

Table of Contents

Support WLCG Transfer Dashboard Guide.....	1
WLCG Transfer Dashboard Overview.....	1
Infrastructure.....	1
MSG Brokers.....	1
Brokers Information.....	1
Topics and queues used in the brokers.....	2
MSG Web Interface.....	2
Production Server.....	3
Current Status (29/10/2012).....	3
Integration Server.....	4
Current Status (29/10/2012).....	4
Dashboard agents.....	4
Log files.....	4
Collectors:.....	5
Database Agents.....	5
Monitors.....	6

Support WLCG Transfer Dashboard Guide

This page documents the support of the WLCG Transfers Dashboard.

WLCG Transfer Dashboard Overview

WLCG Transfer Dashboard is composed by a web application, which runs under an Apache server and aims to show different statistics and multiple agents which perform different tasks, such as:

- Generate Statistics
- Collect information

The web application is managed through the standard Apache server daemon, which permits to stop / start / restart the web service. On the other hand, the dashboard agents are managed using the dashboard service configurator tool. Below is shown an overview of the available WLCG Transfer Dashboard agents:

Name	Description
Collectors	Agents collecting messages sent by FTS and XRootD technologies to a broker. The messages are stored in a database
Monitor	Agents which aim generating statistics about information stored into a database

Note: You can find the log files for each of these components in /opt/dashboard/var/log/#NAME_LOG_FILE#

Infrastructure

- PRODUCTION
 - ◆ hosts: dashb-wlcg-transfers (dashboard63 - dashboard71)
 - ◆ brokers: dashb-mb (gridmsg107 - gridmsg108 - gridmsg109) fts and gridmsg007 xrootd
 - ◆ fts start queue: /queue/Consumer.dashb.transfer.fts_monitoring_start
 - ◆ fts complete queue: /queue/Consumer.dashb.transfer.fts_monitoring_complete
 - ◆ xrootd atlas queue: /queue/Consumer.dashb_wlcg.xrdpop.fax_popularity
 - ◆ xrootd cms queue: /queue/Consumer.dashb_wlcg.xrdpop.uscms_popularity
- INTEGRATION
 - ◆ hosts: dashb-wlcg-transfers-dev (dashboard59)
 - ◆ brokers: dashb-mb (gridmsg107 - gridmsg108 - gridmsg109) fts and gridmsg007 xrootd
 - ◆ fts start queue: /queue/Consumer.dashb-int.transfer.fts_monitoring_start
 - ◆ fts complete queue: /queue/Consumer.dashb-int.transfer.fts_monitoring_complete
 - ◆ xrootd atlas queue: /queue/Consumer.dashb_dev.xrdpop.fax_popularity
 - ◆ xrootd cms queue: /queue/Consumer.dashb_dev.xrdpop.uscms_popularity

MSG Brokers

WLCG Transfer dashboard uses the MSG brokers provided for IT-GT for getting messages. The brokers store the information sent by the FTS waiting for being delivered after by WLCG transfer system.

Brokers Information

hostname	state	ActiveMQ version	DNS alias	Monitoring links
gridmsg007	integration	5.5.1-fuse-01-06	dashb-mb.cern.ch	https://gridmsg007.cern.ch/admin/topics.jsp
gridmsg107	production	5.5.1-fuse-01-06	dashb-mb.cern.ch	https://gridmsg107.cern.ch/admin/topics.jsp
gridmsg108	production	5.5.1-fuse-01-06	dashb-mb.cern.ch	https://gridmsg108.cern.ch/admin/topics.jsp
gridmsg109	production	5.5.1-fuse-01-06	dashb-mb.cern.ch	https://gridmsg109.cern.ch/admin/topics.jsp

Topics and queues used in the brokers

The **topic names** used in any of the above brokers are:

- transfer.fts_monitoring_start
- transfer.fts_monitoring_complete
- xrdpop.fax_popularity
- xrdpop.uscms_popularity

And the **queue names** are:

- Production:
 - ◆ Consumer.dashb.transfer.fts_monitoring_start
 - ◆ Consumer.dashb.transfer.fts_monitoring_complete
 - ◆ Consumer.dashb_wlcg.xrdpop.fax_popularity
 - ◆ Consumer.dashb_wlcg.xrdpop.uscms_popularity
- Integration:
 - ◆ Consumer.dashb-int.transfer.fts_monitoring_start
 - ◆ Consumer.dashb-int.transfer.fts_monitoring_complete
 - ◆ Consumer.dashb_dev.xrdpop.fax_popularity
 - ◆ Consumer.dashb_dev.xrdpop.uscms_popularity

The queues are a link between the collector and topics *transfer.fts_monitoring_start* and *transfer.fts_monitoring_complete* which store the information sent by the FTS until the collector recovers it. The name of the queue is a concatenation of the word *Consumer.dashb* + topic name.

On the other hand, the queues called *transfer.fts_monitoring_rejected_start* and *transfer.fts_monitoring_rejected_complete* are used for the collectors to send those messages that cannot be decoded by this one (this queue should always be empty).

MSG Web Interface

MSG brokers provide a web interface to supervise them in case of incident. The links for each MSG broker are as follow:

Name broker	Link access
gridmsg007	https://gridmsg007.cern.ch/admin/topics.jsp
gridmsg107	https://gridmsg107.cern.ch/admin/topics.jsp
gridmsg108	https://gridmsg108.cern.ch/admin/topics.jsp
gridmsg109	https://gridmsg109.cern.ch/admin/topics.jsp

Note: You have to be register in these brokers

Once inside you will see the topics used for the publishers to send information. When the collector is connected to broker (seen MSG brokers), two queues are created. To see them, go on the Queues link at the top of the page to get the list of queues. Now you should find the queues *Consumer.dashb.transfer.fts_monitoring_start* and *Consumer.dashb.transfer.fts_monitoring_complete*.

To check that everything is working fine, you must check the information about the queues where each field means:

- **Name:** queue name.
- **Number Of Pending Messages:** represents the number of messages that are stored in the server waiting for being delivered. If a consumer is running smoothly it should be 0.

- **Number Of Consumers:** number of the active consumers in the queue. This number should be 1.
- **Messages Enqueued:** number of messages sent to topics where you have an active subscription. Those messages have been allocated in a queue for delivery.
- **Messages Dequeued:** number of messages that you have received (and acknowledged). They have been subtracted from the queue.

Therefore, the ideal situation for the start and complete messages will be: 1 consumer and enqueued messages ~ = dequeued messages.

To understand better follow the next example.

Name	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued
Consumer.dashb.transfer.fts_monitoring_start	0	1	29770	29770
Consumer.dashb.transfer.fts_monitoring_complete	0	1	29730	29730

The best situation is when *enqueued messages* = *dequeued messages* what means that all messages sent to the topic were received by the consumer and successfully inserted into the database.

Production Server

WLCG Transfer Dashboard is running under dashboard63 and dashboard71 (both virtual machines) provided both machines by the IT-PES department. This means that exists redundant mode for collectors and UI in case of downtime in some of the machines.

Therefore, to configure or resolve incidents will be necessary to connect with some of these machines through ssh. To do that, you will need to open a terminal in your computer and type:

```
[ddarias] /home/ddieguez > ssh root@dashboard[63-71]
Last login: Wed Apr 18 10:30:17 2012 from pb-d-128-141-72-77.cern.ch
[root@dashboard71 ~]#
```

Once inside you will need to change of user, type:

```
[root@dashboard71 ~]# su - dboard
[dboard@dashboard71 ~]$
```

Now you could check the different log files, see the agent status and make some actions which will help you to solve different incidents.

Current Status (29/10/2012)

Alias of the web application for these machines: <https://dashb-wlcg-transfers.cern.ch/ui/>

Collectors running for FTS technology (the name of the service group used to lunch the collector is transfer.collector)

Both alias and collectors are running in mode load-balancing.

Queues used for these collectors:

- Consumer.dashb.transfer.fts_monitoring_start
- Consumer.dashb.transfer.fts_monitoring_complete
- transfer.fts_monitoring_rejected_start
- transfer.fts_monitoring_rejected_complete

All the above queues belong to MSG Brokers on Production

Integration Server

Under integration server is running a copy of WLCG Transfer Dashboard which is used for integrating and testing new features before deploying into production server. The virtual machine used for it is dashboard59 and it follows the same behaviour in terms of agents than for the production server.

Current Status (29/10/2012)

Currently, this version includes XRootD and latency prototypes regarding to production server.

Alias of the web application in this machine: <https://dashb-wlcg-transfers-dev.cern.ch/ui/>

There is a collector running to consume messages from FTS Technology and another one to consume messages from XRootD.

The corresponding service group names to launch the collectors are:

- transfer.xrootdcollector for XRootD
- transfer.ftscollector for FTS

Queues used for these collectors:

- Consumer.dashb-dev.transfer.fts_monitoring_start (located in production msg brokers)
- Consumer.dashb-dev.transfer.fts_monitoring_comp (located in production msg brokers)
- transfer.fts_monitoring-dev_rejected_start (located in production msg brokers)
- transfer.fts_monitoring-dev_rejected_complete (located in production msg brokers)
- Consumer.dashb_wlcg.xrdpop.fax_popularity (located in integration msg broker)
- Consumer.dashb_wlcg.xrdpop.uscms_popularity (located in integration msg broker)
- transfer.xrootd_monitoring_rejected_queue (located in integration msg broker)

Dashboard agents

To see the agent status, you can type in the terminal the follow command (see Example [Service Operation](#)):

```
[dboard@dashboard71 ~]$ dashb-agent-list
SERVICE GROUP      STATUS      SERVICES
transfer.stress.test STOPPED    'stress.test',
transfer.mock.producer STOPPED    'transfer.mock.producer',
transfer.republisher STOPPED    'fts_monitoring_start', 'fts_monitoring_complete',
transfer.collector  STARTED    'transfer.collector1', 'transfer.collector2',
transfer.monitor     STARTED    'computeStats', 'aggregateStats', 'computeErrorSummaries',
transfer.curl.data   STARTED    'curlVOFeedCMS', 'curlVOFeedAtlas', 'curlVOFeedLhcb', 'curlV
```

Log files

Each dashobard agent shown privously has a log file placed in `/opt/dashboard/var/log/#SERVICE_GROUP_NAME#`

Therefore, if you want to see the log file from transfer.collector you should type:

```
[dboard@dashboard71 ~]$ tail -f /opt/dashboard/var/log/transfer.collector
2012-04-18 10:48:22,563 - CollectMessages:255 - INFO - dashb-mb.cern.ch/queue/Consumer.dashb.tran
2012-04-18 10:48:22,682 - CollectMessages:255 - INFO - dashb-mb.cern.ch/queue/Consumer.dashb.tran
```

```
2012-04-18 10:48:23,068 - CollectMessages:245 - INFO - dashb-mb.cern.ch/queue/Consumer.dashb.tran
2012-04-18 10:48:23,168 - CollectMessages:255 - INFO - dashb-mb.cern.ch/queue/Consumer.dashb.tran
2012-04-18 10:48:23,194 - CollectMessages:245 - INFO - dashb-mb.cern.ch/queue/Consumer.dashb.tran
2012-04-18 10:48:23,294 - CollectMessages:255 - INFO - dashb-mb.cern.ch/queue/Consumer.dashb.tran
2012-04-18 10:48:23,668 - CollectMessages:255 - INFO - dashb-mb.cern.ch/queue/Consumer.dashb.tran
```

Collectors:

To restart the collectors type the follow command:

```
[dboard@dashboard71 ~]$ dashb-agent-restart transfer.collectors
.STARTED
[dboard@dashboard71 ~]$
```

To check that the log file is updated type:

```
tail -f /opt/dashboard/var/log/transfer.collector
```

The activity of the collectors can be seen trough the below link

http://http://dashb-wlcg-transfers.cern.ch/ai/insertion_rate.html which shows different statistics about the insertion rate.

Database Agents

Database agents are jobs which are running inside of the database. To see details about them you will need to use sqlplus through a previous connection with lxplus machine.

Lets see an example:

```
[ddarias] /home/ddiequez > ssh lxplus

[lxplus420] /afs/cern.ch/user/d/ddiequez > sqlplus

SQL*Plus: Release 10.2.0.5.0 - Production on Thu May 3 11:36:43 2012

Copyright (c) 1982, 2010, Oracle. All Rights Reserved.

Enter user-name: lcg_dashboard_tfr_r@lcgr
Enter password: (see Note)
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Real Application Clusters and Real Application Testing options
```

To know what the password is, please have a look to the database configuration file (*dashboard-dao.cfg*) under directory */opt/dashboard/etc/dashboard-dao/* in one the production machines seen before.

Example:

```
[dboard@dashboard71 ~]$ vi /opt/dashboard/etc/dashboard-dao/dashboard-dao.cfg
```

Once inside of sqlplus, you will have to type this command

```
SQL> set serveroutput on
```

After doing that, you will be able to see all details about jobs executing the following procedure as shown below:

```
BEGIN
  DASHBOARDTRANSFERS.VIEWSCHEDULERJOBS;
```

```
END;  
/
```

And then you will see all detail jobs.

```
Row: 1  
  job_name: SERVER_UPDATE  
  job_action: DASHBOARDTRANSFERS.SERVERUPDATE  
  repeat_interval: FREQ = MINUTELY; INTERVAL = 20  
  enabled: TRUE  
  run_count: 63  
  failure_count: 0  
  last_start_date: 03-MAY-12 11.55.46.013747 AM EUROPE/ZURICH  
  last_run_duration: +000000000 00:00:00.108444  
  next_run_date: 03-MAY-12 12.15.46.000000 PM EUROPE/ZURICH
```

```
Row: 2  
  job_name: VO_UPDATE  
  job_action: DASHBOARDTRANSFERS.VOUPDATE  
  repeat_interval: FREQ = MINUTELY; INTERVAL = 20  
  enabled: TRUE  
  run_count: 63  
  failure_count: 0  
  last_start_date: 03-MAY-12 11.56.12.014350 AM EUROPE/ZURICH  
  last_run_duration: +000000000 00:00:00.051942  
  next_run_date: 03-MAY-12 12.16.12.000000 PM EUROPE/ZURICH
```

.....

```
Row: 8  
  job_name: MSG_ALARM_CHECK  
  job_action: DASHBOARDTRANSFERS.MSGALARM  
  repeat_interval: FREQ = MINUTELY; INTERVAL = 20  
  enabled: TRUE  
  run_count: 59  
  failure_count: 0  
  last_start_date: 03-MAY-12 11.53.48.132221 AM EUROPE/ZURICH  
  last_run_duration: +000000000 00:00:08.754900  
  next_run_date: 03-MAY-12 12.13.48.000000 PM EUROPE/ZURICH
```

Total number of jobs: 8

Monitors

To restart the monitors type the follow commands from lxplus or sqldeveloper:

```
BEGIN  
  DASHBOARDTRANSFERS.STOPMONITOR;  
END;  
/
```

```
BEGIN  
  DASHBOARDTRANSFERS.STARTMONITOR;  
END;  
/
```

-- DanielDieguez - 09-Dec-2011

This topic: [ArdaGrid > SupportWLCGTransferDashboard](#)

Topic revision: r12 - 2013-01-28 - AlexandreBeche



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

SupportWLCGTransferDashboard < ArdaGrid < TWiki

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback