation details.

### 1.1 The New TURL for files in CASTOR v2.1

The new TURL can carry much more information than the old one.

The old one was like "rfio://(host)(:port)/(path) and for castor file it was "rfio:///path) where path was like /castor/cern.ch/user/n/nobody/file.

With the new TURL a file castor can be specified in two different ways:

- i) 'rfio://(stagehost)(:port)/  
    ?(svcClass=MySvcClass&)  
    (castorVersion=MyCastorVersion&)  
    path=/castor/cern.ch/user/n/nobody/file'
  
- ii) 'rfio://(stagehost)(:port)/(/castor/cern.ch/user/n/nobody/file)  
    (?(svcClass=MySvcClass&)(castorVersion=MyCastorVersion))'

In this case I don't have as host the diskserver where the physical file is, but it is the stager host that has to be contacted to know the diskserver from which we can get the file.

The path must be specified as path in the parameters (e.g. i) or before the parameters (e.g. ii).

If we use the path option, it must be the last one given by the TURL. The order of the other options is not taken into account.

## 1.2 Parameters needed by Castor 2

The parameters needed are:

- the stager host (host where the stager is running)
- the stager port (port used to contact the Stager)
- the service class used (type of service asked by the user)
- the castor version (to know if the server is a castor v1 or castor v2 stager)

Before the information was retrieved just using environment variables and that could have lead to situations not thread safe.

Now let's analyse how are retrieved in castor 2.1.

## 2) How are parameters retrieved?

Let's see in the different components how we retrieve the information needed by Castor.

### 2.1 In RFIO api and command line ...

The different parameters can be given using the new TURL, but for all the missing

one we use the following procedure to decide the value.

First I took the information given by the environment variable "STAGE\_HOST", "STAGE\_PORT", "STAGE\_SVCCLASS" and "RFIO\_USE\_CASTOR\_V2".

If I still have some miss parameters I try to retrieve them from the stager mapper or after that from castor.conf file.

If any of the 4 values are still unknown then I use the following default values:

- stage host = stagepublic
- stage port = 9002 (for castor 2) or 5007 (for castor 1)
- stage svc class= "" (the default one)
- castor version =1

Of course the old rfio TURL is still valid and the missing information is retrieved as missing parameters.

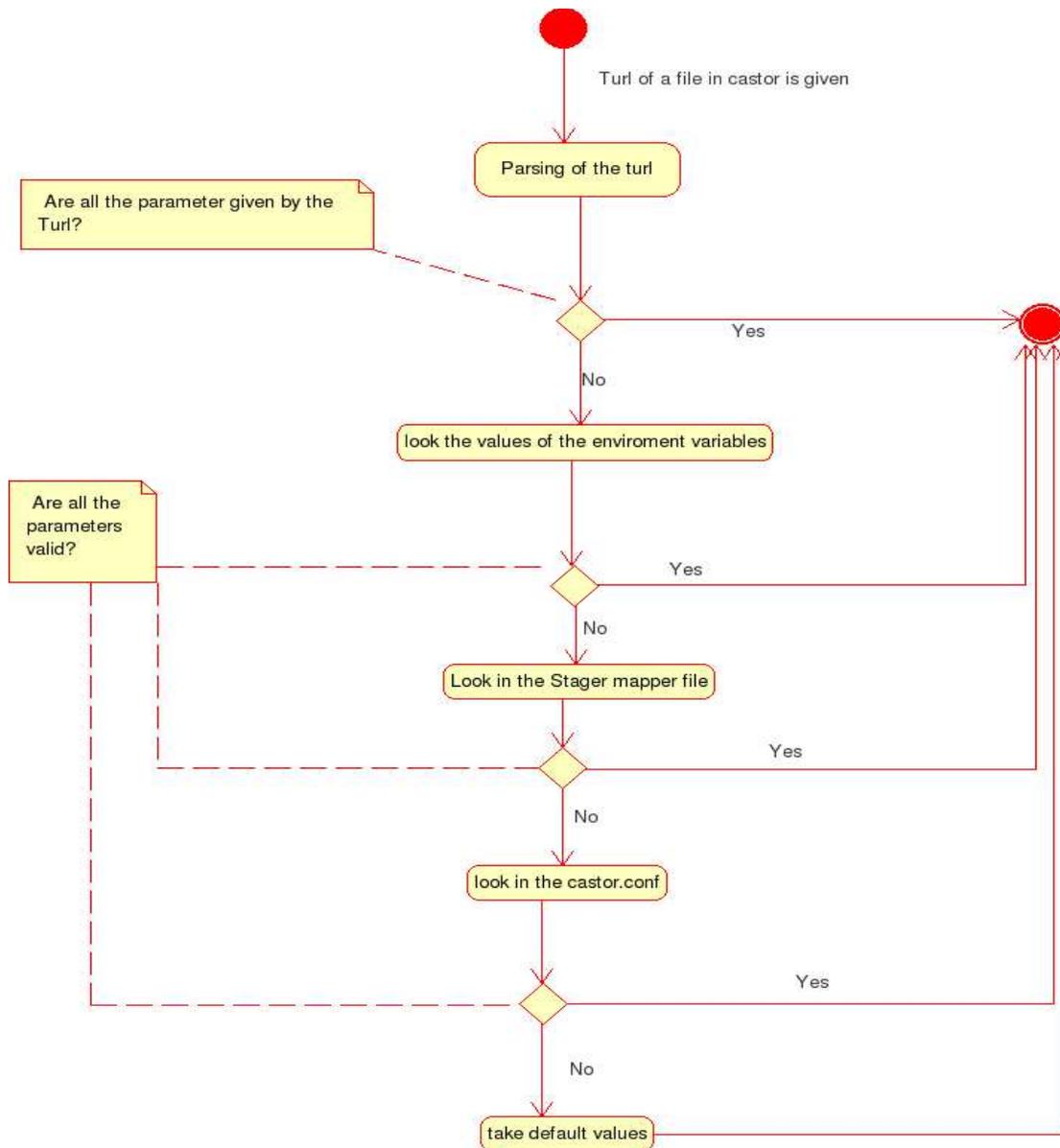


Figure 2.1, In Rfio command line and api.

## 2.2 In Root framework ...

Also in Root framework to use the old syntax will be possible :

- rfio://host:path
- rfio:host:path
- castor://pat
- castor:path

As we see the TURL used inside Root was different from the one used by Rfio api and command line.  
To be able to be backward compatible a restriction on the new TURL has been put:  
the path for the new TURL can be given only as parameter.  
We can have:

- 'rfio://(stagehost)(:port)/  
    ?(svcClass=MySvcClass&)(castorVersion=MyCastorVersion&)  
    path=/castor/cern.ch/user/n/nobody/file'
- 'rfio:(stagehost)(:port)/  
    ?(svcClass=MySvcClass)(&castorVersion=MyCastorVersion&)  
    path=/castor/cern.ch/user/n/nobody/file'
- 'castor://(stagehost)(:port)/  
    ?(svcClass=MySvcClass)(&castorVersion=MyCastorVersion&)  
    path=/castor/cern.ch/user/n/nobody/file'
- 'castor:(stagehost)(:port)/  
    ?(svcClass=MySvcClass)(&castorVersion=MyCastorVersion&)  
    path=/castor/cern.ch/user/n/nobody/file'

In the new TURL, the parameter path is mandatory and it must be the last one between parameters. The order of the other ones is not important.  
If the information carried by the TURL is not enough to retrieve the missing information needed by castor, we follow the procedure we saw for RFIO api.

### **2.3 In the Stager command line ...**

The stager command line is only for Castor 2, for Castor 1 there is the stage one.

The castor version is always 2.

For the other parameters, if I don't have any extra information given as options then I use the following procedure:

First I try to choose the values looking the environment variable "STAGE\_HOST", "STAGE\_PORT" and "STAGE\_SVCCLASS".

If I have still some missing information, we look first in the stager mapper and after in castor.conf.

If there are still some unknown values then I take the following default:

- stager host = stagepublic
- stager port = 9002
- stager svc class= "" (the default one)

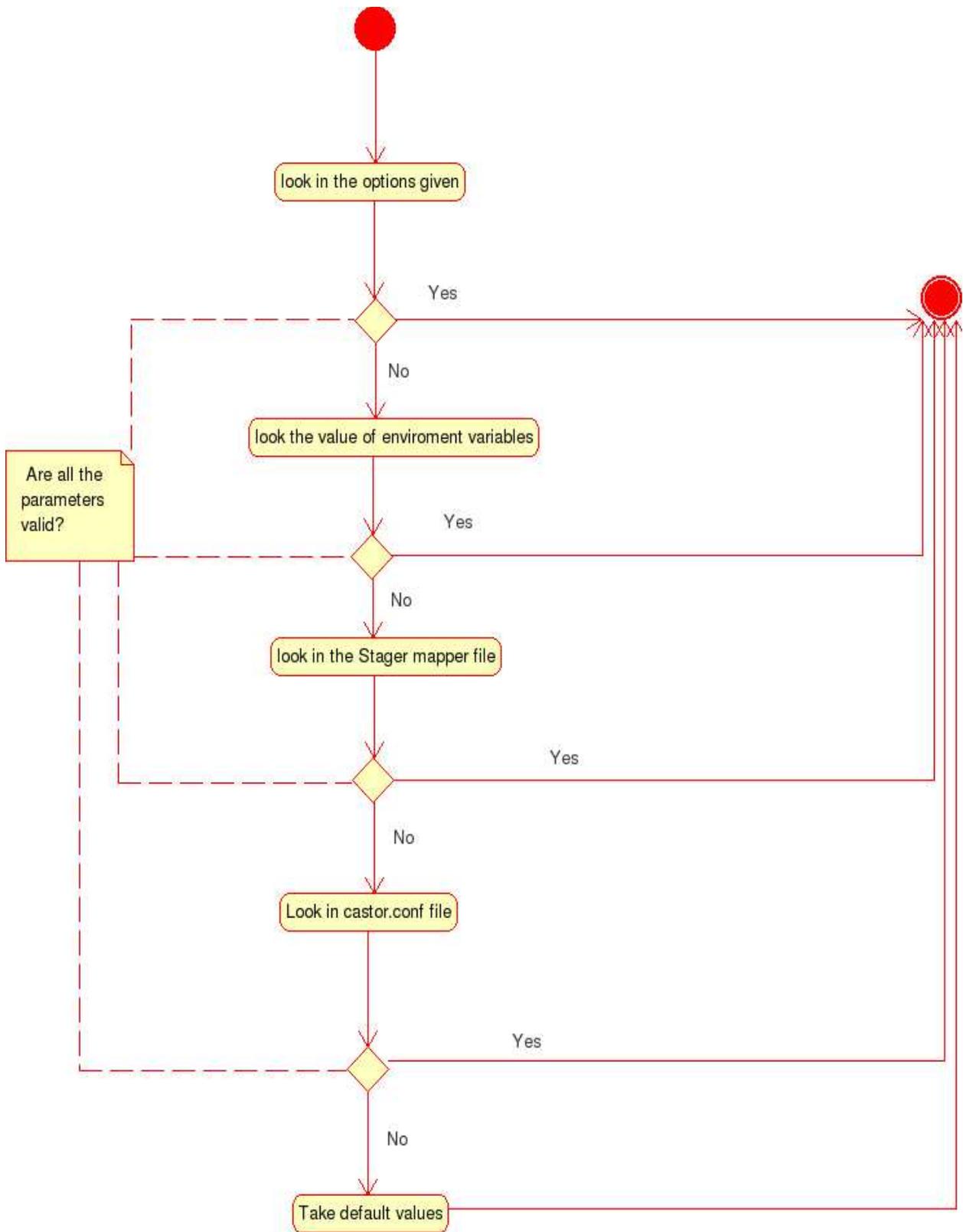


Figure 2.2, In the Stager command line.

## 2.4 In the Stager api ...

The stager api is peculiar of castor 2, for castor 1 we have the stage api, then castor version is always 2.

If the value of stager host, stager port and stager service class are not given as option we use the following default:

- stager host = stagepublic
- stager port = 9002
- stager svc class= "" (the default one)

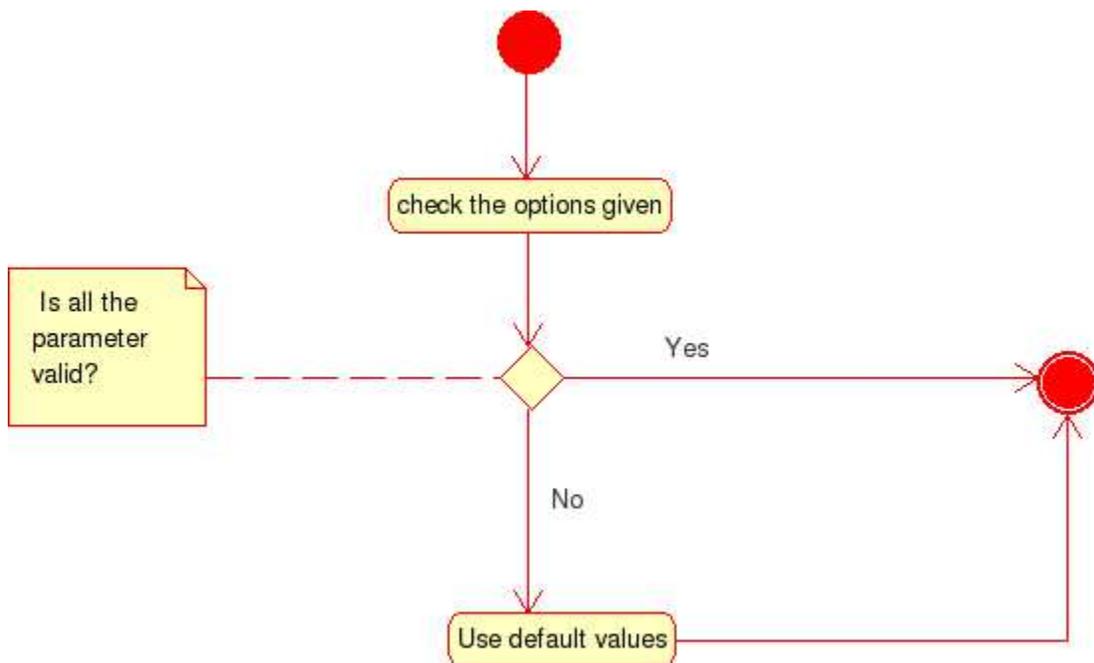


Figure 2.3, In the stager api

## 2.5 In the Request Handler ...

The request handler class is part of castor 2 then the castor version can be only 2.

To retrieve the value of stager host, stager port and stager service class, we look first if they are given as options, otherwise we apply the follow

procedure.

We look for environment variables (RH\_HOST, RH\_PORT, STAGE\_SVCCLASS) first and if there are still unknown values, we use the default:

- stage host = stagepublic
- stage port = 9002
- stage svc class= "" (the default one)

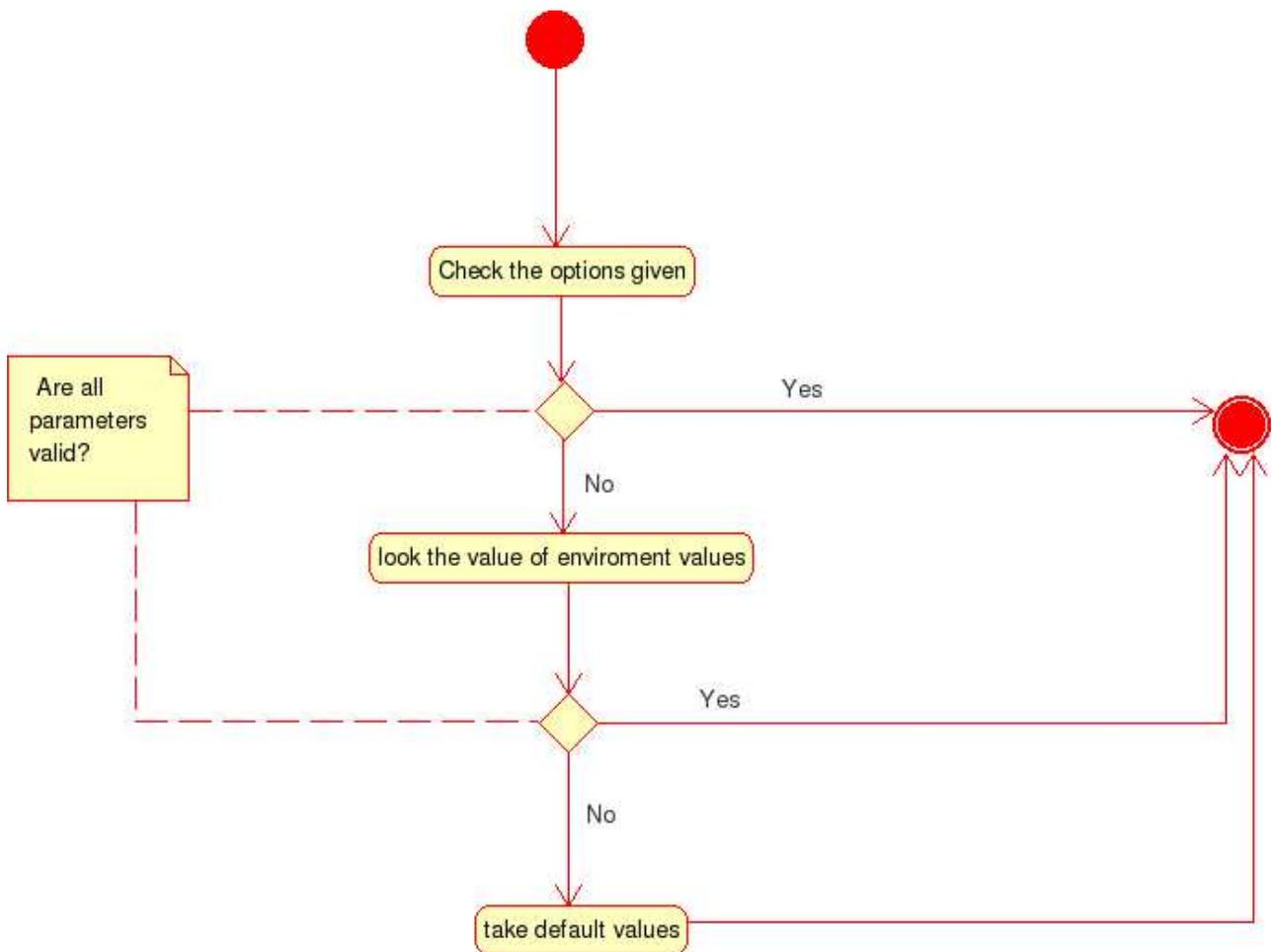


Figure 2.4, In the Request Handler.

### 3) Implementation details

Let's analyse the modifications done in a more detailed way.

#### 3.1) Rfio Api and command line

files involved:

- h/rfioTURL.h
- rfio/rfioTURL.c
- rfio/parse.c
- rfio/rfcp.c
- rfio/rfrm.c
- rfio/rmdir.c
- rfio/rfio\_Hsmlf.c
- h/stager\_mapper.h
- stager/stager\_mapper.c

In rfioTURL has been defined the new function, int getDefaultForGlobal(char\*\* host,int\* port,char\*\* svc,int\* version), which retrieves the stager host, the stager port, the stager service class and the castor version, if the inputs have the following values:

- \*host = null
- \*port = 0
- \*svc = null
- \*version=null

This function is used in rfioTURLFromString() function to retrieved the right values if they are not given in the TURL.

The "rfioTURLFromString()" initializes also the global variables through Cglobal\_get (keys used: tStageHostKey,tStagePortKey,tSvcClassKey,tCastorVersionKey) if I'm working with a castor file (the path must start with '/castor').

rfioTURLFromString() is the function used by the parsing functions defined in parse.c.

The information from the stager mapper are retrieved thanks to int just\_stage\_mapper(const char \*username, const char \*groupname,char \*\*mstager,char \*\*msvcclass,int \*isV2) defined in stager\_mapper.

This function does the same of stage\_mapper\_setenv without setting environment variables.

Rfio api uses host-port-serviceClass-version retrieved as options for the stager api.(opts not NULL anymore and all the field of the struct MUST be initialized).

The rfcpl has been modified because there is not a rfio\_rfcpl in the rfio api for that.

The rfrm has been modified for some problems to deal with the new TURL when there are directories to be removed.

### **3.2) Root Framework**

file involved:

- rfio/inc/TCastorFile.h
- rfio/src/TCastorFile.cxx
- rfio/inc/TRFIOFile.h
- rfio/src/TRFIOFile.cxx

In TRFIOFile class there are some modifications needed to be able to use the new TURL but it is used rfio\_parse to deal with the parsing of the new TURL.

In TCastorFile class the parsing is done by TCastorFile::ParseAndSetGlobal() which has been written to be able to deal with the new TURL.

### **3.3) Stager command line**

file involved:

- stager/stager\_put.c
- stager/stager\_get.c
- stager/stager\_putdone.c
- stager/stager\_rm.c
- stager/stager\_qry.c

All these files call the getDefaultForGlobal function of rfioTURL.c to set host, port, service class and version if it is not given by options.

All these values are given as stage\_option to the stager\_api.

### 3.4) Stager api

files involved:

- h/stager\_client\_api\_common.h
- stager/stager\_client\_api\_get.cpp
- stager/stager\_client\_api\_put.cpp
- stager/stager\_client\_api\_query.cpp
- stager/stager\_client\_api\_common.cpp
- stager/stager\_client\_api\_rm.cpp
- stager/stager\_client\_api\_setFileGCWeight.cpp
- stager/stager\_client\_api\_update.cpp
- stager/stager\_client\_api\_next.cpp

In client\_api\_common is defined a new function :

```
int DLL_DECL setDefaultOption(struct stage_options* opts).
```

If opts contains has fields with a null value, it initializes it with a default one (host=stagepublic port=9002 svcclass="").

When this function returns -1 it's NOT an error status but the parameter opts was null and at the end is necessary to call a free(opts) against memory leak.

After "setDefaultOption" is called, opts it is used in

RequestHelper::setOptions(opts) to set the svc class in the request and in BaseClient::setOption(opts).

### 3.5) Request Handler:

files involved:

- castor/client/BaseClient.hpp
- castor/client/BaseClient.cpp
- castor/stager/RemoteGCSvc.cpp
- castor/stager/RemoteJobSvc.cpp

A field with the request handler svc class std::string m\_rhSvcClass has been added to the class Base Client.

The setRhHost, setRhPort, setRhSvcClass can be called giving a value or with no parameters, in that case are used environment variable or default values to set the fields of the Client.

It has been defined also setOption which use the parameter stage\_options\* opts to set host, port and service class with the information

given or with values taken from environment variables or with default values.

The setOption must be called to initialize the client, because of that castor/stager/RemoteGCSvc.cpp and castor/stager/RemoteJobSvc.cpp have been changed by adding the call to the setOption().

#### **4) Summary of how information needed by castor is retrieved:**

<stage host,stage port,stage svcclass, castor version>

##### **rfio api**

from the new TURL

=> environment variables

(STAGE\_HOST,STAGE\_PORT,STAGE\_SVCCLASS,RFIO\_USE\_CASTOR\_V2)

=> stager mapper

=> castor.conf

=> default <stagepublic,5007,"",1> or, if castor version=2,

<stagepublic,9002,"",1>

##### **root**

from the new TURL

=> environment variables

(STAGE\_HOST,STAGE\_PORT,STAGE\_SVCCLASS,RFIO\_USE\_CASTOR\_V2)

=> stager mapper

=> castor.conf

=> default <stagepublic,5007,"",1> or, if castor version=2,

<stagepublic,9002,"",1>

##### **stager cmd line**

Castor version is always 2.

from options

=> environment variables

(STAGE\_HOST,STAGE\_PORT,STAGE\_SVCCLASS)

=> stager mapper

=> castor.conf

=> default <stagepublic,9002,"",2>

## **stager api**

Castor version is always 2.

from options

=> default <stagepublic,9002,"",2>

## **request handler**

castor version is always 2.

from options

=> environment variables (RH\_HOST, RH\_PORT, STAGE\_SVCCLASS)

=> default <stagepublic,9002,"",2>