

# Table of Contents

<b>WLCG Messaging System for Grids (MSG)</b> .....	<b>1</b>
Overview.....	1
Background Presentations.....	1
Infrastructure.....	1
Testing.....	1
Message Protocols.....	1
Tools provided.....	2
msg-publish-simple.....	2
Using publish simple.....	2
consume2oracle.....	3
Specific Channel definitions.....	3
Proposed usage.....	3

# WLCG Messaging System for Grids (MSG)

## Overview

This page contains an overview of the WLCG MSG - Messaging System from Grids. This is a solution based around Apache ActiveMQ<sup>☞</sup> which aims to help the integration and consolidation of the various grid monitoring systems used in WLCG. It arose out of work done within the context of the WLCG Monitoring Working Groups

## Background Presentations

We have presented a series of different presentations at the WLCG GDB on different aspects of WLCG monitoring in general, and the messaging layer, MSG, in particular:

- Architectural principles (ppt)<sup>☞</sup> - A more general presentation giving the architectural principles of the entire WLCG monitoring system, out of which comes the requirement for some sort of loose coupling and distribution via a reliable messaging system
- Strategy (ppt)<sup>☞</sup>
- Some worked examples (ppt)<sup>☞</sup> - Examples of integrating some actual components, including OSG RSV tests into SAM
- WhiteAreasPresentation\_Final2.pps: White Areas Presentation 23 May 2008

## Infrastructure

MSG uses Apache ActiveMQ<sup>☞</sup> as the message bus. This is a fully featured JMS implementation that also offers non-Java solutions for publishing and consuming messages. It also integrates Apache Camel<sup>☞</sup>, which implements the enterprise integration patterns<sup>☞</sup> from the book<sup>☞</sup> by Gregor Hohpe and Bobby Woolf. We use much of the terminology of these patterns, such as Message Store, Control Bus, Message Broker, ...

The aim is to use MSG as an integration bus for all the grid monitoring systems we currently have within WLCG, and allow new systems to be easily built using a scalable infrastructure. It is currently a non-goal to provide a general messaging system for general usage by grid middleware, but we hope our experience could be useful if such a system is ever deployed!

## Testing

We have carried out (and are in the process of completing) a series of evaluations of various aspects of ActiveMQ as a message broker:

- GridMessagingSystemMasterSlaveTestLogbook - Evaluation of various Master-Slave techniques for building high-reliability message broker clusters<sup>☞</sup>
- GridMessagingSystemPerformanceTestLogbook - Testing message throughput against a variety of persistency<sup>☞</sup> options

## Message Protocols

As a standard message format, we use the message format defined in the GridMonitoringProbeSpecification. Of course, depending on the actual usage (see below) the required and optional fields may change - but what says constant is the basic syntactic format.

- **TODO** - Split out underlying message format from description of probe interactions.

We believe over time other formats will also become standard for interchange, probably some form of JSON notation and a simple XML format (Atom ?)

## Tools provided

Latest releases available at [http://www.sysadmin.hep.ac.uk/rpms/egee-SA1/sl4/x86\\_64/RPMS.release](http://www.sysadmin.hep.ac.uk/rpms/egee-SA1/sl4/x86_64/RPMS.release)

### msg-publish-simple

Current Status

#### Using publish simple

Publish-simple is a lightweight python script which allows publishing into MSG. It accepts a file as input, whose contents should be records separated by EOT (read [GridPublisherSpecificationGridView] for examples). The configuration is done on the side, defining the message class keys, headers and destinations, list of brokers to contact, and logging configuration.

By steps:

1. install the rpm (rpm -ivh <http://www.sysadmin.hep.ac.uk/rpms/grid-services/RPMS.monitoring/msg-publish-simple-x.y.z-r.noarch.rpm>);
2. vim /opt/lcg/etc/msg/msg-publish.conf
3. From your application, call: python /opt/lcg/bin/msg-publish. You'll get:

Usage: msg-publish [options] MESSAGE\_CLASS [FILE1] [FILE2...]

Where MESSAGE\_CLASS must be one of the followings: org.wlwg.usage.jobStatus  
and FILE is a file containing the tuples of test results

OPTIONS:

```
-h, --help
    Print this help text and exit.
-b, --bulk
    Bulk publishing
-v, --verbose
    Be more verbose
-c, --conf
    Alternative configuration file
-a, --attributes
    List of header attributes
-d, --destination
    Alternative message destination
-g, --generic
    Generic message class (do not validate class name and records)
-s, --sign
    Sign with my proxy certificate (NOT IMPLEMENTED)
-e, --encrypt
    Encrypt with given public key (NOT IMPLEMENTED)
--no-cert-publish
    Do not attach certificate to any signed message (NOT IMPLEMENTED)
```

1. At this point, your messages are being sent to the defined broker!

## consume2oracle

Current Status

## Specific Channel definitions

- GridPublisherSpecification - Publication of availability results to SAM
- GridPublisherSpecificationGridView - Publication of throughput + reliability data (gridftp, jobs) to Gridview

## Proposed usage

- Job Reliability Monitoring (ppt) [↗](#)

-- DanielRodrigues - 10 Jul 2008

---

This topic: LCG > MessagingSystemforGrid

Topic revision: r8 - 2009-11-05 - MassimoPaladin



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback