Table of Contents

FTS Channel Administration 2.0

This page describes the basic administration and monitoring of channels upon the service.

Underlying details of the **server** (starting, stopping, logfiles, etc) are described in FtsServerAdmin20.

Server installation and configuration is described in FtsServerInstall20. Standalone client installation and configuration is described in FtsClientInstall20.

New channel configuration parameters

Starting from org.glite.data.transfer-agents version 3.0.3.1, transfer parameters are read from the t_channel table in the database, and not from channel agents configuration files.

A migration script will be installed by the transfer-agents rpm in the ${GLITE_LOCATION}/etc/glite-data-transfer-agents.d directory. The script requires no parameters and can be run with

```
python update_channels.py
```

The script will:

- locate and read the channel agents configuration files.
- generate an sql update script for the t_channel table.
- run the sql script.

The following output files are produced:

- update_channels.log: the script log file (same contents as standard output).
- update_channels.oracle.log: the output of the sqlplus command
- update_channels.sql: the sql update script

To update the channel definitions in the database, run this script **on all the channel agent nodes**.

In case the script returns an error, please send the log files to fts-support.

Creating new channels

The FtsServerInstall20 guide details how to create channels in the database. In short, you need to:

1. Add a new channel in the DB using `glite-transfer-channel-add`.
2. Add a new agent daemon configuration to one of your agent nodes to service this new channel.

Managing channels

A channel or "queue" can be:

- point-to-point network connection, e.g. for the Tier-0 there is defined a single channel to every Tier-1 site.
- a "catch-all" channel

A channel or "queue" is the basic unit of management. The channel manager can control whether a channel is active or not, and how much work is being put on the network for a given channel. Channels can be managed independently.

To perform any channel management operations, you must have a valid grid or VOMS proxy:

```
voms-proxy-init
```

and you must also be authorized, either as a service administrator or as a channel administrator for the given channel. Use the `glite-transfer-getroles` command to see what privileges you have upon the service. Note that by default, the root account on the FTS server machine is listed in the server administrator mapfile (in fact the host cert is listed in the mapfile, which the root account uses as its user cert).

Initially, try a list to see what channels are already defined on the system (if it is a new install on a blank database, there should be none):

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

To see a basic view of the channel:

```
glite-transfer-channel-list CHANNELNAME
```

To see a fuller view of the channel:

```
glite-transfer-channel-list -x CHANNELNAME
```

note that `man` pages are available for all commands.

Problems you may see

If you see:

```
list: listChannels: SOAP fault: SOAP-ENV:Client - CGSI-gSOAP: Could not open connection !
(TCP connect failed in tcp_connect())
```

then either the service is down, or your `services.xml` file is pointing to the wrong endpoint. You can see what endpoint the client is attempting to connect to by using the version flag:

```
glite-transfer-channel-list -v
```

If you see:

```
list: Service discovery: No services of type org.glite.ChannelManagement were found
```

then either the `services.xml` file is missing or not readable by the client.

If you see:

```
list: listChannels: SOAP fault: "http://xml.apache.org/axis/":Server.NoService - The AXIS
engine could not find a target service to invoke!  targetService is ChannelManagement
```

then either the endpoint specified in the `services.xml` file is incorrect, or the service is misconfigured.

To check whether the endpoint specified in the `services.xml` file is correct, connect to it directly with a web browser. e.g. connect to https://yourhostname:8443/glite-data-transfer-fts/services/FileTransfer⧉ (you will need your grid certificate loaded in your browser). If the service is listening, you should see a web page with a

message like:

```
Hi there, this is an AXIS service!
```

If you see:

```
list: listChannels: You are not authorised for channel management upon this service
```

then you are not in the manager mapfile. Look in the FTS server logs (`org.glite.data`) to see how the authorisation decision was made.

Changing the state of channels

Although any parameters may be changed subsequently on a channel, only three operations are common: setting it active, setting it inactive and setting the number of concurrent transfers. Note (new in FTS 2.0) it is recommend to add a comment using the `-m` option.

To set a channel `mychannel` to state `Inactive`:

```
glite-transfer-channel-set -S Inactive CHANNELNAME -m "Because I can"
```

This will stop putting any further work on the network. Any individual file transfers that have already been started will complete, so it can take a minute or two for all activity to stop.

To set the channel to state `Active`:

```
glite-transfer-channel-set -S Active CHANNELNAME -m "Set back to normal operations"
```

This will start putting work on the network if there are any jobs assigned to that channel in the `Pending` state.

To change the number of concurrent transfers being put on the network for a given channel:

```
glite-transfer-channel-set -f 10 CHANNELNAME -m "Increase to 10 to see what happens to the rate"
```

This will make the agent try to maintain 10 concurrent transfers on the channel when it is `Active`.

Auditing operations

The `channel-set` commands are audited by the service. You can view previous operations using the comand:

```
glite-transfer-channel-audit CHANNELNAME
```

Dropping channels

To drop a channel:

```
glite-transfer-channel-drop CHANNELNAME
```

It is a current restriction that a channel **cannot** be dropped if any jobs have been assigned to it (because we do not cascade constraints on the database delete). We will provide a script to do this soon. Make sure that you are happy with a channel definition before submitting work that will be assigned to it.

Problems you may see   3

Setting a per-VO share on a channel

This can be done using the command:

```
glite-transfer-channel-setvoshare VONAME CHANNELNAME SHARE
```

where the shares are all calculated relative to one another (the absolute value is not important). Note also, that in the default mode, the shares are 'elastic' in that if one VO has no jobs, the others will absorb its share until it returns.

To stop a VO transferring for a while, set the share to 0. New jobs will be queued onto the channel.

To remove a VO from a channel, set the share to "-1" (using `getopts --` feature). New jobs will be rejected from the channel.

Setting a per-VO cap on a channel

The elastic nature of the share can cause problems when a SRM pool is not sized to take the full number of transfers configured on a channel. This was noted particularly for the `dteam` background transfers when, in some quiet times, there were no jobs from any of the experiments - this caused `dteam` to absorb the entire share of the channel which overloaded the source and destination `dteam` disk pools with too man requests.

It is therefore possible to limit the maximum number of jobs for a given VO on a channel:

```
glite-transfer-channel-setvolimit VONAME CHANNELNAME LIMIT
```

where `LIMIT` is the number of concurrent files.

To remove a limit, set it to "-1" (using `getopts --` feature).

Channel managers

To view current channel managers:

```
glite-transfer-channel-listmanagers
```

To set and remove someone as a channel manager, using their quoted certificate subject name:

```
glite-transfer-channel-addmanager CHANNELNAME "/DC=ch/DC=cern/OU=Organic
Units/OU=Users/CN=mccance/CN=453200/CN=Gavin Mccance"
```

```
glite-transfer-channel-removemanager CHANNELNAME "/DC=ch/DC=cern/OU=Organic
Units/OU=Users/CN=mccance/CN=453200/CN=Gavin Mccance"
```

Manual interventions on channels

When a file fails on a single transfer attempt, the system will attempt to retry it after some delay. The number of retry attempts and the delay are configurable on the service (defaults are 3 retries, waiting minimum 10 minutes between each retry attempt).

If a file fails, it will be placed in state `Waiting`. After the retry period has elapsed it will be placed back in the `Ready` state, and will subsequently be picked up again by the transfer agent.

◗ Details about Hold state (not configured by default) ◗ Details about Hold state (not configured by default)

If a file has been retried more than the max number of times, it is placed in the `Hold` state. This state indicates that some manual intervention should be made to rescue (or cancel) the file. The channel admin should periodically check for jobs with files in `Hold` state:

```
glite-transfer-list -c CHANNELNAME Hold
```

If there are any, there are two options for you. You may fix the whatever you believe the problem may be (e.g. reboot the SRM cluster) and try for one more retry.

```
glite-transfer-channel-signal -c CHANNELNAME Pending
```

All jobs on the channel in `Hold` state will be dropped back onto the `Pending` queue. The other option is to set all the jobs on the channel in `Hold` state to `Canceling` if you believe that there is nothing you can do to rescue the jobs (for example, if the filenames are invalid).

```
glite-transfer-channel-signal -c CHANNELNAME Cancel
```

You also have the option of applying the signal command to a single job, rather than all jobs on the entire channel.

```
glite-transfer-channel-signal -j 17cc7055-d3ac-11d9-a905-f173b72e4547 Pending
```

Running a test job

Identify a source SRM file name and a suitable destination SRM file name. The file names must be in the full SURL format:

```
srm://castorgridsc.cern.ch:8443/srm/managerv1?SFN=/castor/cern.ch/grid/dteam/storage/transfer-tes
```

Assuming you are using delegation (as below), you do not need to load a proxy into MyProxy.

```
voms-proxy-init

glite-transfer-submit \
   srm://castorgridsc.cern.ch:8443/srm/managerv1?SFN=/castor/cern.ch/grid/dteam/storage/transfer-
   srm://srm1.tier1.sara.nl:8443/srm/managerv1?SFN=/data/dteam/testdestfile-glite-1.1.1-000.dat
```

choose your SURLs appropriately. This will return a UUID identifier, e.g.

```
17cc7055-d3ac-11d9-a905-f173b72e4547
```

To check the status of the job, run:

```
glite-transfer-status --verbose 17cc7055-d3ac-11d9-a905-f173b72e4547
```

which should print the state and some other information. You should see the job progressing through:

- `Submitted` (if you're fast). The job is still to be assigned to channel.
- `Pending` and `Ready`. The job has been assigned to a channel, and is awaiting a transfer slot.
- `Active`. The job is being transferred or the file is awaiting retry.
- `Hold`. The single file in the job failed 3 times and needs to be rescued by manual intervention.
- `Done`. The file has completed successfully.
- `Finished`. The job and its file has completed successfully.
- `Failed`. The job and its (all of its) file(s) failed.
- `Finished`. The job and (some of its) file(s) failed.

See the FtsServerAdmin20 guide which describes what is happening on the FTS machine itself.

If the job has entered state `Finished`, then the test went fine.

Maintainer: GavinMcCance

This topic: LCG > FtsChannelAdmin20
Topic revision: r5 - 2007-11-21 - PaoloTedesco