

Table of Contents

WLCG Transfers Dashboard FTS Queue Status Monitoring.....	1
Requirements.....	1
Development.....	1
Milestones.....	2
Delivered.....	2
In progress.....	2
Future (draft).....	2
Links.....	3

WLCG Transfers Dashboard FTS Queue Status Monitoring

QUEUE STATUS MONITORING HAS NOW BEEN INCORPORATED INTO THE TRUNK CODE AND IS DEPLOYED TO THE INTEGRATION SERVER. SEE [SupportWLCGTransferDashboard#Integration_Server](#). THE REMAINDER OF THIS PAGE IS NOW OBSOLETE.

This page documents the development of FTS queue status monitoring, which is a component of WLCGTransfersDashboard.

Requirements

Key features:

- Consume FTS queue status messages from Dashboard MSG brokers (see [WLCGTransferMonitoring#FTS_message_structure](#)).
- Insert FTS queue status messages into Dashboard Oracle database.
- Expose current queue status data via web API (JSON).
- Visualise current queue status within Dashboard UI (see <http://dashb-wlcg-transfers.cern.ch/ui/>).
- Expose historical queue status data via web API (JSON).
- Visualise queue status history within Dashboard UI.

Possible extensions:

- Correlation between queue status and transfer statistics.

Requirements will be further detailed in consultation with the VOs, see [WLCGTransfersDashboard#VO_requests](#).

Development

Estimate:

- The development is predicted to take 2-3 FTE months.

Environment:

- Development branch of code in SVN
 - ◆ NAME: dashboard-transfers-0-2-0-branch
 - ◆ WEB_URL:
<http://svnweb.cern.ch/world/wsvn/dashboard/branches/arda.dashboard.transfers/dashboard-transfers-0-2-0-branch>
 - ◆ SVN_URL:
<https://svn.cern.ch/repos/dashboard/branches/arda.dashboard.transfers/dashboard-transfers-0-2-0-branch>
 - ◆ NOTES:
 - ◇ Read/write access granted to auzhinsk
 - ◇ Use "python setup.py release -c" to release new 0.2.1_rcX RPM.
- Development database account.
 - ◆ ACCOUNT: lcg_dashboard_tfr_dev @ int11r
 - ◆ CREDENTIALS: see /opt/dashboard/etc/dashboard-dao/dashboard-dao.cfg on dashboard59.
- VM with WLCG Transfers Dashboard (from SVN branch).
 - ◆ SERVER: dashboard59

- ◆ WEB UI:
 - ◇ (OPEN) <http://dashb-wlcg-transfers-devq.cern.ch/ui/#tab=queues>
 - ◇ (CERN) <http://dashboard59/ui/#tab=queues>
- ◆ NOTES:
 - ◇ Root access via kerberos granted to auzhinsk
 - ◇ dashboard-transfers-0.2.1_rc2 is currently deployed.
 - ◇ Use "yum install dashboard-transfers-0.2.1_rcX" to update server.
 - ◇ For the purpose of having some data to show, the server is currently configured to read from lcg_transfers_test_r @ int6r. It should be reconfigured to use lcg_dashboard_tfr_dev @ int11r, see TODO below.
- TO-DO:
 1. Create schema and packages on lcg_dashboard_tfr_dev @ int11r using data-oracle-schema-objects.sql and data-oracle-schema-packages.sql.
 2. Update /opt/dashboard/etc/dashboard-dao/dashboard-dao.cfg on dashboard59 to use lcg_dashboard_tfr_dev @ int11r
 3. Set up CollectMessages.py to consume messages from dashb-mb-test.cern.ch topics.
 4. Start statistics generation procedures on lcg_dashboard_tfr_dev @ int11r, see start/stop procedures in data-oracle-schema-packages.sql
- NOTES:
 - ◆ Daniel can help with the above TODOs as required.
 - ◆ There is already a skeleton view object for the Queues tab with comments: view_queues.js
 - ◆ See also: [Dashboard documentation](#), [Xbrowse documentation](#).

Cycle:

- Develop on dev SVN branch.
- Deploy to dev VM and DB account.
- On delivery of a functional milestone:
 - ◆ review functionality with stakeholders
 - ◆ review code
 - ◆ merge to SVN trunk
 - ◆ deploy/test in integration
 - ◆ deploy/test in production

Milestones

Delivered

1. Create development environment (CERN). DELIVERED 06/06/2012
 - ◆ Create development branch of code in SVN.
 - ◆ Request development database account.
 - ◆ Set-up VM with WLCG Transfers Dashboard (from SVN branch).
 - ◆ Deliverable:
 - ◇ Access to SVN, DB and VM provided to Dubna team

In progress

Future (draft)

1. Consume FTS queue status messages to DB table.
 - ◆ The consumer should re-use the existing collector code.
 - ◆ The table should be partitioned by created_time.
 - ◆ Deliverable:

- ◇ Demonstration that messages are moved from the queue to DB.
- 2. Propose visualisation of current queue status
 - ◆ The visualisation should be designed to integrate into the current UI.
 - ◆ It may include an expandable table, plots, etc.
 - ◆ Deliverable:
 - ◇ Document / presentation outlining the proposed solution.
- 3. Expose current queue status via web API
 - ◆ This should be done within the Dashboard framework
 - ◆ Deliverable:
 - ◇ Demonstration of web API returning JSON.
 - ◇ API documented on twiki.
- 4. Visualise current queue status
 - ◆ This is the implementation of the above proposal.
 - ◆ It should use the above web API.
 - ◆ Deliverable:
 - ◇ Demonstration of web UI integrated into existing UI.
- 5. Propose visualisation of queue status history
 - ◆ The visualisation should be designed to integrate into the current UI.
 - ◆ It may include an expandable table, plots, etc.
 - ◆ Deliverable:
 - ◇ Document / presentation outlining the proposed solution.
- 6. Expose queue status history via web API
 - ◆ This should be done within the Dashboard framework
 - ◆ Deliverable:
 - ◇ Demonstration of web API returning JSON.
 - ◇ API documented on twiki.
- 7. Visualise queue status history
 - ◆ This is the implementation of the above proposal.
 - ◆ It should use the above web API.
 - ◆ Deliverable:
 - ◇ Demonstration of web UI integrated into existing UI.

Links

- User Interface: <http://dashb-wlcg-transfers.cern.ch/ui/>
- SVN: [trunk](#), [branch](#)
- Bugs: [Savannah](#)
- Twikis: [WLCGTransferMonitoring](#), [WLCGTransfersDashboard](#).

-- DavidTuckett - 25-Apr-2012

This topic: LCG > WLCGTransfersDashboardQueueStatus

Topic revision: r6 - 2013-04-10 - DavidTuckett



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)