Office of Energy Research (ER) Financial Management Information System (FMIS)

Software Configuration Management Plan

February 1996

DRAFT

U. S. DEPARTMENT OF ENERGY

Office of Energy Research

Title Page

Document Name:	ER Financial Management Information System (FMIS) Software Configuration Management Plan (Draft)
Publication Date:	February 1996
Contract Number:	DE-AC01-95AD34277
Project Number:	DOE: 25-ER6-02-7-04 DynCorp: 3009
Prepared by:	Bernhard Knutsen, DynCorp John Olivier, DynCorp
Approval:	_____ Ralph DeLorenzo, ER-63 Financial Management Division
Concurrence:	_____ Pat Rice, ER-621 Information Resources Management Branch

THIS IS A WORKING DOCUMENT THAT HAS NOT RECEIVED DOE MANAGEMENT SANCTION FOR PUBLICATION, IS SUBJECT TO REVISION, AND SHOULD NOT BE CIRCULATED OUTSIDE OF THE RECEIVING OFFICE.

U. S. DEPARTMENT OF ENERGY

Office of Energy Research

Contents

1. Introduction

The Financial Management Information System (FMIS) will provide budget formulation and budget execution support functions to the Office of Energy Research Financial Management Division (ER-63). FMIS will replace the functions of the Business Management Information System (BMIS) currently used by ER-63. FMIS will be developed in multiple phases, including the following.

Replace the budget execution functionality provided by BMIS and develop the base database structure to support future phases.

Replace budget formulation functionality provided by BMIS.

Implement an Integrated Procurement System (IPS) interface for Procurement Request (PR) concurrence and data feed for Approved Financial Plan (AFP) updates.

Provide Program Office interfaces for collection of AFP change input, and for the collection of formulation input.

1.1 Purpose

This Software Configuration Management Plan (SCMP) identifies the approach for controlling the development and release of FMIS in phases. For each phase, a baseline will be established and procedures followed to manage the evolution of the application software and the documentation.

Within each of the major releases of FMIS, there may be one or several incremental releases of the system as functionality is developed and problems are fixed. A significant goal of this plan is to have clearly-defined updates rather than an ongoing stream of minor changes. Another goal is to be able to provide updates to a release that is in testing or production while the next baseline is being developed. In addition to having an index of the changes made from version to version, separate copies of the deliverables before and after will be saved to permit comparisons at a detail level. This is useful in the event that problems arise following a new release because
(1) we can compare the before and after versions to isolate the source of the problem, and
(2) we can be in a position to revert to an earlier version if necessary.

The following chapters cover the four mechanics of software configuration management: **Identification** - how Software Configuration Items are identified; **Control** - how change control will be accomplished; **Auditing** - how the software configuration will be audited; **Status Accounting** - how the status of the configuration effort will be reported. This document focuses on the software configuration management procedures as they apply in

the development environment. Once the whole system is in production, the SCMP will be reviewed and revised for application in the production environment.

1.2 Background

The primary language used to program the FMIS application will be ***Microsoft Visual Basic***. The front-end user interface of the FMIS application will be developed to run under the ***Microsoft Windows*** environment. FMIS itself will run from the ***ER (Novell) Local Area Network***. The programs and associated files will be stored on the LAN to provide access to ER-63 staff from a common source. This will simplify system installation, support, and ensure that any software updates are applicable to all users at the same time.

The FMIS database will be stored on a separate server running ***Microsoft SQL Server for Windows/NT***. The most significant advantages of using a SQL-based database server are the ability to build tight access controls directly into the data source and the minimizing of network communications traffic.

1.3 Special Organizational Features

Figure 1 lists the organizations that will be involved with the software configuration management of FMIS. See appendix A for a summary of roles and responsibilities.

Organization	Contact	Phone Number
ER-63	Ralph DeLorenzo Isla Linger	903-3541 903-3540
ER-621	Dick Yockman Pat Rice	903-4478 903-4556
DynCorp	Paula Thompson Bob Kuehne	903-0850 903-9773
DynCorp QA	Lew Bachman	903-0880
System Development Team	John Olivier	903-6587

Figure 1

2. Identification

Software Configuration Items are the software and documentation of an application system that are subject to continuous change and must be maintained concurrently.

2.1 Responsibility

The responsibility and authority for identifying Software Configuration Items will be performed within the DynCorp System Development Team. The system analysts and the developers will identify the Software Configuration Items during the design and development stage of each phase. The Configuration Manager will review and make sure that items are not omitted. The Software Librarian will maintain a record for each item in the Software Configuration Management (SCM) database. These functions will be performed as part of the existing System Development Team's ongoing functions.

2.2 Identify Software Configuration Items

The following subsections detail the Software Configuration Items that have been identified by the System Development Team. Items not included will be identified during the design and development of each phase.

2.2.1 Documentation Identification

During the development of the system, principal documents will be maintained and updated with each software release. Each document will contain the Document Name, Publication Date, Contract Number, Project Number, Author(s), and Approval/Concurrence Signatures. Documents will be issued in draft and final forms. The Project Manager will be responsible for the release of the documents. The Technical Writer will control and maintain the electronic versions.

As changes are made to the system, any or all of these documents will require modification. For draft documents, the changes would be incorporated in the current version and would be noted in the normal communication regarding the affected documents. For final documents, change pages would be issued. Final documents with multiple change pages may be reissued when a new baseline is released or when the whole system is implemented. Change pages will be enumerated in a change page log in each document.

The following principal documents will be issued in each phase.

Requirements and Design Documents will be issued as final and signed off in each phase. Each document will cover the features and functionality planned for that phase. The data dictionary, menus, screen layouts, edits, and report layouts will be maintained throughout the development of the system. Change pages will be issued for documents already issued.

User System Reference Manual will be reissued with each phase. New features and functionality developed for that phase will be added. Changes to portions already reviewed will be incorporated in the current version and will be noted in a cover letter.

Programmer System Maintenance Directory will be reissued with each phase. Enhancements and new modules developed for that phase will be added. Changes to portions already reviewed will be incorporated in the current version and would be noted in a cover letter.

Training Materials will cover the features and functionality developed for that phase. Since the Training Materials are only for that phase, they are not planned to be maintained once the phase is completed.

The following documents are considered to be baseline documents and will be reviewed as necessary by the Development Team for effects of software changes.

Project Phasing Study, Revised Draft, October 1994

Phase 1 Project Plan, October 1995

Data Conversion Plan, October 1995

Phase 1 Conceptual Design, October 1995

Data Sharing Approach, November 1995

2.2.2 Software Identification

The FMIS software consists of a database and the application software, which need to be controlled through configuration management procedures. The database items include data files, their structure as defined in a data dictionary, and SQL stored procedures and triggers. The application software items include screen forms, report forms, modules, and custom controls (VBX's) as listed in a Visual Basic project file. The forms and modules are combined in an executable file for distribution. In addition, there may be other system files necessary for the application to work, i.e., text/data files, help files, initialization files, and dynamic-link libraries. All of these items are DOS files and will be identified and tracked by the DOS file name, date/time last modified, and byte size.

The software components within these physical files include programs for menus, data access, data input, queries/extracts, standard reports, and batch processes. These programs will be labeled for identification and cross-referenced to the actual DOS file. In addition, all development, testing, and conversion utilities will be identified and tracked.

Each Software Configuration Item will have its own version number for identification and tracking purposes. A version control program, such as Polytron Version Control System (PVCS) or Visual SourceSafe, will be used for the following.

Check files in and out
Maintain a record of all file activity
Control access to the files
Label releases
Reconstruct prior releases
Provide development branching

Depending upon the capabilities of the selected version control program, a ***Microsoft Access*** database could be created to provide an inventory of all configuration items.

2.2.3 Software Configuration Stages

FMIS programs, system files, and documentation will be stored on the ER LAN, while the FMIS database will be stored on a SQL Server. At least the source code and forms for the programs will be maintained through a version control program. The system files, documentation, and data structures definitions will be controlled through a version control program. The version control program will keep the complete text of the latest revision of a file and all its change deltas. This will allow any prior revision of the file to be reconstructed. Because of this, formal development, test, and production libraries are not needed.

Each phase of the FMIS development project will have four stages: Development, User/Integration Testing, Acceptance/Operational Testing, and Production Operations. For each release, a plan would be developed for stage transition. This plan would include procedures to load or reload the database for the release in each stage. The stage exit review will determine if the release is ready for the next stage.

Development Items are developed or updated including: modules, database, and documentation. When a version is ready for release, a snap shot of all the items within the revision that makes up the release will be stored on the ER LAN. A version of the database as defined by the release will be stored on the SQL Server. This release is now ready for testing. Development continues on the next release.

The System Development Team working with ER-63 Budget Analysts will test the release. Software and documentation changes may be identified. Software changes would be incorporated in a subsequent release, while documentation changes without software changes may be updated here.

2.2.4 Software Release Identification

A software release consists of all configuration items including documentation, identified by a release label. The system will be incrementally released as major portions of system code are developed for testing by the users. Major releases are primary baseline releases, while minor releases will reflect updates to the current running baseline. The release label will be constructed as follows.

System Acronym: FMIS

Release Version: Starting with 1.00 and incrementing decimally for minor releases, whole numbers for major releases. For example, the first release will be **FMIS 1.00**.

The System Developers will issue multiple releases for testing by the user before making revisions and issuing a final release for operation. The Project Manager and Configuration Manager will be responsible for identifying and planning each release. It is recommended that each phase of the planned system development be a major release. Within each major release, functionality may be staged through minor releases. In addition, corrections to system problems and updates of new or revised requirements under the current major release will be incorporated into minor releases. Each release will include all functionality from prior releases unless changed.

A release is made when the development and unit testing is completed for all items planned for the release. It will include any new functionality targeted for the release plus all configuration items of the prior release and any updates to prior releases. A release will go through four stages: Development, User/Integration Testing, Acceptance/Operational Testing, and Production Operations. Once a release is made, no modifications will be made to that release. If software changes are needed, the release will not go to the next stage. The software changes will be included in subsequent releases. Figure 2 shows a possible scenario for released versions.

Stage 1=Development; Stage 2=User/Integration Testing;
Stage 3=Acceptance/Operational Testing; Stage 4=Production Operations.

Figure 2

Version control programs allow for separate development paths using branches to create parallel versions of the system. For example, the release in production needs to be fixed while development continues on the next release. There also may be an intermediate release in testing. When the production release is fixed, it is merged in the other releases. This example is illustrated by releases 1.4, 2.2, and 3.1 in Figure 2.

3. Change Control

In a multi-phased development project, the introduction of new modules and functionality will be controlled through the change control procedures. Change control requires the systematic evaluation, coordination, and approval or disapproval of proposed changes to the defined baseline. Any addition, alteration, or deletion to the approved baseline is deemed a change and is subject to change control. Changes that are completed and accepted by the user become the approved baseline for that phase of the project.

3.1 Responsibility

The following roles and responsibilities will ensure orderly change.

Project Manager interfaces with ER-63, ER-621, and the System Development Team to ensure that change control procedures are followed and that the developed application meets the users' expectations. The FMIS Project Manager is John Olivier.

Configuration Manager will determine the areas that are affected by each Software Configuration Item. The areas that may be affected include module code changes, documentation, tutorial, test changes, databases, file indexes, conversion, and setup or manual procedures. The FMIS Configuration Manager is to be assigned as part of the Development Team's duties.

Software Librarian ensures version control of all development, test, and operational release versions. The FMIS Software Librarian is to be assigned as part of the Development Team's duties.

Technical Writer ensures that system documentation is controlled in accordance with Software Configuration Items identified by the Configuration Manager. The FMIS Technical Writer is Tom Midlam.

ER-63 and ER-621 Point(s) of Contact are responsible for identifying, approving, and prioritizing software changes with the Project Manager. The ER-621 Point of Contact is Pat Rice, and the ER-63 Point of Contact is Ralph DeLorenzo.

Budget Analysts from ER-63 identify and test items for change. Specific individuals have been identified with the program data for which they are responsible; we expect that all will participate in the testing process.

During the development phase, a formal Change Control Board does not need to be organized. An informal group consisting of the Project Manager, Configuration Manager, ER-63 and ER-621 Points of Contact, and one or more Budget Analysts would meet as necessary to review user software change requests. The Project Manager and Configuration Manager would review internal software change requests from the System Development Team. To ensure that the configuration management of FMIS runs smoothly, all of the above individuals should agree on the following.

Change request procedures.

Criteria for prioritizing, evaluating, and approving a change request.

Procedures for getting a change into production.

Emergency fix procedures.

Adequate testing procedures.

The steps in these procedures are documented in appendix B. The level of involvement by the individuals cited in this section will be negotiated prior to implementation of the procedures.

3.2 Software Configuration Control Documents

Control documents are needed to track and manage changes to software and documentation. These forms will be maintained by the Software Librarian. The following forms will be used.

Software Change Request (SCR) - used to report problems, identify new or changed requirements, and suggestions for improvement. A design of this form is in appendix C.

Software Modification Record (SMR) - used to track new modules developed, modifications made to existing items, and software releases. A design of this form is in appendix D.

4. Auditing

Auditing verifies that configuration items reflect operational objectives, meet DOE standards, and satisfy user requirements.

4.1 Responsibility

The System Development Team, working with ER-63 and ER-621, will follow a technical review process. This will include Structured Walkthroughs, In Stage Assessments, Stage Exits, and System Acceptance.

4.2 Baseline Review Procedures

The following reviews will be performed during each phase of development.

Structured Walkthroughs are informal meetings among DynCorp's Development Team to review and evaluate technical aspects of the specification, design, and code as developed or changed.

In Stage Assessments will be conducted with a DynCorp Quality Assurance (QA) consultant near the end of each project stage within each development phase. This is a technical review to assure that the established system development and project management process is being followed effectively.

Stage Exits are formal meetings with a group of selected individuals to review and evaluate the current status of the project. When a stage has been successfully "exited", it indicates that all deliverables due to date have been completed, all outstanding issues have an acceptable action plan, and there is a sound plan for the remainder of the project. All approvers must provide a written position of concur/non-concur at the Stage Exit.

System Acceptance is the final Stage Exit for each development phase that is implemented for production. It provides a formal basis for determining whether the system is fully operational and has satisfied user requirements.

5. Status Accounting

Status accounting provides recording and tracking of all significant software and documentation development, problems, changes, or actions.

5.1 Responsibility

The System Development Team will be responsible for recording and tracking all configuration items, events, and releases through the Configuration Manager.

5.2 Recording

A ***Microsoft Access*** database will be established for the purpose of software configuration management tracking. This database will be called the FMIS SCM database and will contain information about reported problems, requested changes, and modifications made as they apply to identified configuration items. The software configuration control documents and attachments would provide details and an audit trail.

The following information will be recorded from the Software Change Request Form, depicted in appendix C.

Requirement #
SCR Control Number (assigned by the system)
Originator
Date Submitted
Current Release Number
Type of Request (New Requirement, Requirement Change, Design Change, System Problem, User Interface Problem, Documentation Correction, Suggestion for Improvement, etc.)
Priority (High, Normal, Low)
Title
Description
Status (Open, Reviewed/Estimated, On Hold, Canceled, Approved, Coded, Documented, Completed)
Status Date
Comments (memo field)
SMR Control Number
New Release Number

The following information will be recorded from the Software Modification Record Form, depicted in appendix D.

SMR Control Number (assigned by the system)
Developer
Release Number
Module
Class
Title
Description
List of configuration items that applies.
Status (Coded, Released, Tested, Documented, Approved)
Status Date
Comments (memo field)

A commercial database package, such as ***Microsoft Access***, would easily support the needs of software configuration management tracking by providing an environment for recording, reporting, and flexibility for change.

5.3 Reporting

The System Development Team will be able to use ***Microsoft Access*** to produce standard or ad hoc reports from the FMIS SCM database in order to provide an accurate picture of the status of problem reports and change requests. Standard reports may include the following.

Change Request Report - Summary
Modification Record Report - Summary
Open Change Requests - Detail
Closed Change Requests - Past Month
Software Release Report - Detail

Appendix A
Software Configuration Management Roles and Responsibilities
(Please note that roles and responsibilities may be split or shared between people.)

Project Manager

Interfaces with ER-63 and ER-621 Point(s) of Contact to ensure that SCM plan procedures are followed and that the developed application is within the users' expectations.
Consults with ER-63 and ER-621 Point(s) of Contact for approval and prioritization of software and documentation changes.
Makes Project Staff assignments for preliminary investigation of change request (analysis, feasibility and impact study, resource estimates); making the change; testing the change.
Reviews analysis, feasibility and impact study, and resource estimates.
Monitors schedules and status items to ensure timely delivery.
Reviews system release plans with the Configuration Manager.
Coordinates system releases with the Configuration Manager and ER-63 Point(s) of Contact.
Responsible for the release of documentation.

Configuration Manager

Coordinate with Project Staff in identifying Software Configuration Items.
Determines the areas affected by each Software Configuration Item.
Classifies each Software Configuration Item and assigns it to a Module.
Maintains an inventory of all Software Configuration Items.
Responsible for collecting data and monitoring the Software Configuration Management database.
Responsible for providing status reports on the configuration effort.
Maintains the SCMP; reviews and updates for each new phase.
Ascertains Software Configuration Items, and ensures traceability throughout the information system life cycle.
Ensures that the configuration management process is executed according to the plan.
Interfaces with the Project Manager and the Project Staff.
Plans system release with the Project Manager.
Coordinates system releases with the Software Librarian.
Coordinates documentation changes.

Software Librarian

Maintains copies of all configuration control documents.
Maintains the SCM database.
Ascertains that no unauthorized changes have been made and no files have been added, deleted, or changed erroneously.
Makes system releases as instructed by the Configuration Manager.
Ensures version control and validates all releases.
Maintains distribution lists for changed documents/release updates.

Project Staff (System Analysts and Developers)

Identifies new or previously unidentified Software Configuration Items during design, development, testing, and processing changes.
Initiates Change Request by encountering system problems or recognizing performance improvements during design, development, testing, and processing changes.
Performs preliminary investigation of change request (analysis, feasibility and impact study, resource estimates).
Performs the requested change.
Tests the requested change.

Technical Writer

Ensures changes are clearly presented.
Reviews and updates documents for conformance to established standards and is acceptable for publication.
Controls and maintains issued documents.

ER-621 Point of Contact

Advises ER-63 Point of Contact and the System Development Team (through the Project Manager).
Signs off on major software and documentation deliverables.
Concurs/Non-concurs at stage exits with the successful completion of each phase.
Concurs/Non-concurs upon any substantive system development issues that impact resources, schedules, and other operational considerations.
Advises ER-63 Point(s) of Contact and Project Manager on approving and prioritizing software and documentation changes.
Initiates Change Request when reviewing project by identifying new or previously unidentified requirements.
Provides budgetary resources for the project.

ER-63 Point(s) of Contact

Approves/disapproves system functionality proposed.
Signs off on major software and documentation deliverables.
Concurs/Non-concurs at stage exits with the successful completion of each phase.
Concurs/Non-concurs upon any substantive system development issues which impact resources, schedules, and other operational considerations.
Initiates Change Request when reviewing project by perceiving new or previously unidentified requirements.
Approves and prioritizes software and documentation changes with ER-621 Point of Contact and Project Manager.

Budget Analysts

Test release for integrity, conformance to budget office procedures, and that computations are executed correctly.
Initiate Change Request by encountering system problems, recognizing performance improvements, or perceiving new or previously unidentified requirements during review and testing.

Appendix B
Normal Change Control Steps

CM = Configuration Manager; PM=Project Manager; PS=Project Staff;
SL=Software Librarian; All=Project Staff, Users, Trainers etc.

Appendix C

Software Change Request (SCR) Form

Appendix D
Software Modification Record (SMR) Form