

Using SCM and SAS/Warehouse Administrator[®] software together to provide Version Control for data warehouse processes

The Source Code Manager (SCM) component of SAS/AF[®] software can be used to good effect with code generated by SAS/Warehouse Administrator software to provide a flexible Version Control environment for data warehouse processes. However, to do this requires some discipline. This short paper describes a technique for using the two together. It is based on the procedures used on a major data warehouse project.

This technique is now the subject of a paper at SeUGI 19, Florence. The full paper will be published on the Applied System Knowledge website after presentation at SeUGI.

Steve Morton
© Applied System Knowledge Ltd.
19 April 2001

Introduction

When running data warehouse processes in a production environment, it is often important to be able to "freeze" a version of process code and jobs/scripts each time a "new edition" of a data warehouse is created. A "new edition" is created whenever the *structure* of the data warehouse is altered, either by adding new tables/structures or changing existing ones, or just by changing *processes* by which the data is extracted/transformed/loaded. A structure change naturally requires a process change, so both types of enhancement will add or change process code. Loading new data into the same processes does not constitute a "new edition" under the terms of this definition.

By creating a frozen version each time processes are changed, it is easy to look back at historical versions of data warehouse processes. This becomes important over the long-term life of a data warehouse, and may be mandatory in some customer environments for audit purposes.

When SAS/Warehouse Administrator software is used to define data warehouse management processes you have a powerful tool to create that code. However, SAS/Warehouse Administrator software does not maintain historical versions of the 'meta-process'. When Version Control of data warehouse process is required, it is necessary to use some other form of management; this is available in the SAS system by using the SCM component of SAS/AF software. This has the advantage over external version control that source can be maintained in SAS catalogs, reducing the proliferation of loose source files that can otherwise result.

The principle is that all unit-tested code should be generated into SAS catalog entries that are then 'checked in' to the relevant SCM team library. This is then the source for Integration test and System test, which will (after completion of testing) be promoted as a Production version. Whenever a Version Label is defined in SCM, you are freezing a copy of code that you can return to at any point in the future (subject to retaining enough archive levels - see below).

Since SAS/Warehouse Administrator software is often run on PC desktop systems to create and manage data warehouse structures on larger corporate servers (Unix, NT or OS/390), the Version Label is the natural base from which to transfer code to the production environment

Setting up SCM

See SCM documentation for the basic installation/set-up of SCM, and the creation of a Control Database (CDB). In general you should configure SCM to use unlimited archive levels (or a large number); the default is 15 - consider whether this will be enough for you, my advice is at least 50 levels but 'unlimited' is best. Once the basic set-up has been done, you will need to create team libraries and archive-space for each source library that you will need. Follow these steps:-

1. Identify each SAS library into which you will generate code from SAS/Warehouse Administrator. These will be considered to be your 'Local' libraries, which each member of the warehouse administration team will use to unit-test new or changed process code.
2. For each of these, create a corresponding SAS libref (and directory) as a 'Team' library; these should be defined using SAS/SHARE[®] 'remote' librefs on the development platform, to avoid lockouts later. *Tip: Adopt a consistent naming convention for these librefs to help identify which Team library corresponds to which Local library. For example, prefix each libref with X (which will place all the Team librefs together alphabetically) - but if the Local libref names are more than 7 characters take care with the Team library names used.*
3. Using the Administration Wizard of SCM, add each Team library to your SCM CDB. You will also be asked to identify an archive path for each library; place this on a networked drive, in a directory that everyone is permitted write access.

4. Each user of SAS/Warehouse Administrator software should then create their own 'Local' librefs and directories, and then use the 'Library Mappings' tab in their SCM Options window to map each Team library to the corresponding Local library.
5. Finally, remove the Local and Team libref definitions from personal autoexec or other standing definitions. Best practise is to rely on SCM to allocate those libref's for you, which it does when starting SCM; by removing non-SCM definitions you will require SCM to be run whenever using those libraries, and help ensure the use of SCM versioning. You will need to pre-allocate the SCM CDB, since that is where the shared definitions reside, and whatever library is used by each person for their SCM Profile (usually SASUSER).

Using SCM with SAS/Warehouse Administrator day-to-day

Once set up, you can use SCM routinely, but this requires each administrator to stick to some manual procedures as follows:-

1. When creating or modifying some data warehouse process, each member of the administration team will perform Unit Testing of their new process code. For an existing process, they should 'Check-out' the relevant SCM objects from the correct Team library - usually the individual source entries, but possibly the whole catalog. For a new process, these objects will not yet exist - they will be added to SCM after they have been created (see step 3 below).
2. Code generated at this time should be stored into the required Local library catalog (locked by check-out in step 1 above, or new); this can be done either before running the Unit Test (using the catalog source code to run the test) or immediately after (if direct test execution from SAS/Warehouse Administrator is preferred).
3. After satisfactory Unit Test, 'Check-in' the generated code objects (or 'Add' the new ones) to the Team library. Repeat steps 1 to 3 for all the Unit Test process code required, until you are ready for Integration test.
4. For Integration Test, create a 'Version Label' in SCM for each code library to be tested; then use 'Copy Version Label' to create the Integration Test master copy. If Integration Test is to be done on a different platform, this is the base from which to upload the code to the test environment. If defects are found, create an "Integration Test version x.01" version label to distinguish changed entries, then create modified code within SAS/Warehouse Administrator (follow steps 1 to 3 above again for defect resolution) until the code has passed Integration Test.
5. For System Test, create the next 'Version Label' in SCM for each library to be System Tested. Proceed as at step 4 above while performing System Test and rectifying any further defects.
6. For Production, create a final Production 'Version Label' in SCM, and use 'Copy Version Label' to create the Production master. Code from this will then be uploaded to the production platform where it will routinely run using the system scheduler. At this point, it is good practise to also take a copy of the SAS/Warehouse Administrator metadata that corresponds to the Production process version; this can be most easily done in version 2.0 or later by using the SAS/Warehouse Administrator desktop Copy feature.

The advantage of following this scheme is that SCM archiving automatically stores previous copies of entries or catalogs whenever you check in a new version. Combining this with Version Label - which references each code version in effect at the time - allows you to reconstruct the process code at any time in the life of the warehouse (subject to any archive limits you set in SCM) by copying out the relevant Version Label. It is not necessary to retain those copies outside SCM, since they can always be recreated.

Documentation

SCM is documented in the Help files for version 8 SAS - though it is not immediately obvious in the Contents. Use the 'Search' tab to find SCM, and you will reach the documentation. SCM was experimental with version 6.12 as a downloadable add-on, but has been production since version 7.

Note that SCM requires the use of SAS/SHARE software in the development environment, to support the use of a central shared CDB and for proper use of the Team libraries.