Table of Contents

File Transfer Service support area

The purpose of this page is to keep track of the problems and support requests posted to GGUS.

This page is relevant to the gLite FTS 1.4.1 and FTS 1.5 release, and most of them to the FTS 2.0 release.

Todo: split these out onto separate pages. At least for FTS 2.0.

Summary

Configuration

- YAIM Configuration explained

File Transfer Service

- My DN changed. Could you please grant me the same priveleges I had before?
- I tried to submit a job and it said: submit: You are not authorised to submit jobs to this service
- I submitted a job from site X to Y but it didn't work. The channel Y-X exists and has a share for my VO!
- Which format should I use for the SURLs?
- I've tried to submit a job but I get back an error saying: SOAP-ENV:Server.userException - org.xml.sax.SAXException
- I've tried to submit a job but I get back an error saying: No match
- I was able to list the channels but I cannot get the channel details
- How do I setup a non-dedicated Channel?
- After upgrading to FTS 1.5 I got "No Channel found or VO not authorized ..." error
- My jobs fail if I have a short time left on the proxy in MyProxy

File Transfer Agent

- Job always in Submitted state
- Job always in Pending state
- All my transfers fail with a SECURITY_ERROR
- Which MyProxy Server is used?
- "Error in bind()" from MyProxy server
- I've noticed a warning "Cannot Get Agent DN" in the agent log files
- My srmcopy transfers fail with a dCache MalformedUrl exception
- I've upgraded to 1.4.1 but srmcopy doesn't seem to work
- I've upgraded to 1.4.1 but the transfer failed with Error in srm__ping: NULL
- The transfer failed with the error: No site found for host ...
- The transfer failed with the error: an end-of-file was reached
- Which Service Types are used?
- I've tried everything, and it still doesn't seem to work

Channel Administration

- How do I set the number of files transferred per VO instead of per channel?

Discovery Service

- "No site for host" error
- "No channel found, channel closed for your VO..." error

- "No SRM method factory found" error

Configuration

YAIM Configuration explained

Starting from FTS version 1.5, the configuration has moved from the gLite python configuration script to YAIM (you can find an example in `/opt/glite/yaim/example/site-info.def`). For the Yaim details, please refers to the the related documentation. The relevant part for us is the FTS and the FTA one. See FtsYaimValues15

File Transfer Service

My DN changed. Could you please grant me the same privileges I had before?

In case a user DN changed, for example because of the change of the CERN CA, all his/her privileges on the FTS Server should be updated. If the old certificate is still valid, the user can perform this operation by his own, without the help of the FTS amdinistrator. In order to due that, the user has to execute the following steps with a valid proxy generated from the old certificate:

- Invoke `glite-transfer-getroles` to retrieve the list of priviledges
- For each channel he/she has the management provileges on, execute

      glite-transfer-channel-addmanager CHANNEL_NAME NEW_DN

- For each VO he/she has the management provileges on, execute

      glite-transfer-addvomanager VO_NAME NEW_DN

In case the old user's certficate expired, the FTS administrator has to list all managers of all the channels (`glite-transfer-channel-listmanagers`) and VOs (`glite-transfer-listvomanagers`) and then executes `glite-transfer-channel-addmanager` and `glite-transfer-addvomanager` as above.

The user can then check that the privileges are correct by executing `glite-transfer-getroles` with a proxy generated from the new certificate.

In case the user is also and FTS administrator, the file `/opt/glite/etc/glite-transfer-admin-mapfile` should be manually modified in every node where the FTS-WS is installed and a new entry corresponding to the new DN soudl be added.

When the old certificate expires or is no longer needed, the user should then remove the priviledges granted to the old DN by executing the following commands, with a proxy generated from the new certificates:

- Invoke `glite-transfer-getroles` to retrieve the list of priviledges
- For each channel he/she has the management provileges on, execute

      glite-transfer-channel-removemanager CHANNEL_NAME OLD_DN

- For each VO he/she has the management provileges on, execute

      glite-transfer-removevomanager VO_NAME OLD_DN

Symptom: I tried to submit a job and it said: `submit: You are not authorised to submit jobs to this service`

The user is not authorised to submit jobs to the FTS service. In order to authorize him/her, you have to add his/her DN in the `submit-mapfile` on the FTS server. You can have a look at FtsServerInstall in the `Mapfile` section and at FtsServerSubmitMapfile

However, due to bug in the FTS (#10362⧉), if the user has a double or more delegated proxy (i.e. the DN ends with `/CN=proxy/CN=proxy`), a parsing error will cause a authorization denied. This bug has being solved in FTS version 1.4 and in the latest QuickFix for 1.3

If the user is still not authorized to submit request, check his/her DN is not in the `veto-mapfile`

Symptom: I submitted a job from site X to Y but it didn't work. The channel Y-X exists and has a share for my VO!

From version 1.3 onwards the channel definitions are mono-directional. You have to create another channel in the opposite direction (`glite-transfer-channel-add`), set the share for the VO interested in using the channel (`glite-transfer-channel-setvoshare`) and install an Channel Agent that will managed it

Which format should I use for the SURLs?

Starting from gLite 1.4.1, the FTA implements the enhancement request #8364⧉, that allows a user to specify any format he prefers: the agent would then convert each SURL before transfering or registering into the catalog to either a fully qualified format

```
srm://<host>:<port>/srm/managerv1?SFN=<file_path>
```

or a compact one

```
srm://<host>/<file_path>
```

depending on the configuration. By default it would use the compact format. In case you want to change this parameter, you have to set the related ChannelAgent configuration parameter `ACTIONS_SURLNORMALIZATION` (`transfer-agent-channel-actions.SurlNormalization`) to one of the following values:

- `compact` all the SURLs will be converted to the format:

  ```
  srm://<host>/<file_path>
  ```

- `compact-with-port` all the SURLs will be converted to the format:

  ```
  srm://<host>:<port>/<file_path>
  ```

- `fully-qualified` all the SURLs will be converted to the format:

  ```
  srm://<host>:<port>/srm/managerv1?SFN=<file_path>
  ```

- `disabled` no SURL convertion will be performed

If you're using a previous version, for interoperability reasons we suggest to use fully qualified SURLs, i.e. in the format

```
srm://<srm_host>:<srm_port>/srm/managerv1/?SFN=<file_path>
```

If you know the type of the SRM that would be involved in the transfer, you can also specify one of the supported compact format. For Castor, as example, you can use

```
srm://<castorsrm>:8443/srm/managerv1?SFN=<file_path>
srm://<castorsrm>:8443//srm/managerv1?SFN=<file_path>
srm://<castorsrm>:8443/?SFN=<file_path>
srm://<castorsrm>:8443/<file_path>
srm://<castorsrm>/<file_path>
```

In case the transfer is processed by a channel configured to use `srmcopy`, the fully qualified format may not work. Please have a look here for a workaround

Symptom: I've tried to submit a job but I get back an error saying: SOAP-ENV:Server.userException - org.xml.sax.SAXException

Usually this issue is related to an endpoint pointing to the wrong server (typically `ChannelManagement` instead on `FileTransfer`): when you observe an error similar to

```
submit: SOAP fault: SOAP-ENV:Server.userException -
org.xml.sax.SAXException: Deserializing parameter 'job':  could not find deserializer for type {h
```

please ask the user to look at the command he just submitted and to check that the specified endpoint is correct; all the CLIs commands that start with `glite-transfer-channel-*` require to use a `ChannelManagement` interface, while the ones that start with `glite-transfer-*` require the `FileTransfer` interface. In order to check if the endpoint is correct, the user can also re-run the command with the `-v` option and checks if the line `Using Endpoint` ends with `FileTransfer` or `ChannelManagement`

Symptom: I've tried to submit a job but I get back an error saying: No match

When the user submit a transfer job, he usually specify some SURLs that may contains a question mark (`?`). In some shells this character has to be escaped by simply quoting it (`'?'`): for example, if the SURLs are

```
srm://castorgridsc.cern.ch:8443/srm/managerv1?SFN=/castor/cern.ch/grid/dteam/src_file
srm://castorgridsc.cern.ch:8443/srm/managerv1?SFN=/castor/cern.ch/grid/dteam/dst_file
```

please make sure you run `glite-transfer-submit` in this way

```
glite-transfer-submit \
    srm://castorgridsc.cern.ch:8443/srm/managerv1'?'SFN=/castor/cern.ch/grid/dteam/src_file \
    srm://castorgridsc.cern.ch:8443/srm/managerv1'?'SFN=/castor/cern.ch/grid/dteam/dst_file
```

Symptom: I was able to list the channels but I cannot get the channel details

Listing channels is open to any user as long as he/she is not in the veto mapfile - you only get the channel name from this call.

However, getting the details of a channel - source, destination, bandwitch, etc is restricted. For this you need to be:

- an admin
- manager of the channel being queried
- manager of any VO on the given FTS

You can check your roles on a given FTS by running `glite-transfer-getroles`. Information on channel and VO managers can be managed by a service admin or other managers by using the appropriate client tools. Information on service ADMINs is stored inside the admin-mapfile.

How do I setup a non-dedicated Channel?

Non-dedicated channels (a.k.a. "catch-all" channels) are a special channel configuration that allows matching any site as source or destination, therefore not coupled with the underlying network. Using "catch-all" channels allows to limit the number of channels you need to manage, but also limits the degree of control you have over what is coming into your site (although it still provides the other advantages like queueing, policy enforcement and error recovery). The usage of these channels is mainly recommended in Tier1 for providing full connectivity to all other sites, where the suggested channels definition is:

- Dedicated channels from any other Tier1 to the T1
- Non-dedicated channels to each of the related Tier2
- A non-dedicated channel to the T1

You can setup a non-dedicated channel that will manage all the transfers from any site to your site by issuing a `glite-transfer-channel-add` and using `*` and source site name, like:

```
glite-transfer-channel-add -f NUM_OF_FILES -S CHANNEL_STATE [...] CHANNEL_NAME "*"
YOUR_SITE
```

Of course, you have then to issue a `glite-transfer-channel-setvoshare` for each VO that should be authorized to use the channel and then configure a ChannelAgent for that channel.

Please note that is a VO is not authorized to use a channel between site `A` and `B` but has privileges on a `*-B` channel, transfer requests for that VO from site `A` to `B` are denied since the non-dedicated channel is evaluated *after* all the dedicated ones.

In addition, please also note that the default ChannelAgent configuration for that channel requires that all the SRM that would be involved in the managed transfers should be listed in the information system. In case a VO needs to relax this constraint, for example in order to transfers files to/from Classic SEs not included in the information system, the following parameters should be added to the VOAgent configuration:

- `ACTIONS_ENABLEUNKNOWNSOURCE` (`transfer-agent-vo-actions.EnableUnknownSource`) should be set to `true` if SEs not known to the InfoSys should be allowed as valid source (these would be matched by the `*-Site` catch-all channels)
- `ACTIONS_ENABLEUNKNOWNDEST` (`transfer-agent-vo-actions.EnableUnknownDest`) should be set to `true` if SEs not known to the InfoSys should be allowed as valid destination (these would be matched by the `Site-*` catch-all channels)

In case a VO needs these parameters, it would be better to turn off the SURL Normalization, or at least set it to `fully-qualified`, for all the ChannelAgents associated to non-dedicated channels, since it would be impossible to resolve the correct endpoint for the SRM not listed in the InformationSystemOverview. It will also be worth to reccommend the users to use fully-qualified SURLs for transfers that should be processed through these channels.

Use of the `*-*` 'catch everything' channel is not recommended for production grids.

Symptom: I was able to list the channels but I cannot get the channel details   5

Symptom: After upgrading to FTS 1.5 I got "No Channel found or VO not authorized ..." error

Running the FTS service we encountered many inconsistencies in the way the information was published in BDII, especially related to the case used to publish the site name. This not not a probalem when BDII is used directly, since it's is case insensitive, but creates some intereoperability issues when used via ServiceDiscovery (that is case sensitive). We therefore decided to apply a convention, within the FTS boundaries, in order to have all the site names uppercase in the channel definitions. Starting form version 1.5, the FTS WebService forces the case when you create a new channel, but when upgrading from previous versions, this convention may conflict whit already defined channels. In order to fix this, we have provided an admin pack hat allows changing the channel definitions. The instruction how to use that tools are available here.

Therefore, if you hit this problem, download the `glite-data-transfer-scripts` RPM and follow the instuction reported above in order to replace all the site names that contains lowercase letters in all the channel definition (you may need the support of your DBA).

Note: If this RPM is not yet available in the repository, please contact fts-support.

Symptom: My jobs fail if I have a short time left on the proxy in MyProxy

Make sure you have a fresh version in MyProxy that will last at least the length of all your jobs (assume queue length of 2 days from your last submission).

File Transfer Agent

Symptom: Job always in Submitted state

The first action that is executed on a transfer request is the Allocation, performed by the VO agent associed with the VO of the submitter. This actions checks the source and destination SURLs of the job request, find the sites of the involved SEs using ServiceDiscovery and then look up in the registered channels for a matching. When this operation succeed, the job is moved to Pending and the `channel_name` property is filled with the name of the found channel.

Due to a bug in FTA 1.3 and 1.4 (#10076⧉) a job stays in Submitted state instead of going to Failed in one of the following cases

- The channel doesn't exist but the source and destination SE are registered in ServiceDiscovery or the VO is configured to accept unknown source and destination
- The VO of the user who submitted the job has no valid share on the channel
- The channel is in Stopped, Drain or Halted (actually, when the channel status is Halted, a job should go in Pending and not in Failed)

Usually this problem is due to a configuration error. The first thing to do is to retrieve the status of the channel that should be involved in the transfer

```
glite-transfer-channel-list CHANNEL_NAME
```

check the channel state, that the VO has a share and that the names of the source and destination sites match the ones retrived using ServiceDiscovery: in case the file plugin is used, look at the `site` element of the SRM services reported into the `services.xml` file

```
<service name='CERNSC3-SRM'>
  <parameters>
```

```
    <endpoint>httpg://castorgridsc.cern.ch:8443/srm/managerv1</endpoint>
    <type>SRM</type>
    <version>1.1.0</version>
    <site>CERN-SC</site>
    <param name='SEMountPoint'>/castor/cern.ch/grid/dteam/storage</param>
  </parameters>
</service>
```

and compare them with the value returned by `glite-transfer-channel-list`

In case this doesn't fix the problem, check that a VO agent is configured and running for that VO. Do

```
glite-transfer-status --verbose JOB_ID
```

And check that the value of the `VOName` property is correct; in case is not, it's a problem with the FTS `glite-data-transfer-submit-mapfile`: edit that file manually or regenerate it following teh procedures reported by FtsServerSubmitMapfile, cancel the job, wait that the files is reloaded by the FTS and ask the user to resubmit the request.

In case the VO is set correctly, check on the agents node that an agent is configured:

- if you're using gLite 1.3, please have a look at `/opt/glite/etc/config/glite-data-transfer-agents-oracle.cfg.xml` and see if there is an instance for the VO:

```
<instance name="YOUR_VO-fts">
  <parameters>
    <transfer-vo-agent.Name value="YOUR_VO"/>
    <!-- Other parameter -->
    <!- ... -->
  </parameters>
</instance>
```

- if you're using gLite 1.4, open the file `/opt/glite/etc/config/glite-file-transfer-agents-oracle.cfg.xml` and look for an instance:

```
<instance name="YOUR_VO" service="transfer-vo-agent-fts"/>
```

If the instance is missing, or the naming convention is not correct, edit the appropriate file and rerun the configuration script.

If the instance is there, check if it's running, using the command

```
/opt/glite/etc/init.d/glite-data-transfer-agents --instance
glite-transfer-vo-agent-YOUR_VO status
```

or

```
service transfer-agents --instance glite-transfer-vo-agent-YOUR_VO status
```

(was `service glite-data-transfer-agents ...` before 1.5)

If the job is still Submitted, follow the procedure reported here

Symptom: Job always in Submitted state   7

Symptom: Job always in Pending state

After the a transfer request is allocation to a channel, its status is moved to Pending. The ChannelAgent will then process this request based on its internal inter-VO scheduling.

In case the job state remaing Pending forever, you have to check the follwoing things:

- The related ChannelAgent daemon should be running
- The Channel state should be set to Active
- The VO should have a share on the channel that is greater than 0

In order to check if the agent is running, use the command

```
/opt/glite/etc/init.d/glite-data-transfer-agents --instance
glite-transfer-channel-agent-TYPE-CHANNEL_NAME status
```

or

```
service transfer-agents --instance glite-transfer-channel-agent-TYPE-CHANNEL_NAME status
```

(was `service glite-data-transfer-agents ...` before 1.5)

You can check the Channel state and VO share using the command:

```
glite-transfer-channel-list CHANNEL_NAME
```

In case the job are still Pending and the FTS version is less than `2.0`, you may need to check if there are FTS transfer process alive. In fact, it may happen that due to network problem, some of these processes don't complete correctly or die unexpectedly, leaving the related log files in `/var/tmp/glite-url-copy-edguser` and wasting transfer slots. If that is the case, you have to stop the related channel agents, kill the "zombie" processes and cleanup the transfer log files for the involved channels. Once, you'll restart the channel agents, they will detect the abnormal termination of the transfers and the VO agents will reschedule them according to the configured retry policy

If the job is still Pending, follow the procedure reported here

Symptom: All my transfers fail with a SECURITY_ERROR

This issue is usually due to a problem in the interaction from a FTA and the MyProxy server. This mainly happens in the following cases:

- User is mistyping the MyProxy passphrase when submitting the job
- User has an invalid or expired certificate in MyProxy
- The agent is not an authorized retrieves for MyProxy
- There is a authentication problem (expired certificate or crl)

In the first two cases, all the transfers of this user should fail while the ones of other users succeed, while in the others all the transfers would faild, indipendently of the user.

Usually, you can detect the type of the error by having a look at the agent log file in
`/opt/log/glite/glite-transfer-channel-agent-TYPE-CHANNEL_NAME.log` or
`/opt/log/glite/glite-transfer-vo-agent-VO_NAME.log`

- If the problem is due to a wrong passphrase, you'll see

```
2005-08-26 07:25:52,281 ERROR  transfer-agent-myproxy - Failed to get the proxy from the
Reason is Error in bind()
ERROR from server: invalid pass phrase
```

Ask then the user to resubmit his/her file, possibly using the `-p` option of `glite-transfer-submit`. In case the problem persists, maybe the user forgot teh passphrase, so ask him/her to restore the credential in myproxy using

```
myproxy-init -s MYPROXY_SERVER -d
```

- In case the agent is not an authorized retriever, you'll see the a similar entry

```
2005-08-26 07:25:52,281 ERROR  transfer-agent-myproxy - Failed to get the proxy from the
ERROR from server: "<anonymous>" not authorized by server's authorized_retriever policy
```

If that is the case, you have to contact the MyProxy server administrator and ask him to add the DN of the certificate of the account used to run the agent. If it still doesn't work, please also check the the agent is running with a valid certificate, following what described here

- in case the entry is similar to

```
2005-08-26 07:25:52,281 ERROR  transfer-agent-myproxy - Failed to get the proxy from the
Error authenticating: GSS Major Status: Authentication Failed
GSS Minor Status Error Chain: (null)
```

This problem is usually due to an expired certificate or to an expired certificate revocation list (crl). Please check the validity of the certicates and update the crl in both the agent and MyProxy nodes

- In case you see errors like:

```
2005-08-26 07:25:52,281 ERROR  transfer-agent-myproxy - Failed to get the proxy from the
Reason is Error in bind()
```

  whithout any other details, please check that the environment variables MYPROXY_TCP_PORT_RANGE and GLOBUS_TCP_PORT_RANGE are unset for the account used to run the agents.

- In the other cases, ask the user to store again his/her certificate in MyProxy, running the command
  `myproxy-init -s MYPROXY_SERVER -d`

Please note that the the `-d` option is required in order to associte the credentials to the DN of the user instead of the account name

If you need to know which MyProxy server is used, have a look here

Which MyProxy Server is used?

When an agent has to perform an operation in behalf of the user, it retrieves the user's delegated credentials from the configured MyProxy server, cache it in the local file system and then impersonate the user by setting the environment variable X509_USER_PROXY. The operations where this is required are:

- Retrieve services endpoints and information from ServiceDiscovery

Symptom: All my transfers fail with a SECURITY_ERROR   9

- Perform the transfer
- Contact the catalog for retrieving the list of replicas and registering the new ones when the transfer is finished (only in case of FPS VO Agent)

The endpoint of the MyProxy server is usually retrieved using ServiceDiscovery, so in case of the file plugin, you need to have an entry in `/opt/glite/etc/services.xml` like

```
<service name='MyProxy'>
   <parameters>
     <endpoint>myproxy://myproxy.cern.ch</endpoint>
     <type>MyProxy</type>
     <version>1.14</version>
   </parameters>
 </service>
```

You can query the InfoSys using the command

```
glite-sd-query -t MyProxy
```

In order to resolve which MyProxy server should be used, the FileTransferAgent looks into the associated services of the FileTransferService who received the user's request (available from gLite 1.3 QF23) or, if not found, takes the first MyProxy server returned by the InformationSystemOverview; you can also force the server to use a specific instance by setting the agent configuration property `MYPROXY_SERVER` (`transfer-agent-myproxy.Server`). In case this property is not set and there is no MyProxy entry registered in the InfoSys, the environment variable $MYPROXY_SERVER is used.

Starting from version gLite 1.3 QF23, the user is also allowed to specify the myproxy he want to use by providing the option `-m myproxy_hostname` in the `glite-transfer-submit` command line.

Error: 'Failed to get proxy certificate from myproxy-fts.cern.ch . Reason is Error in bind()'

When using MyProxy servers, you should ensure that the outgoing port range is set correctly in the agent servers' environments.

This is not reliably done via the `/etc/profile.d/` grid scripts.

See mail from Maarten:

```
Hi Jason,
please check if all the agents have this in their environment:

    MYPROXY_TCP_PORT_RANGE=20000,25000

Note the comma.  The bind() error usually comes from the Myproxy client code defaulting to using

    GLOBUS_TCP_PORT_RANGE=20000 25000

Note the space: the Myproxy client does not handle that properly, leading to occasional bind() er
```

It is recommended to set these explicitly in the file:

```
/etc/sysconfig/glite-data-transfer-agents
```

See bug:

https://savannah.cern.ch/bugs/index.php?31169

Which MyProxy Server is used?   10

Symptom: I've noticed a warning "Cannot Get Agent DN" in the agent log files

You can see this entry in case the agent doesn't run with a valid certificate. When an FTA starts, it put an logs the DN of the certificate the agent will use. This certificate is used to perform the following actions:

- Retrieve the user delegated credentials from MyProxy using the passphrase provided by the user. This happend both on the Channel and the VO Agents
- Perfom the transfer

If the agent doesn't have a valid certificate, it's likely that these operations would fail.

In order to fix this problem, check first that the user running the agents has a valid certificate: usually this certificate are installed in `$HOME/.globus/usercert.pem` and `$HOME/.globus/userkey.pem` and should be owned by the user. In case the certificate is installed in a different place, the environment variables X509_USER_CERT and X509_USER_KEY shoudl be set accordingly. You should also check that the certificate is not expired, by running:

```
openssl x509 -text -in ~/.globus/usercert.pem
```

or

```
openssl x509 -text -in $X509_USER_CERT
```

In case the certificate is valid but the agent always reports the warning, check if there is an expired proxy certificate in `/tmp/x509up_uUSER_ID` (where `USER_ID` is the uder id of the account used to run the agent) and delete it.

Symptom: My srmcopy transfers fail with a dCache MalformedUrl exception

You may notice this error when a user is transfering files to a dChache SE using a channel configured to perform `srmcopy` transfers. This is due to a bug in dCache version <= 1.6.5 in parsing the URL. You have to ask the user to resubmit his/her requests using the following conventions:

- In case the destination SE is dCache, and the source is Castor or DPM
  - Source SURL can be

        srm://<castorsrm>:<port>//srm/managerv1?SFN=<path>
        srm://<castorsrm>:<port>/?SFN=<path>
        srm://<castorsrm>/<path>

  - Destination SURL should be

        srm://<dcachesrm>:<port>/srm/managerv1?SFN=<path>
        srm://<dcachesrm>/<path>

- In case the source SE is dCache and the destination one is Castor or DPM
  - Source SURL should be

        srm://<dcachesrm>:<port>/srm/managerv1?SFN=<path>
        srm://<dcachesrm>/<path>

  - Destination SURL can be

        srm://<castorsrm>:<port>/srm/managerv1?SFN=<path>
        srm://<castorsrm>:<port>//srm/managerv1?SFN=<path>
        srm://<castorsrm>:<port>/?SFN=<path>
        srm://<castorsrm>:<port>/<path>

```
srm://<castorsrm>/<path>
```

- In case both the source and destination SE are dCache
    - ♦ Source SURL should be

```
srm://<dcachesrm>:<port>//srm/managerv1?SFN=<path>
srm://<dcachesrm>/<path>
```

    - ♦ Destination SURL should be

```
srm://<dcachesrm>:<port>/srm/managerv1?SFN=<path>
srm://<dcachesrm>/<path>
```

This problem is fixed in dCache v 1.6.6, however this new version doesn't seem to accept the compact SURL format

```
srm://<srmhost>/<path>
```

If the destination SE is then dCache and it's version is 1.6.6, we suggest to use for both source and destination SURLs either:

```
srm://<srmhost>:<port>/<path>
```

or the fully qualified one:

```
srm://<srmhost>:<port>/srm/managerv1?SFN=<path>
```

Symptom: I've upgraded to 1.4.1 but srmcopy doesn't seem to work

Starting from version 1.3QF23, the FileTransferAgent normalize the SURLs before executing all the SRM get, put and copy requests and the default normalization is to convert them into the compact format

```
srm://<srmhost>/<path>
```

As illustrated here, we observed a problem with dCache srmcopy in version 1.6.6 not working with this format: after ~30 minutes the error returned is

```
number of retries exceeded:org.dcache.srm.scheduler.NonFatalJobFailure: java.io.IOException: both
```

In order to workaround this problem, you have to change the configuration of FilteTransferAgent normalization to use a different format, by setting the ChannelAgent configuration property `ACTIONS_SURLNORMALIZATION` (=transfer-agent-channel-actions.SurlNormalization) to either `compact-with-port` for converting to the format

```
srm://<srmhost>:<port>/<path>
```

or `fully-qualified` for the format

```
srm://<srmhost>:<port>/srm/managerv1?SFN=<path>
```

Symptom: My srmcopy transfers fail with a dCache MalformedUrl exception   12

Please note that this is not a bug in FTS, but a problem in dCache; you might have observed after upgrading to 1.4.1 because this version of FTS has been release more or less at the same time as dCache 1.6.6

I've upgraded to 1.4.1 but the transfer failed with Error in srm__ping: NULL

Starting from version 1.4.1, FTS retrieves the srm endpoint from the information system, instead of parsing the SURL and, in case one of the compact formats are used, using the default port (8443) and service path (srm/managerv1). In case your transfers start failing after the upgrade with an error:

```
Cannot Contact SRM Service. Error in srm__ping: NULL
```

probably the entry in the information system is not correct: in fact, a common error that has been observed is that the SRM endpoint is stored as

```
srm://<srmhost>:<port>/srm/managerv1
```

instead of

```
httpg://<srmhost>:<port>/srm/managerv1
```

You can also check by looking into the transfer log files (located in `/var/tmp/glite-transfer-url-copy-UID/CHANNEL_NAMEfailed` in the related ChannelAgent box) and check the endpoint that is used for the SRM calls

Symptom: The transfer failed with the error: No site found for host ...

During the allocation phase the VOAgent needs to resolve what are the sites that will be involved during the transfer. In order to do that, the agent will look up in the information system the site names of the source and destination SRMs, querying by the hostname retrieved from the provided SURLs.

In case the user gets an error like:

```
Failed to Get Channel Name: No site found for host ...
```

You have to look at the following things:

- The entry concerning the SRM services should be listed in the information system
- The SD library plugins are defined and configured properly (environament variables, files, etc)
- If the file-based plugin is chosen, the `/opt/glite/etc/services.xml` file is properly formatted

In order to do detect errors, it's useful to run the command:

```
su - ACCOUNT_USED_TO_RUN_THE_VOAGENT -c '/opt/glite/bin/glite-sd-query -t SRM --host SRM_HOSTNAME
```

and check the result (this command execute the same query as the agent).

In the problem still persists, it may be worth to have a look at the /proc tanle and see if the

```
/proc/VOAGENT_PROCESS_ID/environ
```

contains the correct values for the `GLITE_LOCATION` and `GLITE_SD_*` environment variables.

Symptom: I've upgraded to 1.4.1 but srmcopy doesn't seem to work   13

In case the StorageElement should not be listed in the information system, you may want to have a look here

The transfer failed with the error: an end-of-file was reached

This error is returned by the globus gridftp library to the ChannelAgent. We don't have many details, but the experience seems to demonstrate that this error happens when the destination SE is full and there's no more space available on disk. In this sense, the `end-of-file was reached` could be interpreted as a `write` command that returned 0 bytes written. If the number of this kind of error increases, set the channel status to `Inactive` and then contact the administrator at the destination site in order to verify the status of the SE.

Which Service Types are used?

The File Transfer Agent needs to interact with external services in order to accomplish its tasks and used the gLite ServiceDiscovery API in order to discover their properties. The involved services are:

- MyProxy: used to retrieve the clients' delegated credentials
- SRM & GridFtp: the site information is used to allocate a transfer job to a channel
- FileCatalog: used by the vo-agent in FPS mode in order to retrieve the sourec replicas to be used for a transfer and registered the new replicas when the transfer is finished

In order to discover that information the File Transfer Agent used the service types listed in Glue Service Types

As reported in bug #12961, however, the service type for a GridFtp server is set to `GridFTP` instead of `gsiftp` and a backward compatible fix is foreseen for a future release. As a temporary workaround you could follow the comments reported on the bug.

I've tried everything, and it still doesn't seem to work

In case your problem is listed in this page, but none of proposed solutions doesn't seem to work, you can generate verbose log files and send them to fts-support. In order to generate these files, please follow the procedure:

For each agent involved (the VO one responsible to allocate a transfer to a channel and retry failed transfer; and the Channel one, responsible to transfer the files and monitor the status), please edit the files `glite-transfer-vo-agent-VO_NAME.log-properties` (in case of VO FTA) and/or `glite-transfer-channel-agent-TYPE-CHANNEL_NAME.log-properties` (in case of Channel FTA) in `/opt/glite/etc/glite-data-transfer-agents.d/` and replace the lines

```
log4j.rootCategory=INFO, file
```

with

```
log4j.rootCategory=DEBUG, file
```

and e
```
log4j.appender.file.fileName=/var/log/glite/glite-transfer-channel-agent-TYPE-CHANNEL_NAME.log
```

or

```
log4j.appender.file.fileName=/var/log/glite/glite-transfer-vo-agent-VO_NAME.log
```

with

```
log4j.appender.file.fileName=/var/log/glite/glite-transfer-channel-agent-TYPE-CHANNEL_NAME.debug.
```

or

```
log4j.appender.file.fileName=/var/log/glite/glite-transfer-vo-agent-VO_NAME.debug.log
```

Restart the agents and let them running for ~ 1 minute; then stop the agents, restore the original values of the modified files, start the agents again and mail these `/var/log/glite/*.debug.log` files to fts-support

Channel Administration

Symptom: How do I set the number of files transferred per VO instead of per channel?

In the FTS Channel Agent you have three parameters you can act on in order to tune the inter-vo scheduling: the channel VO share, the numbers of files that the channel can process concurrently and the `AGENT_VOSHARETYPE` (`transfer-channel-agent.VOShareType`) configuration property. The purpose of this configuration parameter is to define a policy how the VO share should be interpreted for a channel and you can add it to the instance that corresponds to the related channel agent in the configuration file. The allowed values are:

- **normalized**: the share is the value of the channel `voshare` property for the given VO, normalized to the sum of all the shares for all the VOs in the same channel. This option could be used when channel administrators want to guarantee slots for certain VOs, in order to implement some sort of QoS, accepting to eventually penalize the total throughput (transfer slots would be reserved to a VO even if that VO has no job to process)

- **absolute**: the share is the value on the channel `voshare` property expressed as a percentage. No normalization is performed, that means that the sum of all the shares on the same channel can exceed 100%. This option could be used when channel administrators want to balance the share between the VOs, without allowing that a single VO fully allocate a channel but minimizing the risk to allocate slots to VOs that don't have any job to process. This option implies some tuning on the VO share values based on experience, but it would allow to have a compromise between throughput and QoS.

- **normalized-on-active**: the share is the value of the channel `voshare` property for the given VO, normalized to the sum of all the share for all the VOs in the same channel that has at least one job that can be processed by the Channel Agent (job state should be Active, Pending or Canceling). This option is the default and should be used when the channel administrators want to optimize the throughput of the channel (the channel can be fully allocated even by one VO), but with a lower QoS

As an example, supposing you have a channel that has 30 files and 3 VOs, you could have:

VO	Share	Normalized Max Files	Absolute Max Files	Normalized-on-active* Max Files
VO_1	50	15	15	0
VO_2	30	9	9	18
VO_3	20	6	6	12

(* supposing VO_1 has no job to submit)

As you can notice, in case the sum of the VO share is 100, there's no difference between the "normalized" and "absolute" setup. But if this constraint is not respected, you can have:

I've tried everything, and it still doesn't seem to work   15

		Normalized	Absolute	Normalized-on-active*
VO	Share	Max Files	Max Files	Max Files
VO_1	70	14	21	0
VO_2	50	10	15	19
VO_3	30	6	9	11

(* supposing VO_1 has no job to submit)

Please note that the value of the column "Max Files" correspond to the maximum number of files a VO is authorized to submit at the same time. In any case the constraint imposed by the "files" channel property is always respected.

If you want to start with two VOs, setting them each to be able to perform up to 15 transfers concurrently: Set the `AGENT_VOSHARETYPE` (`transfer-channel-agent.VOShareType`) to *normalized* (or *absolute*), having the VO share set to 50 and the channel files set to 30: you'll allow then up to 30 parallel transfers on the channel, but each VO would not be able to submit more than 15 at the same time. In case you'll have to support other VOs, you'll need to adjust these percentages.

Discovery Service

This is how an entry in the `/opt/glite/etc/services.xml` should look:

```
<service name="httpg://lxdpm101.cern.ch:8446/srm/managerv2">
    <parameters>
        <endpoint>httpg://lxdpm101.cern.ch:8446/srm/managerv2</endpoint>
        <type>SRM</type>
        <version>2.2.0</version>
        <site>CERN-PROD</site>
        <wsdl>unset</wsdl>
        <volist>
            <vo>atlas</vo>
            <vo>cms</vo>
            <vo>dteam</vo>
        </volist>
        <param name="atlas:SEMountPoint">/dpm/cern.ch/home/atlas</param>
        <param name="cms:SEMountPoint">/dpm/cern.ch/home/cms</param>
        <param name="dteam:SEMountPoint">/dpm/cern.ch/home/dteam</param>
    </parameters>
</service>
```

"No site for host" error

- Check that the information in the **endpoint** node is correct
- Check that the **volist** node contains an entry for your VO

"No channel found, channel closed for your VO..." error

- Check that the **site** node is correct for the endpoints for which the job failed
- Verify that a channel is defined between those two sites
    - ♦ `glite-transfer-channel-list` command
- Verify that your VO has a (non-null) share defined on the channel

"No SRM method factory found" error

- Check the **version** node for the endpoint. Allowed values are:
    - 1.1 or 1.1.*
    - 2.2 or 2.2.*

Last edit: LaurenceField on 2008-09-26 - 15:44
Number of topics: 1

Maintainers: GavinMcCance, PaoloTedesco

This topic: EGEE > DMFtsSupport
Topic revision: r35 - 2008-09-26 - LaurenceField