



## QCDGrid2 Metadata Tools Requirements Capture

**Project Title:** QCDGrid2

**Document Title:** QCDGrid2 Metadata Tools Requirements Capture

**Document Identifier:** QCDGRID2-WP2-Req

**Document Filename:** QCDGrid2-WP2Req.doc

**Distribution Classification:** Commercial In Confidence

**Authorship:** Daragh J. Byrne (DJB), George Beckett (MGB)

**Approval List:** QCDgrid Development Team

**Distribution List:** QCDgrid Development Team

### Document History:

<i>Personnel</i>	<i>Date</i>	<i>Summary</i>	<i>Version</i>
DJB	9 <sup>th</sup> June2005	Document released.	1.0
MGB	20 <sup>th</sup> June 2005	Minor update.	1.0.1

## Contents

<b>1. Introduction .....</b>	<b>2</b>
<b>1.1. The QCDgrid project .....</b>	<b>2</b>
<b>1.2. Document purpose .....</b>	<b>2</b>
<b>2. Requirements capture.....</b>	<b>3</b>
<b>2.1. Objectives .....</b>	<b>3</b>
<b>2.2. Methodology .....</b>	<b>3</b>
<b>3. Results .....</b>	<b>4</b>
<b>3.1. Description of data and metadata archived on the UKQCD Grid .....</b>	<b>4</b>
<b>3.2. Metadata management use cases.....</b>	<b>5</b>
<b>3.2.1. Use Case 1a: Place ensemble metadata in the Metadata Catalogue .....</b>	<b>5</b>
<b>3.2.2. Use Case 1b: Create ensemble metadata .....</b>	<b>6</b>
<b>3.2.3. Use Case 2a: Add one or more configurations to the grid .....</b>	<b>6</b>
<b>3.2.4. Use Case 2b: Transform a partial CMD to a complete CMD .....</b>	<b>7</b>
<b>3.2.5. Use Case 3: Withdraw a configuration from the grid .....</b>	<b>7</b>
<b>3.3. Gap analysis .....</b>	<b>8</b>
<b>4. Requirements.....</b>	<b>10</b>
<b>5. References .....</b>	<b>12</b>

# 1. Introduction

## 1.1. The QCDgrid project

The UKQCD Collaboration aims to “procure and jointly exploit computing facilities for lattice field theory calculations, whose primary aim is to increase the predictive power of the Standard Model of elementary particle interactions through numerical simulation of Quantum Chromodynamics”. Such numerical simulations produce significant amounts of data in the form of binary files. The purpose of the QCDgrid project is to provide a software application and supporting infrastructure that simplifies the management, storage and manipulation of this data.

In the first three years of the project (2002 – 2004), software engineers at EPCC developed a software application called *QCDgrid* – a data management system that combines the distributed resources of the collaborators into a robust facility called the *UKQCD Grid*. The result is a multi-terabyte storage facility over six UK sites at: Edinburgh (including the University of Edinburgh Advanced Computing Facility), Liverpool, RAL, Southampton, and Swansea. Glasgow is also a member of the consortium.

The facility is based on commodity hardware and open-source software. The hardware consists primarily of high specification PC-based servers running the Linux operating system and managing large RAID storage arrays. On top of this infrastructure, the QCDgrid software (built with Globus Toolkit 2.4, EGEE, and an XML Database Server (XDS)) provides *Datagrid* management and user functionality – furnishing a simple and intuitive environment that hides the complexities of the underlying grid and presents a standard file system to the user. It incorporates a robustness metric that automatically disperses datasets across the grid, providing a resilience that ensures data is not affected by the loss of one (or possibly more) storage nodes.

QCDgrid allows the user to query and manipulate associated metadata using a *Metadata Catalogue Browser*. The software also provides a *Job Submission System* that allows a user to schedule computations on remote HPC systems, from the comfort of their desktop computer. Security is leveraged from the Globus Toolkit, based on digital certificates issued by the UK e-Science Certificate Authority. The result is a reliable, secure data management system.

Looking to the future, the collaboration aims to integrate the UKQCD Grid with similar activities in the International Lattice Data Grid (ILDG), allowing like-minded scientists around the world to share their data and benefit from the scientific progress of other groups.

## 1.2. Document purpose

This document records the results of a requirement capture exercise carried out for Work Package 2 of the project – focusing on metadata creation and management tools. The exercise has been conducted by the QCDgrid project team at EPCC, in conjunction with members of the UKQCD collaboration. Section 2 describes the objectives and methodology of the requirements capture exercise. Section 3 begins with an appraisal of the structure and usage of metadata on the UKQCD Grid. The section also identifies five use cases that express the preferred model of usage of metadata on the Grid, and highlights the discrepancies with this preferred model and the current situation. Based on the results collected in Section 3, Section 4 establishes a realistic set of prioritised requirements for the work package.

## 2. Requirements capture

The requirements capture exercise occurred in late April and early May 2005.

### 2.1. Objectives

The objectives of the requirements capture exercise were as follows:

- Allow the QCDgrid project team to fully understand the structure of the metadata used by QCD scientists;
- Allow the project team to understand the processes required to create and manipulate the metadata during the day to day activities of QCD scientists;
- Having understood the above processes, examine the gaps in functionality in the existing software that prevent them being carried out effectively;
- Define what is required, in terms of new software and enhancements to existing software, to bridge these gaps.

### 2.2. Methodology

The requirements capture exercise consisted mainly of face-to-face meetings between members of the UKQCD community and members of the EPCC QCDgrid2 project team. In total three meetings were held, with representation from EPCC (Daragh Byrne, George Beckett and James Perry) and UKQCD (Chris Maynard, Balint Joo, and Zbigniew Sroczynski).

The first meeting considered largely the structure of the metadata describing the output of QCD simulations. The second and third meetings considered the process of creating and manipulating such metadata. Based on the output from these meetings, a draft of this document was produced and presented to the above participants for comment. Comments were also solicited from a number of other members of the UKQCD collaboration and, based on the feedback received, a final release of the document made.

In order to streamline the finer details of the requirements capture and implementation, it was agreed that Chris Maynard would act as primary user for the work package. All enquiries and questions are submitted to Chris for either an immediate response or referral to other users as appropriate.

## 3. Results

The following subsections describe the understanding of the EPCC project team of various aspects of the metadata management process.

### 3.1. Description of data and metadata archived on the UKQCD Grid

The UKQCD Collaboration primarily uses the UKQCD Grid to archive gauge configurations. A gauge configuration (from henceforth referred to as a configuration) is the output from a simulation run on either QCDOC or another HPC resource, using a QCD code such as CPS or QDP/Chroma. A configuration consists of a single binary file.

A configuration is typically initially produced in a "raw" format and needs to be post-processed before it is ready for archival on the UKQCD Grid. This post-processing involves transforming the raw data into the ILDG Binary File format. This post-processing is the responsibility of the UKQCD Collaboration and is beyond the scope of the QCDgrid project. In this document, any reference to a configuration pertains to the post-processed form of the configuration file, as mandated by the ILDG.

Individual configurations are collected into families, called ensembles. Configurations that are part of a single ensemble share common physical parameters that naturally suggest this relationship. The concept of an ensemble is purely logical – an ensemble has no explicit physical presence on the UKQCD Grid beyond the individual configurations that are members of it (however, as noted below, descriptions of ensembles are stored in the UKQCD Metadata Catalogue). Every configuration is a member of one and only one ensemble.

Each configuration that is archived on the UKQCD Grid is identified by a logical filename (LFN<sup>1</sup>). The LFN for a configuration is defined by the owner of the configuration, following constraints defined by UKQCD and the ILDG. The two important constraints on an LFN, from the perspective of the QCDgrid project, are that it is unique (identifies one and only one configuration) and persistent (will never be used to identify any other configuration).

When choosing the LFN for a configuration, the user typically collects configurations belonging to a particular ensemble into a directory, using a form for the LFN that is analogous to a UNIX directory path. For example, the following toy LFNs represent two configurations that both live in an ensemble directory `/a/b/c`:

```
/a/b/c/configuration-1
```

```
/a/b/c/configuration-2
```

This form of LFN helps the user to identify the ensemble to which a configuration belongs. Furthermore, the existing `put-file-on-qcdgrid` command-line tool allows the user to submit multiple configurations to the Data Grid by specifying only the common directory component of the LFN for the configurations to be committed.

Each ensemble is identified by an ensemble URI<sup>2</sup>. The ensemble URI is prescribed by the originator of the ensemble, subject to constraints defined by the ILDG. As with the configuration LFN, the two most important constraints on an ensemble URI, from the perspective of the QCDgrid project, are that it is unique and persistent.

---

<sup>1</sup> There is a discrepancy between the notations used by different members of the ILDG: UKQCD refer to the configuration identifier as a logical filename (LFN), while some other collaborations refer to the identifier as a global filename (GFN). Currently, configuration metadata is marked up using LFN, though in the future this reference may be replaced by GFN.

<sup>2</sup> A Uniform Resource Identifier (more commonly referred to as a URI) is a simple naming convention defined by the World Wide Web Consortium that allows resources (typically published on the internet) to be uniquely identified. One of the most common examples of URI is the Uniform Resource Locator (URL) that is used to identify files/services/hosts on the World Wide Web. See [3] for more details.

For every configuration stored on the UKQCD Grid, a single XML document that contains metadata describing that configuration *should* exist. For every (logical) ensemble stored on the UKQCD Grid, a single XML document that contains metadata describing the properties of that ensemble *should* exist. Each such XML metadata document is archived in the UKQCD Metadata Catalogue.

The XML document associated with an ensemble – henceforth referred to as an Ensemble Metadata Document (EMD) – should be valid with respect to a version of the QCDML ensemble schema. The ensemble described in a particular EMD is identified by the ensemble URI, which is included as a field in the EMD.

The XML document associated with a configuration - henceforth referred to as a Configuration Metadata Document (CMD) - should be valid with respect to a version of the QCDML gauge configuration schema. The configuration described in a particular CMD is identified by the LFN, which is included as a field in the CMD. Furthermore, the ensemble to which a particular configuration belongs is also recorded in the CMD, using a field that records the corresponding ensemble URI.

At the time of writing, the current version of the QCDML ensemble schema is Version 1.1 [2]. For the purpose of this work package, it is assumed that any future revisions to the schema will preserve the presence and form of the ensemble URI field.

Furthermore, at the time of writing, the current version of the QCDML gauge configuration schema is also Version 1.1 [2]. For the purpose of this work package, it is assumed that any future revisions to this schema will preserve the presence and form of both the LFN and the ensemble URI fields.

## 3.2. Metadata management use cases

The following use cases have emerged during the requirements capture process. Note that these use cases are ordered according to perceived importance, rather than chronology.

The process of creating and maintaining ensembles and configurations is covered by five related use cases, as follows:

1. Create an ensemble and add to the grid.
  - a. Place ensemble metadata in the Metadata Catalogue;
  - b. Create ensemble metadata;
2. Add a newly-created configuration file, and its metadata, to the grid.
  - a. Add configuration to the grid;
  - b. Transform partial CMD to complete CMD;
3. Withdraw a configuration from the grid.

These use cases are now described, in turn.

### 3.2.1. Use Case 1a: Place ensemble metadata in the Metadata Catalogue

This use case describes the steps that are taken when placing a complete EMD in the Metadata Catalogue.

#### Inputs

A completed EMD, possibly the output of some tool designed to automate use case 1b below.

#### Process

1. Confirm that the EMD is valid with respect to the appropriate QCDML Schema document.
2. Inspect the input EMD to determine the ensemble URI;
3. Consult the Metadata Catalogue to ensure that no EMD containing the ensemble URI already exists;
4. If step 1 and 3 are passed, then add the EMD to the catalogue.

#### Outputs

An indication of whether the submission process has been successful.

#### Notes

This is currently carried out using the eXist command-line client tool, which does not check for the previous existence of ensemble URIs (step 3, above).

This could most easily be implemented by extending the current metadata browser GUI client, although a scriptable interface may also be desired.

### 3.2.2. Use Case 1b: Create ensemble metadata

This use case describes the steps that are taken to create the metadata document associated with an ensemble.

#### Inputs

1. Complete set of ensemble metadata for insertion into the EMD.
2. An Ensemble URI.

#### Process

Using the inputs, create and populate an ensemble metadata document that describes the ensemble.

#### Outputs

A well-formed EMD that is valid according to the current version of the QCDML ensemble schema, which contains the correct physics and management metadata and ensemble URI, and is ready to be placed in the Metadata Catalogue as per Use Case 1a.

#### Notes

The matter of choosing the ensemble URI is beyond the scope of this project, and is addressed separately by UKQCD/ILDG.

This use case requires that all metadata is available at the beginning of the process, as the document is created in a single user session.

### 3.2.3. Use Case 2a: Add one or more configurations to the grid

This use case describes the steps that are taken when adding one or more configurations to the Data Grid.

#### Prerequisites

1. Existence of an ensemble to which the configuration(s) is (are) to be added.

#### Inputs

1. An LFN for each configuration to be added;
2. A corresponding set of complete configuration files, in ILDG format<sup>3</sup>;
3. A corresponding set of complete CMDs describing the configurations, which are well formed and valid according to a QCDML configuration schema document;
4. The ensemble URI of the ensemble to which the configurations belong.

#### Process

1. The Metadata Catalogue is checked to ensure that the specified ensemble exists;
  - a. If so, the following steps are taken;
2. Each CMD is validated against the appropriate schema document, as identified within the CMD.
3. The Metadata Catalogue is consulted to ensure that no CMDs identified with the same LFN as one of the inputs exist in the Metadata Catalogue;
4. A sanity check (details, to be determined) is performed on each CMD to determine whether each configuration is consistent with other configurations in the ensemble. If any of these

---

<sup>3</sup> The form of each configuration file is not checked so, in fact, the ILDG formatting of the file is anticipated, though not required.

sanity checks fail, a warning is produced and the user is asked to confirm they wish to continue with the submission.

5. The Data Grid is consulted to ensure that no configurations having the same LFN as one of the inputs exist on the grid;
6. If steps 2, 3, 4 and 5 succeed, then the CMDs are added to the Metadata Catalogue and the corresponding configurations are added to the Data Grid
7. The ensemble management metadata is updated to reflect the fact that the configuration has been added.

### Outputs

An indication of whether any of the parts of the process failed.

### Notes

Functionality to execute much of this use case exists in the metadata browser and the command-line tools. The current implementation is lacking in the following respects:

1. The Metadata Catalogue is not consulted to see if a CMD with the same LFN exists prior to a document being submitted;
2. The ensemble metadata is not updated after the configuration is submitted to the grid;
3. At present, it is possible to submit configurations without submitting corresponding CMDs by use of the command-line clients. This discrepancy should be removed and all tools should require both a configuration and CMD to be provided.

If the use case fails at any point, then the Data Grid and Metadata Catalogue should both be returned to their original state.

Step 7 of the above process may be dropped if, in a revision by the ILDG Metadata Working Group, the corresponding metadata elements are withdrawn from the QCDML Schema.

### 3.2.4. Use Case 2b: Transform a partial CMD to a complete CMD

This use case describes the steps that are taken when transforming a raw CMD as produced by QCD codes, to it finished state ready to be submitted to the grid.

#### Prerequisites

1. The raw output from QCD codes is transcribed into a partially populated QCDML configuration document.

#### Inputs

1. A partially populated CMD as produced by a QCD code or otherwise;
2. The ensemble URI of the ensemble to which the configuration associated with the CMD belongs;
3. The LFN of the associated configuration;
4. The CRC-32 checksum for the associated configuration;
5. Initial management metadata relating to the configuration.

#### Process

The partially populated CMD is enriched with the additional inputs to produce a complete CMD that is valid with respect to the appropriate version of the QCDML configuration schema.

#### Output

A complete CMD containing all important metadata, which is ready to be placed in the Metadata Catalogue.

### 3.2.5. Use Case 3: Withdraw a configuration from the grid

This use case describes the process that is undertaken whenever a configuration needs to be withdrawn from the UKQCD Grid.

#### Inputs

1. The LFN of a configuration that is to be withdrawn from the Grid.

**Process:**

1. Check that a CMD exists in the Metadata Catalogue corresponding to the LFN provided.
2. If a CMD exists, then update the configuration metadata corresponding to the data file to reflect the withdrawn status of the document.
3. Update the EMD of the parent ensemble to reflect the fact that the configuration has been withdrawn.

**Notes:**

Although the configuration has been withdrawn from the archive, neither the configuration data file nor corresponding configuration metadata document are physically deleted from the Data Grid/Metadata Catalogue. At some later time, it is perceivable that the configuration will be deleted from the Data Grid. This occurrence is outwith the scope of this work package.

Step 3 in the above process may be dropped if, in a revision by the ILDG Metadata Working Group, the corresponding metadata elements are withdrawn from the QCDML Schema.

### 3.3. Gap analysis

The use cases described in section 3.2 suggest a set of fundamental tasks that be carried out when dealing with QCD data. These tasks can be divided into those associated with metadata (configuration and ensemble), and those associated with configuration data. An analysis of which of these tasks is possible using the current software, and those which are not possible or partially possible, will be useful in determining requirements for future work. Firstly, we list tasks associated to the creation/modification of the content of the metadata:

- **Create ensemble** metadata;
- **Create configuration** metadata;
- **Edit configuration** metadata – specifically:
  - insert the LFN of the configuration;
  - insert the CRC-32 checksum;
  - insert the ensemble URI of the parent ensemble.

Secondly, we list tasks associated to the archival of metadata in the Metadata Catalogue:

- **Add configuration** metadata to the grid;
- **Add ensemble** metadata to the grid;
- **Update ensemble** metadata; specifically –
  - update an ensemble metadata document when a configuration is added;
  - update an ensemble metadata document when a configuration is withdrawn.
- **Update configuration** metadata; specifically –
  - update configuration metadata when a configuration is withdrawn from the grid.

At present:

- Neither update metadata process is automated.
- The process for creating an EMD is ad-hoc – it is completed using either a text or XML editor.
- The process for adding ensemble metadata to the grid is ad-hoc – it is currently added directly to the XML database using the database clients.
- The process of creating metadata for a configuration is completed in two stages. A partial CMD is created during the configuration generation process. Some fields within this partially populated document need to be completed at a later point. At present, this is done by hand, using a text or XML editor.
- The process for adding configuration metadata to the grid needs to be rationalised. Currently, when data is added to the grid via `put-file-on-qcdgrid`, there is no requirement to add

corresponding metadata. However, when adding data via the GUI client, there is a requirement to also add corresponding metadata. This inconsistency should be eliminated by requiring a CMD to accompany each configuration for both methods of submission.

## 4. Requirements

In this section, we present a list of requirements for implementation within the work package. These are determined from a prioritisation of the five use cases in Section 3, agreed with UKQCD, as follows:

- 1a: Place ensemble metadata in the Metadata Catalogue.
- 2a: Add one or more configurations to the Data Grid.
- 2b: Transform a partial CMD to a complete CMD.
- 1b: Create an EMD.
- 3: Withdraw a configuration from the grid.

Combining the gap analysis with the prioritised list above, determines the following requirements. Each requirement includes some initial commentary on its plausibility.

**R1** *A mechanism for adding an EMD to the Metadata Catalogue MUST be provided.*

This would most likely be implemented by extending the Metadata Browser Tool, although other options exist.

**R2** *The deficiencies described in the notes to Use Case 2a MUST be removed from the system. This will involve adding several layers of validation to the existing software, including:*

- Ensure that a CMD document is valid with respect to the QCDML schema when submitted to the catalogue;
- Check that a CMD with the same LFN does not exist in the Metadata Catalogue before a new CMD is added;
- Ensuring corresponding EMD is updated when a configuration is added to an ensemble.

**R3** *The `put-file-on-qcdgrid` tool SHOULD be modified to require that a CMD be submitted with each configuration.*

There are two issues with this requirement. Firstly, a format that identifies the CMD filenames with corresponding configuration files, when a user submits a directory of configurations, is to be determined. Secondly, as the command-line software is programmed in the C language – the plausibility of connecting to the Metadata Catalogue from a C language program is to be determined.

**R4** *A mechanism to allow partially completed CMDs to be converted to complete CMDs MUST be provided.*

The options for fulfilling this requirement are as follows:

1. Utilise existing software, including:
  - a. a text editor (for example, GNU Emacs)
  - b. a dedicated XML editing tool, such as XMLSpy by Altova [4].
2. Develop a custom software tool for editing CMDs, such as a Java GUI application or web-based form

In the case of (2), a further more detailed requirement capture exercise may be necessary.

**R5** *A mechanism for creating EMDs SHOULD be provided.*

The options for fulfilling this requirement are as follows:

1. Utilise existing software, including:
  - a. a text editor (for example, GNU Emacs)
  - b. a dedicated XML editing tool, such as XMLSpy by Altova [4].

2. Develop a custom software tool for creating EMDs, such as a Java GUI application or web-based form.

In the case of (2), a further more detailed requirement capture exercise may be necessary.

**R6** *A tool for updating configuration metadata and ensemble metadata to reflect the withdrawal of a configuration from the UKQCD Grid SHOULD be provided.*

This tool would allow use case 3 to be carried out.

## 5. References

- [1] UKQCD Collaboration home page <http://www.ph.ed.ac.uk/ukqcd/> (June 2005).
- [2] The QCDML schema and tutorials are available from [http://www.ph.ed.ac.uk/ukqcd/community/the\\_grid/QCDml1.1/](http://www.ph.ed.ac.uk/ukqcd/community/the_grid/QCDml1.1/) (May 2005).
- [3] The World Wide Web Consortium, *Uniform Resource Identifier (URI): Generic Syntax*, electronic document available from <http://www.gbiv.com/protocols/uri/rfc/rfc3986.html> (January 2005).
- [4] XMLSpy by Altova [http://www.altova.com/products\\_ide.html](http://www.altova.com/products_ide.html) (May 2005).