

RootCnvSvc Code Review

July 7 & 13, 2004

RootCnvSvc and RootSvc Package Author: Ursula Berthon

Reviewers: Joanne Bogart, Toby Burnett, David Chamont, Heather Kelly, Tracy Usher

Introduction

This is the first review of the new packages: RootCnvSvc and RootSvc. These packages are meant to replace RootIo as the means to handle I/O to/from the TDS to ROOT.

RootCnvSvc uses the Gaudi Conversion Service that provides greater flexibility in terms of controlling what is and is not output to files for each run.

We would like to thank Ursula Berthon for her efforts on this project. Having joined GLAST temporarily for about a year and half, Ursula learned Gaudi and the GLAST Offline software, despite a lack of good documentation. Her hard work is very appreciated and we look forward to utilizing this new package.

Documentation

The existing documentation is a good start at providing detailed information for future developers and maintainers of this code. There are few areas where some improvements would be helpful.

Doxygen Documentation

Most methods do have an associated Doxygen comment in the header file, as requested; however, the comments could be more explicit. For example, in the RootSvc.h file, each method is commented, such as:

```
/// add input tree to the list
virtual StatusCode addInput(const std::string& treename, const
std::string& file);
```

More details would be helpful such as, what list is the tree being added to. What is input in this context? A Doxygen comment for the whole RootSvc class is missing, and would provide an area to explain some of the details concerning this class.

Other Documentation

Adding a general overview of the conversion service and its converters to the mainpage would be helpful. One or two UML diagrams demonstrating how the parts fit together would aid future developers. A pointer to the Gaudi documentation should be added to allow easy viewing of the background documentation concerning converters and conversion services.

There are areas in the source code where it would be useful to add a comment or two concerning what blocks of code are doing. For example, in the AcdDigiCnv.cxx more guidance concerning how the conversion works should be added to aid future maintainers.

Code Review

Some details of the code were discussed.

It was decided that RootSvc would be merged with RootCnvSvc. There is only one class in RootSvc now, as it was decided that it should not be a Gaudi algorithm. At this time, it seems unlikely that another package can re-use the functionality provided by RootSvc and even if it does – it is a very small amount of code. In addition, the RootSvc class will be renamed RootUtil, as it is no longer a Gaudi service.

There was some discussion concerning exchanging the vectors used in RootSvc to keep track of TFile and TTrees for maps instead. This would allow the code to retrieve the TTrees using their names – which would be clearer than utilizing a contrived index number. Additionally, it would be nice to automate the retrieval of the TTree names from the ROOT files, rather than forcing the users to explicitly provide the TTree name in the jobOptions parameters.

There was a brief discussion concerning RootCnvSvc's use of three specific converters: McEventCnv, DigiEventCnv, ReconEventCnv. It should be noted that this is not typical Gaudi behavior. The conversion service generally does not need to call specific converters in this fashion. This is a special case. The ROOT objects need to be cleared after their data has been written to file. The Gaudi conversion service interface does not provide a clean-up method where each converter is called and is allowed to clean itself. To handle this, the data objects are cleaned at the end of the commitOutput method. To do so, requires calling the three converters explicitly – they are the ones that contain the ROOT data objects that require cleaning.

Before RootCnvSvc can fully replace RootIo, the following items need to be addressed:

- Complete testing on Windows and Linux
- Provide a mechanism to select what events are output to ROOT files i.e. triggered events.
- Allow for random event access given run/event ids.
- Add converters for the LDF TDS objects, such as LdfTime.
- Add converters for relations.

During the review, we discussed the timeline for some of these additions. Testing of the existing converters will be completed in the coming weeks. The selection criteria will be added at the end of August. Work on converters for the relations will begin in early September.

We discussed the possibility of creating a new package to share the internals of the conversion process between RootCnvSvc and RootIo. This would be helpful during the transition phase, since modifications to the existing TDS or ROOT classes will require modifications to code in two places. This discussion did not result in a decision. Some felt the time required to create a new package could be better spent on other activities. Others questioned how long this transition period would really be. If it were deemed useful to have all of the conversion details in its own package, then creating a new

package would seem to be justified at this time. The core group should further consider this issue.

It was asked if we knew how Gaudi determines the ordering of the calls for writing the output. Ursula has studied this and has determined that it is in inverse alphabetical order (based on the name of the converter or the object being converted?)

Action Items:

- Complete testing on Windows and Linux. Testing includes:
 - Compare the output files created with both RootIo and RootCnvSvc
 - Test reading of the conversion service by reading in a known ROOT file, and writing the contents back out to file. A comparison of the new and old ROOT file will demonstrate if the reading and writing occurred successfully.
- Fix requirements file in RootCnvSvc to work on Windows
- Merge the RootSvc package into RootCnvSvc and rename the class RootUtil.
- Add selection using the trigger to control what events are output to file.
- Add selection using run/event ids to allow for random event access when running the GUI and reading from ROOT files.
- Add converters for the relations.
- Add converters for the LDF TDS objects such as LdfTime and Diagnostic data
- Modify the scripts in RootCnvSvc to make them more re-usable
- Complete removal of all ^M characters from the source code.
- Consider alternatives to RootCnvSvc call specific converters directly to clear ROOT objects, such as utilizing the list of converters in the RootCnvSvc and providing a virtual method in the RootBaseCnv class.

Documentation Action Items

- Add one or two UML diagrams demonstrating dynamic behavior
- Add a reference to the Gaudi documentation concerning converters and conversion services.
- Document any recommendations for improving the TDS classes to make the conversion process easier – these recommendations should come from Ursula based on her experience writing the original converters.
- Bring the release.notes up to date noting what major modifications correspond to what tags.
- Add a general description of each jobOptions parameter to the mainpage, including the default values each parameter takes.

Core Group

- Determine if a new package should be created to contain the internals of the converters.
- Evaluate the current TDS and ROOT data classes.