LHSG-CT-2003-503420

BioXHIT

A European integrated project to provide a highly effective technology platform for Structural Genomics.

Life Sciences, Genomics and Biotechnology for Health

WP5: Ms 5.2.4 Proposed Extensions to the CCP4i def file tracking database

Due date of deliverable: **31.03.2007**
Actual submission date: **31.03.2007**

Start date of project: **1.1.2004** **Duration: 60 months**

Organisation name of lead contractor for this deliverable: CCLRC-CCP4
Daresbury Laboratory, Warrington WA4 4AD UK **Author** Peter Briggs

Identification of additional functionality for the extended CCP4i def file tracking database (BIOXHIT milestone 5.2.5)

Peter Briggs, CCP4

1 Introduction

This report describes proposed extensions to the initial implementation of the tracking database. The proposals came from consultations with potential and actual users of the system, and are intended to improve the usefulness of the system for recording the history of structure solution projects for a broader range of systems than just CCP4i.

A subset of extensions will be implemented as part of deliverable D 5.2.10, reported separately.

2 Additional data items

This section outlines the set of suggested data items that could be added to the current tracking database.

Proposed Data	Description
Project title	A short user-supplied description of the project
Project description	A longer description (like an abstract) that could be added by a user. Both the project title and description could be useful if the user has a lot of projects or is able to share them with others
User-agent/driver application	The name of the client application program that acts for the user when making changes to the database - for example, for jobs run in CCP4i the user agent would be "ccp4i", for jobs run by XIA2 it would be "xia2" and so on. Some kind of versioning information could also be included.
Subjobs/substeps	these would be smaller jobs within a single larger job. It is likely that for example a single run of an automated pipeline would be a "job" but that the automated process would explicitly divide this run into smaller jobs
History data	History data is not stored explicitly in the tracking database; it is derived implicitly based on filenames. Explicit storage of history links between jobs would be better. There would be two types of link: • **Data link**: when job X uses data produced from job Y • **Logical link**: when job X follows on from from job Y due to application logic (i.e. the logic encoded in a script or other system, which determines that one step follows another even if there is no apparent data flow - but it could also apply to "procedural logic", when a human user follows a procedure in which the steps are linked by some logical scheme. It is possible that we might also wish to store inferred links (which is what is generated by the ccp4i.history class at the moment) and broken links (links that have been explicitly removed by the user).
Application control file	Currently the name and location of the CCP4i parameter file is not explicitly stored; it is generated as jobid_taskname.def (e.g. 123_scala.def) and is stored in the CCP4_DATABASE subdirectory of the project. To extend the tracking to other applications means

	explicitly storing this data, or at least its location.
Logfile	Similar to application control files, in CCP4i the logfile location isn't explicitly stored at present. The name of the logfile is generated as job_id_taskname.log (e.g. 123_scala.log) and is stored in the project directory
Notebook	Similar to the application control and log files above. The notebook data is stored in a file with the name generated as jobid_notebook.txt (e.g. 123_notebook.txt.
Tags	Tags would allow users or client applications to associate one or more arbitrary keywords with particular jobs. These could be used for selection purposes by other functions.
Operation type	This would allow the application to specify the type of operation that a job represents, and allow program or script runs to be distinguished from reported jobs or editing operations.

3 Additional functionality

This section describes the additional functionality suggested for the tracking database.

Proposed Function	Description
Manipulating projects	A number of suggestions related to manipulating projects: • Allow jobs to be moved between projects • Allow import/export of projects • Allow projects to be split or merged • Allow branched projects to be synchronised

4 Selection and implementation of new features

It is the work associated with deliverable De 5.2.10 which will select which functions to implement, based on an assessment of priority and usefulness.