

| | |
|--|----|
| 1. eSTOMP Configuration Management Plan, Rev. 1.0 | 2 |
| 1.1 1.0 Introduction to eSTOMP Configuration Management | 2 |
| 1.1.1 1.1 Purpose | 2 |
| 1.1.2 1.2 eSTOMP Utilities | 2 |
| 1.1.3 1.3 Definitions | 3 |
| 1.1.4 1.4 References | 3 |
| 1.2 2.0 Configuration Management | 4 |
| 1.2.1 2.1 Organization and Responsibilities | 4 |
| 1.2.1.1 2.1.1 Developers for eSTOMP and Related Software | 4 |
| 1.2.1.2 2.1.2 Code Custodian (CC) | 4 |
| 1.2.1.3 2.1.3 Software Configuration Manager (SCM) | 5 |
| 1.2.1.4 2.1.4 Testers | 5 |
| 1.2.1.5 2.1.5 Training | 5 |
| 1.2.2 2.2 Additional Constraints | 5 |
| 1.3 3.0 Software Configuration Management Activities | 5 |
| 1.3.1 3.1 Communications | 6 |
| 1.3.2 3.2 Software Configuration Control | 6 |
| 1.3.2.1 3.2.1 Documenting Releases | 6 |
| 1.3.2.2 3.2.2 Placing Items under Software Configuration Management | 6 |
| 1.3.2.3 3.2.3 Processing Software Change Requests | 6 |
| 1.3.3 3.3 Software Configuration Identification | 7 |
| 1.3.3.1 3.3.1 Baselines Changes | 7 |
| 1.3.3.2 3.3.2 Identifying Configuration Items | 7 |
| 1.3.3.2.1 3.3.2.1 Identifying Software Documentation | 8 |
| 1.3.3.2.2 3.3.2.2 Software Source and Documentation Storage and Preservation | 8 |
| 1.3.3.3 3.3.3 Life-Cycle Model | 8 |
| 1.3.3.4 3.3.4 Project File Directory Structure | 8 |
| 1.3.3.5 3.3.5 Storage and Handling | 8 |
| 1.3.4 3.4 Software Configuration Management Assessments and Reviews | 9 |
| 1.3.5 3.5 Software Configuration Status Accounting | 9 |
| 1.4 4.0 Software Configuration Management Resources | 9 |
| 1.4.1 4.1 Personnel | 9 |
| 1.4.2 4.2 Software and Hardware | 9 |
| 1.5 5.0 SCMP Maintenance | 9 |
| 1.6 6.0 Software Configuration Management Schedules | 10 |
| 1.7 7.0 STOMP Roles and Responsibilities | 10 |

eSTOMP Configuration Management Plan, Rev. 1.0

Vicky Freedman

February 2015

PNNL-24121

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest National Laboratory
Richland, Washington 99352
This document maintained electronically at

<https://spcollab.pnnl.gov/sites/stompshare/Shared%20Documents/Forms/AllItems.aspx>

Approvals

This document is electronically approved and signed via the PNNL Electronic Records & Information Capture Architecture (ERICA) system.

Revision History

| Revision | Date | Summary of Changes |
|----------|---------|--------------------|
| 1.0 | 02/2015 | Original Version |

1.0 Introduction to eSTOMP Configuration Management

This document describes the software configuration (CM) management activities to be performed in support of the scalable version of STOMP (Subsurface Over Multiple Phases), known as eSTOMP. This Software Configuration Management Plan (SCMP) plan applies to all software listed in the Software Test Plan (STP), and is referred to as eSTOMP and related software in this document.

1.1 Purpose

This SCMP provides information on the requirements and procedures necessary for the software CM activities of the eSTOMP simulator. This SCMP identifies the software, hardware, and documentation requirements for software configuration management and establishes the methodology to generate configuration identifiers, manage engineering, scientific and input/output formatting changes, maintains status accounting, and perform assessments and reviews during requirements analysis, design, development, and maintenance of the Software Configuration Items (SCIs).

1.2 eSTOMP Utilities

In addition to the STOMP simulator, three perl scripts can be used to extract STOMP output into Tecplot, Gnuplot and Surfer formatted input files. An additional pre-processor is used to generate input for reactive geochemical simulations. All these processors are maintained under [Subversion \(SVN\)](#) control in the same manner that the STOMP source code is maintained.

1.3 Definitions

Glossary

| <u>Term</u> | <u>Definition</u> |
|-------------|--|
| CC | Code Custodian |
| CM | Configuration Management |
| CSA | Configuration Status Accounting |
| DOE | Department of Energy |
| OTD | Office of Technology Development |
| PNNL | Pacific Northwest National Laboratory |
| RIDS | Record Inventory and Disposal System |
| RL | Revision Log |
| SCI | Software Configuration Item |
| SCM | Software Configuration Manager |
| SCMP | Software Configuration Management Plan |
| SCR | Software Change Request |
| SDD | Software Design Document |
| SQP | Software Quality Plan |
| SRS | Software Requirements Specification |
| eSTOMP | Scalable version of STOMP |
| STP | Software Test Plan |
| SVN | Subversion (Subversion Control System) |

1.4 References

Balay, S., K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith and H. Zhang. 2004. PETSc User Manual. ANL-95/11 – Revision 2.3.2, Argonne National Laboratory.

Fayer, M.J. 2012. Software Quality Plan (SQP) for Subsurface Transport Over Multiple Phases (STOMP).

Nichols, W.E., N.J. Aimo, M. Oostrom and M.D. White. 1997. STOMP Subsurface Transport Over Multiple Phases Application Guide. PNNL-11216, Pacific Northwest National Laboratory, Richland, Washington. Ward AL, MD White, EJ Freeman, and ZF Zhang. 2005.

STOMP Subsurface Transport Over Multiple Phase Addendum: Sparse Vegetation Evapotranspiration Model for the Water-Air-Energy Operational Mode. PNNL-15465, Pacific Northwest National Laboratory, Richland, Washington.

PNNL SharePoint Site STOMPshare (<https://spcollab.pnnl.gov/sites/stompshare/>)

PNNL STOMP website (<http://stomp.pnl.gov>).

White M.D. and M. Oostrom. 2006. STOMP Subsurface Transport Over Multiple Phases Version 4.0: User's Guide. PNNL-15782, Pacific Northwest National Laboratory, Richland, Washington.

White, M.D., and M. Oostrom. 2000. STOMP Subsurface Transport Over Multiple Phase: Theory Guide PNNL-11216 (UC-2010), Pacific Northwest National Laboratory, Richland, Washington.

Zhang, Z.F., V.L. Freedman, G.D. Tartakovsky and M.D. White. 2012. Requirements for Subsurface Transport Over Multiple Phases (STOMP).

2.0 Configuration Management

This section describes the software CM organization, allocation of responsibility and authority for software CM activities, and references the software CM policies and directives pertaining to the use of the eSTOMP simulator and related software (i.e. eSTOMP utilities).

eSTOMP and related software developed by PNNL for DOE use and its related documentation will remain under software CM as long as eSTOMP and related software is under continued development, in use for supporting site decision making, or further use of the software is contemplated.

2.1 Organization and Responsibilities

CM for the eSTOMP simulator and related software is designed to ensure clear lines of authority and to provide a framework within which administrative and technical control of development, enhancement and modification activities can be effectively integrated into a high-quality product for scientific research and analysis.

Primary authority for managing the eSTOMP software is vested in the SCM. The SCM also serves as the CC.

The Software Configuration Manager (SCM) and Code Custodian (CC) for the STOMP software is Vicky Freedman. PNNL's quality organization has identified Kary Cook to serve as the Software Quality Practitioner (SQP) for STOMP. The following table identifies project software maintenance team members.

Key Staff for STOMP Software Maintenance

| Role | Staff | Responsibilities |
|--------------------------------------|--|--|
| Software Configuration Manager (SCM) | Vicky Freedman | Decision authority to approve software changes |
| Code Custodian (CC) | Vicky Freedman | Software preservation (backup); software configuration management record keeping; source control |
| Software Developers | Diana Bacon, Yilin Fang, Vicky Freedman, Mart Oostrom, Mark Rockhold, Mark White, Signe White, Fred Zhang | Software source code development |
| Software Testers | Diana Bacon, Yilin Fang, Vicky Freedman, Mart Oostrom, Mark Rockhold, Guzel Tartakovsky, Mark White, Signe White, Xiaofan Yang, Fred Zhang | Software testing |

2.1.1 Developers for eSTOMP and Related Software

Developers for eSTOMP and related software are responsible for ensuring that changes to SCIs managed under this plan are only undertaken in accordance with the policies, procedures, and requirements of this SCMP. Developers must receive training on this SCMP before they are allowed to perform duties as implementers of changes approved in the software change control (SCR) process. Developers will implement changes for approved SCRs as assigned by the SCM. Developers may run applicable baseline tests, based on the STOMP mode they are running, but independent testing will be performed on their implementation of SCRs.

2.1.2 Code Custodian (CC)

The general responsibilities of the Code Custodian (CC) are to maintain and track changes made to eSTOMP and related software. Emphasis is placed on providing a traceable history of the software and a mechanism for recreating all releases.

The CC is responsible for the following:

- ensuring that the SVN code repository is established and maintained to support software and test suites
- assuring that an SCR log has been successfully completed to show the status of proposed, approved, and implemented changes
- configuration maintenance activities under this SCMP
- maintaining regression tests to be run when the baseline changes
- ensuring that backups of software, data, and records are maintained
- establishing the software configuration management system log and records

The STOMP/eSTOMP SharePoint will be used by the CC and the eSTOMP development and testing team for the following:

- ensuring that SCI naming and numbering conventions, as described in the *Software Configuration Identification* section are followed
- maintaining the original SCRs and Revision Logs
- maintaining QA training records
- maintaining a eSTOMP Users List

Procedures for managing change are defined in the section entitled *Software Configuration Management Activities*. The SCR form on the SharePoint site will be used to track the status and document the sequence of steps required to complete the configuration change for external users. Internally, the eSTOMP development team will use the JIRA software to track and document code changes.

2.1.3 Software Configuration Manager (SCM)

The general responsibilities of the SCM are to control changes in the software and documentation and to ensure testing and verification is performed for each released version and revision. Emphasis is placed on providing guidance on software development, and leading other software developers.

The SCM is responsible for maintaining management control of eSTOMP and related software by

- approving or rejecting all SCRs based on full consideration of change impacts and software needs
- assuring that all changes made under approved SCRs are properly implemented and independently tested
- controlling the release of SCIs.
- ensuring eSTOMP and related software developers are trained to the requirements, procedures, and policies of this SCMP before they can serve as change evaluators or implementers
- assisting in software configuration maintenance assessments and reviews, as required

2.1.4 Testers

The general responsibilities of the Testers are to test eSTOMP and related software and verify that SCRs are implemented based on new requirements. Testers are responsible for creating and implementing Test Plans prior to testing. Testers are also responsible for generating Test Reports with the results of the testing and uploading these documents to the eSTOMP SharePoint.

2.1.5 Training

Staff members charged with responsibilities under this plan will receive training to learn the requirements, procedures, and policies of this plan. The SCM is responsible for assigning training to eSTOMP team members. Completion of training will be documented and maintained on the STOMP/eSTOMP SharePoint site.

2.2 Additional Constraints

Identification of all files, documentation, media and software directories follow the procedures outlined in the *Software Configuration Management Activities* section. Any additional external constraints placed on this SCMP will be identified as they arise. These include additional client requirements, additional standards, or other constraints. Client awareness of and agreement to such constraints and their impact on the project will be obtained before they are integrated into this SCMP.

3.0 Software Configuration Management Activities

Software development and testing are performed to ensure the highest quality standards. Verification testing and configuration management provide assurance of correct and consistent results for selected test problems.

The primary means to minimize the risk of a software error of consequence are:

- Strict adherence to the STOMP/eSTOMP Software Quality Plan (SQAP),
- Strict adherence to this SCMP,
- Strict adherence to the STP, and
- Timely identification, response, and communication regarding software errors and anomalies discovered by PNNL staff involved in use, maintenance, and development of the eSTOMP software.

This section identifies all functions and activities required to manage the software configuration of eSTOMP and related software products and documentation.

Software configuration management activities are grouped into four general areas:

- software configuration identification
- software configuration control
- software configuration status accounting
- software configuration management assessments and reviews.

3.1 Communications

The SCM is responsible for timely communication of training requirements, risks/hazard exposure, and assignment of commitments/performance of software development and testing staff. If adverse impacts to completed, reported software calculation results are discovered, the SCM will investigate the impact of the error, and if found consequential, report the error and impact to the concerned project manager(s) and product line manager(s). eSTOMP users will be notified via the STOMP/eSTOMP RSS feed set up on SharePoint. Subscription to this feed is required to receive STOMP announcements.

3.2 Software Configuration Control

Software configuration management consists of establishing procedures for

- placing configuration items under software configuration management
- processing SCRs
- creating new versions of software and documentation.

3.2.1 Documenting Releases

The information needed to reconstruct any release (that is, the state of eSTOMP on a given date) will be available through use of the SVN configuration management software. The SVN tools enable the CC to extract source code from the repository by date, which can then be used to build any eSTOMP executable in effect on the specified date. The full specification of a release will include the SVN revision number and the date of the entire source release. The SCR form and JIRA will also document release changes, as applicable.

3.2.2 Placing Items under Software Configuration Management

Documents are placed under configuration control upon publication release or final review of unpublished documents (e.g., letter reports, addendums, errata). Software documentation will be subject to manual version control. Documents are stored and controlled in electronic form on the eSTOMP Confluence site, and paper copies are stored in project files.

Source code modules are placed under the SCM when they become stable and have passed testing. All files related to a source code file or module, such as header and include files, are promoted along with the corresponding source code.

3.2.3 Processing Software Change Requests

During the development process, SCRs are maintained in JIRA. If an external user has made a software request via the STOMP/eSTOMP SharePoint site, then the CM will ensure that the final software change information will be transferred to the SCR so that the user can access the results of the SCR. Code revisions are also tagged with a description of changes when source code is checked into SVN. Final testing and verification of the changes are performed for all software changes.

If the SCR is a request for a bug fix, the SCM is responsible for the SCR approval. If the SCR is a development task, the SCM may evaluate the request independently, or may opt to alert members of the eSTOMP development team that an SCR needs to be evaluated for approval. This decision process is at the SCM's discretion. The SCM or the members of the eSTOMP development team will assess the following:

- Change Evaluation: is the proposed change reasonable and practical?
- Software Design Requirements: is the proposed change consistent with the Software Requirements Document, as outlined in Requirements Document?
- Criticality: is the proposed change critical to completing client deliverables?
- Time Required: how much time will the proposed change require for implementation?
- Configuration Control: will this change create a necessity to constitute a new release?
- Quality Control: How will this change impact the existing code and how much regression testing will be required?

If the change is approved, the SCM will assign it to one or more software development team members for implementation. The software developer(s) will make the software changes proposed in the SCR and perform unit testing as needed on the changes, and notify the CC that the code is ready to be checked in. The CC will then check the updated source code into the SVN repository.

Tests identified in the eSTOMP STP are performed once a new release is updated. The amount of testing required will be evaluated by the eSTOMP developer that implemented the change, which may require that testing be performed on all operational modes in the test suite. An eSTOMP Tester will verify the new source by running the new source through the appropriate test cases. For a new functionality, new test cases will be developed and verified by independent testing. When verification is complete, the eSTOMP developer will check in the revised source code to SVN and will release the source to the eSTOMP user.

When new source is released, the CC may complete the SCR process by notifying users of the final status of the SCR, which may include:

- SCR originator (person who requested one of the changes implemented in the new release)
- All eSTOMP development team members
- All eSTOMP users

Internal software change requests will be tracked using the JIRA software. Software changes will also be tracked in the SVN system by tagging the revised code with text that describes the code changes. JIRA tickets will record the SVN revision number for the updated code, as well as the auto-generated SCR number (if applicable) that is generated on the sharepoint.

3.3 Software Configuration Identification

eSTOMP source code is dynamic research oriented code, as new capabilities are continuously being added to the simulator. Currently, only three modes of eSTOMP exist: -W-(R), -WAE-(B), and -CO2e. Each independent source file, which may contain one or more subroutines for a particular operational mode, will be identified as a unique SCI using the scheme described in the following sections. This identification scheme will enable all project team members, including test engineers, quality advocates, and the SCM to locate each SCI quickly and easily.

3.3.1 Baselines Changes

Each software module will be subject to configuration baselines changes as they are affected by each completed software change request implementation. These baseline changes will be documented for each final code release.

3.3.2 Identifying Configuration Items

Version control and release management for all software source code will be supported using SVN as the configuration management system. Modified source code will be checked into SVN by the CC or other eSTOMP developers as part of the SCR process.

A system of revision numbers will be implemented by SVN, incrementing each revision number by one with each change approval by the SCM. For example, the first source file checked into SVN is assigned a revision number of 1. If a subroutine is revised, then the newly assigned revision number is 2. The new revision number will also be recorded in the SCR form during change implementation.

Because the user is responsible for creating an executable, source code is released directly to users. Complete specification of a version of eSTOMP requires listing the revision number of the source code release.

eSTOMP source code is executed using the PETSc solver (Balay et al. 2006). PETSc is available at <http://www.mcs.anl.gov/>

[petsc](#).

3.3.2.1 Identifying Software Documentation

Documents will be divided into three categories: 1) principal documents, 2) addendums and errata and 3) supporting documentation. Principal documents will be identified by version numbers only. Currently, principal documents are the eSTOMP User's Guide (accessible through stomp.pnnl.gov), the STOMP Theory Guide (White and Oostrom 2000) and the STOMP Application Guide (Nichols et al. 1997). Addendums and errata, published or unpublished, will typically be used to document major revisions, developments or modifications. These documents will be identified with revision and version numbers. Currently, the theory guide is updated with addendums, whereas the user's guide is updated as needed.

Supporting documentation for eSTOMP includes the Software Quality Assurance Plan, Requirements and Design Documents, this SCMP, and the Software Test Plan. This documentation will be maintained in Confluence, and pdf files will be generated and placed on the STOMP/eSTOMP SharePoint for external user access. As new documentation is developed, all older versions of documents will be maintained electronically, as archival material within Confluence. To create a new version, the latest version will be exported as a pdf, and the version number incremented in the title page and the file name. The older version will be moved into an archive subdirectory.

3.3.2.2 Software Source and Documentation Storage and Preservation

Storage and preservation of project software files will be the responsibility of the CC. All source code will be maintained electronically on svn.pnl.gov/estomp. All documentation pertaining to eSTOMP and related software CM, user guides, software requirements, and related documents will be maintained electronically in the eSTOMP space on Confluence. Copies of this documentation will also be maintained on the STOMP SharePoint site.

The contents of these sites are backed up by PNNL's backup services.

3.3.3 Life-Cycle Model

The life-cycle model is established to ensure the proper evolution of SCIs from concept to release of the software. The model for an SCI progressing through the software and management process is as follows:

Requirements --> Design --> Implementation and Unit Testing --> Integration and System Testing --> Release --> Operation

3.3.4 Project File Directory Structure

A standard hierarchical directory structure will be maintained for eSTOMP and related software to facilitate the development and maintenance of electronic documentation and software source code.

All eSTOMP source code is stored on the external PNNL SVN site which is accessible to the eSTOMP developers. The directory structure is aligned with the three operational modes of eSTOMP:

1. eSTOMP-W (-R): svn.pnl.gov/estomp/estomp1
2. eSTOMP-WAE (-B): svn.pnl.gov/estomp/estomp3
3. eSTOMP-CO2e: svn.pnl.gov/estomp/estomp33

eSTOMP utilities used for pre- and post-processing data are stored in:

- svn.pnl.gov/estomp/tools

Software documentation (e.g., manuals, bug fixes, etc.) is accessible through the eSTOMP Confluence space and on the STOMP/eSTOMP SharePoint site.

eSTOMP test cases are also described on the STOMP/eSTOMP SharePoint Site. All input and output files are also accessible at the web site so that users can verify their source code with the test cases. A user may also identify which capabilities have been covered by internal testing.

Other directories may be permitted but the directories identified in this plan are considered core directories that must be maintained.

3.3.5 Storage and Handling

All project SCIs will be identified to ensure that the items are properly stored for traceability, defensibility, and reproducibility. The Code Custodian will ensure that project SCIs are stored and handled appropriately. This includes storage of documents and electronic media, marking and labeling of SCIs, and retention periods. The released software and SVN archive, which can be used to recreate unique versions/revisions of the software will be maintained on the PNNL SVN site. Backups of these files are maintained by PNNL backup services.

3.4 Software Configuration Management Assessments and Reviews

Software CM assessments and reviews will be conducted as needed to determine to what extent the actual software and documentation reflect the required physical and functional characteristics. These activities are formal examinations of the software maintenance and documentation. The SCM will participate in all assessments and reviews and record the resulting action items. As each baseline is defined, a quality engineer may also assist with in-process assessments of software CM activities.

3.5 Software Configuration Status Accounting

Configuration Status Accounting (CSA) is a means by which enhancements/changes and new versions/ revisions of configuration items are identified and tracked. A CSA system is already established with the SVN revision system and the JIRA tracking system. These records provide traceability between versions of SCIs and associated documentation.

4.0 Software Configuration Management Resources

The SCM will ensure that an effective software CM program is established, supported, implemented, and maintained. The SCM or a designated authority is responsible for establishing procedures, plans, and training for the implementation of configuration management.

4.1 Personnel

The SCM is responsible for implementing this plan. The SCM will establish and maintain the development support directories, and process SCM documentation, generating status accounting reports, and preparing and distributing source code. The SCM is also responsible for ensuring that all testing activities are evaluated, documented, and reported according to the requirement of the SCMP. In addition, the SCM will participate in and provide documentation for system development assessments and reviews, if requested. The development team members are responsible for the generation of software, electronic documentation, and other SCIs applicable to the project under the direction of the SCM.

4.2 Software and Hardware

The SCM and development team members will utilize SVN and JIRA to perform identification, control, and status accounting to system SCIs.

- Primary eSTOMP development is carried out on Linux-based platforms.
- eSTOMP testing is carried out on Linux-based platforms using the Intel FORTRAN compiler (version 10.0), on PIC (PNNL Institutional Computing) cluster. The compiler is the FORTRAN intel compiler, version 11.1.069. The output of the `uname -a` command on PIC yields the following information about the operating system:
 - `Linux olympus.local 2.6.32-131.17.1.el6.x86_64 #2 SMP Mon Nov 7 06:23:40 PST 2011 x86_64 x86_64 x86_64 GNU/Linux`
- If a new compiler is implemented, and/or a new operating system is introduced, then the complete set of tests will be run to establish that eSTOMP functionality has not been adversely impacted, and establish a new baseline if needed.

Project records are stored and maintained as specified in the project Records Inventory and Disposition System (RIDS).

5.0 SCMP Maintenance

SCMP maintenance is necessary to document CM activities throughout the software's life cycle. If any procedures defined in this document are changed, those changes will be reflected in the SCMP, as needed.

It is the Software Configuration Manager's (SCM's) responsibility to ensure the compliance and cooperation of development team members in abiding by this plan. The SCM's responsibility is to monitor compliance, and ensure that changes and updates are reflected in the SCMP, as required.

Reviews of this SCMP will occur periodically throughout the eSTOMP simulator's life. At a minimum, reviews will occur at the start of each major development phase. At that time, proposed changes, if any, will be evaluated by the SCM and may be approved by the SCM in conjunction with the CCB for implementation. All changes to this plan will be communicated to the development team in a timely manner.

6.0 Software Configuration Management Schedules

Software CM activities will span the entire life of the software. Software development will consist of the following phases:

- Software Requirements Specification
- Design the Capability
- Assemble the Capability
- Testing and Verification
- Reporting
- Closeout*

The closeout phase of the software will include archiving the source code files and documentation. During retirement, support for eSTOMP and related software is terminated. eSTOMP users and developers will be notified electronically (e.g., email, website) to terminate all eSTOMP and related software usage.

7.0 STOMP Roles and Responsibilities

A list of the STOMP Team and their responsibilities (e.g., developers, testers) is maintained on the STOMP/eSTOMP SharePoint site.