EGG-WM--9103

DE91 001889

CONFIGURATION MANAGEMENT AND SOFTWARE CONTROL PROCEDURE
FOR SUBSURFACE FLOW AND TRANSPORT SIMULATION CODES

Robert G. Baca

Published August 1990

EG&G Idaho, Inc.
Idaho Falls, Idaho 83415

Prepared for the
U.S. Department of Energy
Idaho Operations Office
Under DOE Contract No. DE-AC07-76ID01570

CONFIGURATION MANAGEMENT AND SOFTWARE CONTROL PROCEDURE
FOR SUBSURFACE FLOW AND TRANSPORT SIMULATION CODES

Prepared by: Robert H. Baca Date: 9/20/90
R. G. Baca, Scientific Specialist
Geosciences Unit

Reviewed by: R. C. Arnett Date: 9/20/90
R. C. Arnett, Senior Engineering
Specialist
Geosciences Unit

Reviewed by: M. G. Koll Date: 9/20/90
M. G. Koll, Unit Manager
Technical Support and Integration of
ERP

Reviewed by: J. P. Shea Date: 9/20/90
J. P. Shea, EIRC Chairman
Compliance Assurance Unit of ERP

Reviewed by: V. W. Watson Date: 9/20/90
V. W. Watson, Quality Assurance Officer
Compliance Assurance Unit of ERP

Approved by: Paul Hebdon Date: 9/21/90
J. B. Hebdon, WAG 7 Manager
Environmental Restoration Program

CONTENTS

ACRONYMS AND DEFINITIONS	iv
1. PURPOSE AND SCOPE	1
2. CODE CUSTODIANS AND DEVELOPERS	3
3. METHODS FOR CONFIGURATION MANAGEMENT AND QUALITY	4
REFERENCES	6

TABLE

1. Primary computer codes used in modeling studies for the RWMC	2
---	---

ACRONYMS AND DEFINITIONS

INEL -- Idaho National Engineering Laboratory
RCRA -- Resource Conservation and Recovery Act
RWMC -- Radioactive Waste Management Complex
VCS -- Version Control System

Baseline. A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedure.

Code benchmarking. A process of testing the simulation capability of a computer code by making comparisons against other independent computer codes.

Code verification. A process of testing the accuracy and correctness of a computer code by making comparisons against analytical solutions.

Computer code. A computer program written in FORTRAN language that performs a numerical solution of a set of governing equations and produces calculations of state variables (i.e., temperature, hydraulic head, concentration).

Confirmation testing. Code testing performed to confirm if the modified computer code is fully operational. Testing is performed by executing a standard set of test cases and comparing the output to previous results or a known reference solution. Acceptance criteria is agreement of results to four digits.

Development code. A computer code that is operational in a preliminary manner but has not been verified and benchmarked.

Master library. A software library containing controlled versions of the computer simulation codes.

Production code. A computer code that is fully operational as demonstrated by a code verification and benchmarking.

Quality. The totality of features and characteristics of a computer code that depends on its ability to meet preset technical requirements.

Quality assurance. Planned and systematic actions performed to provide adequate confidence that the computer code conforms to technical requirements.

Doc. No.: EGG-WM-9103
Section No.: _____
Revision No.: 0
Date: August 1990
Page No.: v of 6

Software. A set of programs and/or subprograms written in FORTRAN that makes up a computer code.

Software reliability. The ability of a computer code or program to perform its function under applicable assumptions and conditions.

Software verification. The process of testing a modified computer code to ensure that the code is working as required. This testing is performed as part of confirmation testing.

Source code. The ASCII FORTRAN program and/or subprograms that make up a computer code.

CONFIGURATION MANAGEMENT AND SOFTWARE CONTROL PROCEDURE
FOR SUBSURFACE FLOW AND TRANSPORT SIMULATION CODES

1. PURPOSE AND SCOPE

Computer modeling studies will be performed to assess environmental concerns associated with the mixed-waste disposal site at the Radioactive Waste Management Complex (RWMC). For these studies, the Geosciences Unit has assembled and developed a set of simulation codes for flow and transport in the subsurface environment. Some of the simulation codes are currently being used to predict the downward migration of organic and radioactive contaminants. These model predictions are being used to anticipate potential problems and as a basis for selecting remedial actions. Because these predictions support the Resource Conservation and Recovery Act (RCRA) activities for the disposal site, there is a need to ensure software integrity, quality, and reliability.

Computer simulation codes used to support RCRA activities fall into three basic categories: (a) codes that simulate natural environmental processes in the subsurface (e.g., heat transfer, air flow, moisture transport), (b) organic contaminant transport (e.g., volatile organic chemicals, oils), and (c) radionuclide transport (e.g., plutonium, americium). Some of the computer codes have multiple capabilities and thus fall into more than one category. Although a large number of computer codes and software¹ are being used to model these different processes, this procedure is limited to the principal simulation codes.

The purpose of the procedure is to establish the methods for configuration management and responsibilities for software control. As used here, configuration management refers to the systematic process of controlling/documenting all modifications and updates to the simulation codes. Software control refers to authority and responsibility for the maintenance, development, and controlled use of the computer codes. The combination of configuration management and software control ensures that:

- All code modifications and updates are traceable and auditable
- Production versions represent verified and benchmarked software
- Codes are documented internally with regard to their primary application.

This particular procedure applies primarily to the production versions of the subsurface flow and transport simulation codes which are installed on the CRAY supercomputer.

This procedure outlined in the subsequent sections is consistent with general computer software configuration management procedure QP-21² and specifically applies to the simulation codes listed in Table 1. Production versions of these codes are installed in a master directory on the CRAY supercomputer.^{3,4}

TABLE 1. PRIMARY COMPUTER CODES USED IN MODELING STUDIES FOR THE RWMC

<u>Computer Code</u>	<u>Description</u>	<u>Responsible Code Developer</u>
UNSAT-H	Heat and Moisture Transport	S. O. Magnuson
FLASH	Heat and Moisture Transport	R. G. Baca
MAGNUM-2D	Air or Water Flow, Heat Transport	R. G. Baca
CHAIINT	Gas or Solute Transport	R. G. Baca
MOFAT	Multiphase Immiscible Flow	R. G. Baca
PORFLO-2D	Air Flow, Gas or Solute Transport	J. C. Walton
PORFLO-3D	Air Flow, Gas or Solute Transport	A. J. Sondrup
TRACR3D	Moisture and Solute Transport	J. C. Walton
MAGNUM-3D	Air or Water Flow, Gas Diffusion	R. C. Arnett

The production versions of the codes listed in the table have been verified and benchmarked and, therefore, are approved for use in modeling studies for RCRA activities.

2. CODE CUSTODIANS AND DEVELOPERS

The responsibility for code control is divided between a master custodian and the code developers to provide a system of checks and balances. The master custodian shall have the sole responsibility for modifying/updating the production versions of the simulation codes. At present, the designated master custodian is N. H. Marshall of the Scientific Applications Unit. Other responsibilities of the master custodian are:

- Implement the configuration management procedure (see Section 3)
- Maintain documentation on code modifications
- Perform confirmation testing of modified/updated versions
- Limit production code access to code developers and designated users only.

Code developers shall be responsible for testing proposed code changes prior to requesting a formal change or update of the controlled versions. Correctness of these updates will be confirmed by the master custodian as part of the configuration management procedure. Other general responsibilities of the code developers are:

- Document the source code with regard to: (a) responsible individual, (b) status and revision number, (c) general description of code application, and (d) input instructions
- Prepare a set of standard test problems for code verification and benchmarking
- Initiate requests for modification/update of production versions.

The designated code custodians for development versions of the simulation codes are listed in Table 1.

3. METHODS FOR CONFIGURATION MANAGEMENT AND QUALITY

Most production versions of the simulation codes are installed on the Idaho National Engineering Laboratory (INEL) CRAY X-MP Model 2/16 supercomputer. Development versions are implemented on IBM PS/2 (Models 60, 70, and 80) and a Sun Workstation (Model 4/1100). The master custodian of production codes will use the following tools and approaches for configuration management:

- Maintain an electronic logbook of code modifications
- Use the Version Control System (VCS) software to track and document changes to the baseline source code
- Assign code revision or release numbers
- Use the CRAFT software to check software quality and integrity
- Perform confirmation testing for software verification and reliability.

Existing production versions of the codes will be modified/updated as a result of a request from a code developer or user. A request for code modification is made to the master custodian that: (a) identifies the code to be changed, (b) describes the new functions or requirements of the code, (c) identifies the specific line changes to be made to the source code, and (d) specifies the test cases to be used for confirmation testing.

The master custodian shall use the VCS software⁵ to implement the requested changes. This software is part of the CRAY system software and is designed to track all changes to a computer code without actually altering the baseline code. Periodic updating of the actual baseline code will be performed only after (a) completion of code verification, benchmarking, and software validation and (b) a sufficient period of successful code operation. After such an update, the simulation code will be given a new revision or release number.

For the purpose of software verification, the master custodian shall use the CRAFT software to perform a subprogram and common block analysis. The CRAFT software produces: (a) a cross-reference table of common blocks and subprograms in which they appear, (b) a table for each common block which shows the characteristics of individual parameters in the common blocks, (c) a cross reference table of subprograms indicating which routines each subprogram calls (as well as which routines call it), and (d) a hierarchical call tree of subprogram. Summary information for various potential coding problems is output such as:

- Variable names in common blocks that are used but not defined
- Inconsistent usage of parameter in common blocks
- Subtle typographical errors in variable names.

These types of coding errors occur frequently during computer code modifications and are difficult to identify by manual methods.

Prior to finalizing the modifications/updates, the master custodian shall perform confirmation testing on the new revised code. The standard test cases identified by the code developer shall be run on the CRAY and the results checked for correctness. The standard test cases shall be taken from the verification and/or benchmark test cases used to check prior versions of the code. In the event code results do not confirm previous results, the master custodian will notify the code developer for further action.

REFERENCES

1. P. T. Laney et al., Annual Progress Report: FY-1987, Subsurface Investigations Program at the Radioactive Waste Management Complex of the Idaho National Engineering Laboratory, DOE/ID-10183, EG&G Idaho, Inc., Idaho Falls, Idaho, 1988.
2. EG&G Idaho, Inc., Quality Manual, QP-21, "Computer Software Configuration Management," Idaho Falls, Idaho, 1988.
3. CRAY Research, Inc., UNICOS Commands Reference Manual, SR-2100, Mendota Heights, Minnesota, 1987.
4. CRAY Research, Inc., UPDATE Reference Manual, SR-0013, Mendota Heights, Minnesota, 1987.
5. D. G. Barber, VCS: A Networked Version Control System, EGG-CATT-8926, EG&G Idaho, Inc., Idaho Falls, Idaho, February 1990.

END

DATE FILMED

12 / 03 / 90