

SoftRelTools version 2 at Fermilab

J. Amundson¹, for the Configuration Management Working Group

Computing Division, Fermi National Accelerator Laboratory, Batavia IL, USA

Abstract

We describe the rewrite and use of SoftRelTools version 2, which was developed as part of the Fermilab CDF/D0/CD Joint Offline Project. The new SoftRelTools is currently in use by CDF and D0 Online and Offline projects; it has also been adopted by BTeV. The goal of the rewrite was to be easier to use and to maintain than its predecessor while providing a backwards-compatible replacement for the original SoftRelTools written by Bob Jacobsen. Maintainability is enhanced by a modular structure and the separation of the basic tool from project-specific and site-specific settings through an inheritance hierarchy. Ease of use has also been improved through increased functionality. The new version of SRT is proving robust against the experiment development needs as they move to new compilers, new options, and new platforms.

Keywords: configuration management, build tools

1 Introduction

SoftRelTools was originally written for the BaBar experiment by Bob Jacobsen. Its use at Fermilab began in 1997. It is currently being used at the D0, CDF and BTeV experiments. The Fermilab version of SoftRelTools has not been kept in sync with the BaBar version.

Although the use of SoftRelTools at Fermilab has generally been considered a success, maintenance problems have grown over time. There were three main sources of problems. First, the code changed often, on the order of once per day. This high rate of change can be largely attributed to the rapidly changing needs of a large group of active developers. Second, the need to accommodate the needs of several different experiments had turned the code into a spaghetti-like series of if statements. Thirdly, the code carried long-distance correlations between seemingly unrelated sections.

In order to address these problems, the decision was made to produce a completely rewritten version of SoftRelTools. The goal was to enhance maintainability while preserving the well-established and well-tested features of the original; the rewrite was to be a conservative rewrite. The interface was (largely) preserved. The original implementation was in Bourne shell scripts and GNU make files as an interface to cvs modules. These features were unchanged. Nonetheless, nearly every line of code changed between the old and new version. In this paper I describe the features of the newly rewritten SoftRelTools and discuss how these features address the problems of the previous version.

2 Modular Structure

The conceptually simplest problem to address was the long-distance correlations in the code. SoftRelTools has three main functions: code management, providing a build system, and providing an

interface to external packages. The code management system acts as an interface to cvs as well as providing mechanisms for mixing and matching versions between modules. The build system provides standardized rules for building libraries and executables. The external package interface provides a standardized interface to such things as TCL, Motif, GEANT, etc.

In the SoftRelTools version 2, these three functions are considered different modules. The interface between them is defined to be the environmental variables they share. (In addition, the code management and build systems talk to each other through a single script.) The division itself is mostly conceptual; very few changes to the original code would have been required to accomplish it. Nonetheless, it has greatly enhanced maintainability. Changes can now be made, e.g., to the build system without worrying if there will be any consequences for the code management system because the interface between the two has been well defined.

3 Inheritance hierarchy

The inheritance hierarchy of SoftRelTools version 2 addresses both the spaghetti-code problem and the rapid-change problem of the old version. The idea is to apply the ideas of object-oriented design, even though the tools used, Bourne shell and GNU make do not lend themselves easily to the object paradigm. The implementation of the inheritance hierarchy in these tools is the novel feature of the rewrite.

The hierarchy itself looks like this:

SoftRelTools base → Project-specific → Location-specific. (1)

The inheritance hierarchy supports two concepts from object-oriented design, *specialization* and *overrides*. The project-specific portion of SoftRelTools can specialize the behavior of the base SoftRelTools; the location-specific behavior can, in turn, specialize the behavior of the project-specific part. Where specialization fails, overrides are available. The difference is whether the new behavior augments or replaces the behavior of the original.

The implementation of these ideas is through packages. A project XXX can have a package SRT_XXX. An installation of SoftRelTools can also have its very own SRT_LOCAL package. Specialization is achieved in shell scripts through the use of functions and optional includes. To specialize the behavior of a script, one only needs to write a new function and place it in the appropriate place in SRT_XXX or SRT_LOCAL. It will be read last, replacing the definition of the function in the base SoftRelTools package. In similar fashion, each make fragment in SoftRelTools optionally includes a corresponding file allowing extra functionality.

Overrides are achieved primarily through the binary-search path (PATH) for shell scripts and the include path for make fragments. Portions of the SRT_XXX and SRT_LOCAL directories are always searched before the base SoftRelTools. For future maintenance, the specialization option is preferable to the override option. Some behaviors are too difficult to specialize, however, so the override option is always available.

4 Contact information

The manual for SoftRelTools version 2 is available on the web at <http://RunIIComputing.fnal.gov/cmgt/SoftRelTools-Manual/SoftRelTools-Manual.html>. A link to download the software is on the same page. Questions about SoftRelTools should be addressed to the SoftRelTools mailing list, <mailto:softreltools@fnal.gov>.

This work is sponsored by DOE contract No. DE-AC02-76CH03000.