ENGINEERING CHANGE NOTICE

Page 1 of 2

1. ECN ~~105495~~

Proj. ECN 400327

2. ECN Category (mark one) Supplemental ☐ Direct Revision ☒ Change ECN ☐ Temporary ☐ Standby ☐ Supersedure ☐ Cancel/Void ☐		3. Originator's Name, Organization, MSIN, and Telephone No. T. A. Gates, BCSR/LANSS, B5-24, 376-8967		4. Date 9/23/94	
		5. Project Title/No./Work Order No. Activity Management System/E12089		6. Bldg./Sys./Fac. No. N/A	
		8. Document Numbers Changed by this ECN (includes sheet no. and rev.) WHC-SD-HWV-SDP-004, REV 0		9. Related ECN No(s). N/A	
				7. Approval Designator N/A	
				10. Related PO No. N/A	
11a. Modification Work ☐ Yes (fill out Blk. 11b) ☒ No (NA Blks. 11b, 11c, 11d)		11b. Work Package No. N/A		11c. Modification Work Complete N/A	
		Cog. Engineer Signature & Date		11d. Restored to Original Condition (Temp. or Standby ECN only) N/A	
		Cog. Engineer Signature & Date			
12. Description of Change Small Job Development Software Project Documentation for the Activity Management System, WHC-SD-HWV-SDP-004, Revision 1 changes. Incorporates changes made in AMS functionality.					
13a. Justification (mark one)					
Criteria Change ☒		Design Improvement ☐		Environmental ☐	
As-Found ☐		Facilitate Const. ☐		Const. Error/Omission ☐	
13b. Justification Details AMS users desired that additional functions be added.					
14. Distribution (include name, MSIN, and no. of copies) See attachment				RELEASE STAMP OFFICIAL RELEASE 41 BY WHC DATE SEP 22 1994 <i>Station #12</i>	

A-7900-013-2 (06/94) GEF095

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

A-7900-013-1 (11/88)

400327

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

RELEASE AUTHORIZATION

Document Number: WHC-SD-HWV-SDP-004, REV. 1

Document Title: Small Job Development Software Project Documentation
For The Activity Management System

Release Date: 9/22/94

* * * * *

**This document was reviewed following the
procedures described in WHC-CM-3-4 and is:**

APPROVED FOR PUBLIC RELEASE

* * * * *

WHC Information Release Administration Specialist:

N.L. SOLIS
(Signature)

(Date)

SUPPORTING DOCUMENT

1. Total Pages **146**

<p>2. Title</p> <p>Small Job Development Software Project Documentation for the Activity Management System</p>	<p>3. Number</p> <p>WHC-SD-HWV-SDP-004</p>	<p>4. Rev No.</p> <p>1</p>
<p>5. Key Words</p> <p>Computer software, activity management, commitment tracking, small job development, software management, configuration management plan, test plan, requirements documentation, design documentation</p>	<p>6. Author</p> <p>Name: T. A. Gates/ M. B. Burdick</p> <p><i>T. A. Gates / M. B. Burdick</i> Signature</p> <p>Organization/Charge Code 62340/E12089 D1 B3C</p>	
<p>APPROVED FOR PUBLIC RELEASE 9/22/94 U. S. Dept. of Energy</p>		
<p>7. Abstract</p> <p>This document contains all the required software development documentation as required by WHC-CM-3-10, <i>Software Practices</i></p>		
<p>8. PURPOSE AND USE OF DOCUMENT - This document was prepared for use within the U.S. Department of Energy and its contractors. It is to be used only to perform, direct, or integrate work under U.S. Department of Energy contracts. This document is not approved for public release until reviewed.</p> <p>PATENT STATUS - This document copy, since it is transmitted in advance of patent clearance, is made available in confidence solely for use in performance of work under contracts with the U.S. Department of Energy. This document is not to be published nor its contents otherwise disseminated or used for purposes other than specified above before patent approval for such release or use has been secured upon request, from the Patent Counsel, U.S. Department of Energy Field Office, Richland, WA.</p> <p>DISCLAIMER - This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.</p>	<p>10. RELEASE STAMP</p> <div style="border: 2px solid black; padding: 10px; text-align: center;"> <p>OFFICIAL RELEASE 41</p> <p>BY WHC</p> <p>DATE SEP 22 1994</p> <p><i>Station # 12</i></p> </div>	
<p>9. Approval Designator N/A</p>		

MASTER

[illegible]

CONTENTS

1.0	INTRODUCTION	1-1
1.1	OVERVIEW	1-1
1.2	PURPOSE	1-2
1.3	SCOPE	1-3
2.0	SYSTEM PROJECT MANAGEMENT PLANNING	2-1
2.1	PROJECT DELIVERABLES	2-1
2.2	PROCESS MODEL	2-1
2.3	REVIEWS	2-2
2.4	ORGANIZATIONAL STRUCTURE	2-2
	2.4.1 Customer Organization	2-2
	2.4.2 Performing Organization	2-3
	2.4.3 Other Supporting Organizations	2-3
2.5	PROJECT RESPONSIBILITIES	2-3
	2.5.1 Customer Organization	2-4
	2.5.2 Performing Organization	2-4
	2.5.3 Other Supporting Organizations	2-4
2.6	PROCEDURES USED	2-5
2.7	APPROVAL DESIGNATOR	2-5
2.8	CHANGE AUTHORITY	2-5
2.9	STATUS REPORTING	2-5
2.10	ASSUMPTIONS, DEPENDENCIES, AND CONSTRAINTS	2-5
	2.10.1 Assumptions	2-5
	2.10.2 Dependencies	2-6
	2.10.3 Constraints	2-6
2.11	WORK TASKS	2-6
	2.11.1 Prepare Concepts Definition	2-6
	2.11.2 Define Preliminary Requirements	2-6
	2.11.3 Write Functional Requirements Document	2-6
	2.11.4 Customer Review and Approval of Functional Requirements	2-7
	2.11.5 Code First Prototype Version	2-7
	2.11.6 In-Process Demonstration to Customer	2-7
	2.11.7 Install First Prototype for Customer Evaluation	2-7
	2.11.8 Customer Evaluation--Prototype Update	2-7
	2.11.9 Prepare Small Job Documentation	2-7
	2.11.10 Prepare User Manual	2-7
	2.11.11 Prepare Test Procedures	2-7
	2.11.12 Perform Customer Acceptance Test	2-8
	2.11.13 Perform Final V&V Review	2-8
	2.11.14 Receive Final Project Approvals	2-8
2.12	PROJECT SCHEDULE	2-8
3.0	ACCEPTANCE CRITERIA	3-1
4.0	REQUIREMENTS SPECIFICATION	4-1
4.1	INTRODUCTION	4-1
	4.1.1 Purpose	4-1
	4.1.2 Scope	4-1
	4.1.3 Definitions	4-1
	4.1.4 Overview	4-2

CONTENTS (cont)

4.2	GENERAL DESCRIPTION	4-2
4.2.1	Product Perspective	4-2
4.2.2	Product Functions	4-3
4.2.3	User Characteristics	4-4
4.2.4	General Constraints	4-4
4.2.5	Assumptions and Dependencies	4-5
4.3	SPECIFIC REQUIREMENTS	4-5
4.3.1	Functional Requirements	4-5
4.3.2	Performance Requirements	4-13
4.3.3	External Interface Requirements	4-14
4.3.4	Design Constraints	4-16
4.3.5	Attributes	4-16
4.3.6	Other Requirements	4-17
4.4	AMS MENUS	4-17
4.4.1	AMS Main Screen Menu Selections	4-18
4.4.2	AMS Data Entry Form Menu Selections	4-19
4.4.3	AMS MemoEdit Field Menu Selections	4-19
4.4.4	AMS Validation Screen Menu Selections	4-19
4.4.5	AMS Retrieved Activities Screen Menu Selections	4-20
5.0	ALTERNATIVES ANALYSIS	5-1
5.1	ALTERNATIVES CONSIDERED	5-1
5.1.1	Criteria	5-1
5.2	ALTERNATIVES	5-1
5.2.1	Buy Commercial System	5-1
5.2.2	Use Existing System	5-2
5.2.3	Enhance Existing System	5-2
5.2.4	Create New System	5-2
5.2.5	No-Action Alternative	5-3
6.0	SDD	6-1
6.1	INTRODUCTION	6-1
6.1.1	Purpose	6-1
6.1.2	Scope	6-1
6.1.3	SDD Structure	6-1
6.2	DECOMPOSITION DESCRIPTION	6-2
6.2.1	Module Decomposition	6-2
6.2.2	Concurrent Process Decomposition	6-5
6.2.3	Data Entity Decomposition	6-5
6.3	DEPENDENCY DESCRIPTION	6-5
6.3.1	DFDs	6-5
6.3.2	Data Dependencies	6-12
6.3.3	Entity Descriptions	6-12
6.3.4	Conversion Dependencies	6-19
6.4	INTERFACE DESCRIPTION	6-19
6.4.1	External Interfaces	6-19
6.4.2	Internal Interfaces	6-20
6.4.3	Human/Machine Interfaces	6-20
6.5	DETAILED DESIGN	6-20
6.5.1	Procedure Descriptions	6-21
6.5.2	Procedure Dependencies	6-29

CONTENTS (cont)

6.6	DFDs	6-44
6.7	SCREEN FLOWS	6-60
7.0	SYSTEM USER DOCUMENTATION	7-1
8.0	SOFTWARE CONFIGURATION CONTROL	8-1
8.1	PURPOSE	8-1
8.2	SCOPE	8-1
8.3	MANAGEMENT	8-1
8.3.1	Organization	8-1
8.3.2	Responsibility	8-2
8.3.3	Interface Control	8-3
8.3.4	Implementation	8-4
8.3.5	Policies and Procedures	8-4
8.4	SOFTWARE CONFIGURATION ACTIVITIES	8-5
8.4.1	Configuration Identification	8-5
8.4.2	Configuration Control	8-5
8.4.3	Configuration Status Accountability	8-5
8.4.4	Audits and Reviews	8-5
8.4.5	Access Control	8-6
8.4.6	Backup and Retrieval	8-6
8.4.7	Disaster Recovery	8-6
8.5	TOOLS, TECHNIQUES, AND METHODOLOGIES	8-6
8.6	RECORDS COLLECTION AND RETENTION	8-6
8.7	CCB CHARTER	8-7
8.7.1	Purpose	8-7
8.7.2	Scope	8-7
8.7.3	Membership	8-7
8.7.4	Membership Responsibilities	8-7
8.7.5	Change Disposition	8-9
8.7.6	CCB Meetings	8-9
9.0	TEST PLAN	9-1
9.1	AMS MODULES	9-1
9.2	USER PROCEDURES	9-1
9.3	FEATURES TO BE TESTED	9-1
9.4	FEATURES NOT TO BE TESTED	9-2
9.5	AMS TESTING APPROACH	9-2
9.5.1	Job Stream Testing	9-3
9.5.2	Interface Testing	9-3
9.5.3	Security Testing	9-3
9.5.4	Regression Testing	9-3
9.5.5	User/Machine Testing	9-3
9.6	ITEM PASS/FAIL CRITERIA	9-3
9.7	TEST PROCEDURES	9-4
9.7.1	Database Certification	9-4
9.7.2	Menus	9-5
9.7.3	Update Screens	9-5
9.7.4	Display Screens	9-5
9.7.5	Report Screens	9-5

CONTENTS (cont)

9.7.6	Security	9-6
9.7.7	Admin Functions	9-6
10.0	V&V FINAL REPORT	10-1
11.0	REFERENCES	11-1
12.0	ACRONYMS AND GLOSSARY	12-1
12.1	ACRONYMS	12-1
12.2	GLOSSARY	12-1

LIST OF FIGURES

2-1	Hanford Waste Vitrification Plant Activity Management System Software Development Schedule	2-9
6-1	Activity Management System Context Diagram	6-44
6-2	Activity Management System Level 1 Activity Management System Data Flow Diagram	6-45
6-3	Activity Management System Level 2 Main Menu Data Flow Diagram . .	6-46
6-4	Activity Management System Level 2.1 Create Activity Data Flow Diagram	6-47
6-5	Level 2.1.2 Activity Data Entry Data Flow Diagram	6-48
6-6	Level 2.3 Retrieve Activity Data Flow Diagram	6-49
6-7	Level 2.3.2 List Retrieved Activities Data Flow Diagram	6-50
6-8	Level 2.3.2.3 Edit Activity Data Flow Diagram	6-51
6-9	Level 2.3.2.4 Clone Activity Data Flow Diagram	6-52
6-10	Level 2.4 List Management Data Flow Diagram	6-53
6-11	Level 2.5 Validate Activity Data Flow Diagram	6-54
6-12	Level 2.6 Administration Functions Data Flow Diagram	6-55
6-13	Level 2.6.1 Administration User Functions Data Flow Diagram . . .	6-56
6-14	Level 2.6.2 Administration Activity Functions Data Flow Diagram	6-57
6-15	Level 2.6.3 Administration Miscellaneous Functions Data Flow Diagram	6-58
6-16	Activity Management System Entity-Relationship Diagram	6-59
6-17	Activity Management System Screen Flow	6-60
8-1	Change Request and Problem Report Guidelines	8-10
8-2	Software Revision Record Guidelines	8-12

LIST OF TABLES

6-1	Data Design Entities	6-6
6-2	Each Procedure that a Procedure Calls	6-30
6-3	Each Procedure that Calls a Specific Procedure	6-37

SMALL JOB DEVELOPMENT SOFTWARE PROJECT DOCUMENTATION FOR THE ACTIVITY MANAGEMENT SYSTEM

1.0 INTRODUCTION

This single document, specified by *Software Practices, Small Job Development* (WHC-CM-3-10, SP-3.4), contains all the required documentation for the Activity Management System (AMS). Each section represents one of the necessary areas of documentation.

- Section 1.0 presents a general introduction to the project.
- Section 2.0 contains the system project management planning.
- Section 3.0 documents the customer's acceptance criteria.
- Section 4.0 contains the system requirements specification.
- Section 5.0 contains the Alternatives Analysis.
- Section 6.0 contains the system design description (SDD).
- Section 7.0 contains a reference to WHC-SD-HWV-UM-004, *Activity Management System User Reference Manual* (Gates 1994).
- Section 8.0 consists of the Software Configuration Management Plan (SCMP) and Configuration Control Board (CCB) charter.
- Section 9.0 contains the Test Plan.
- Section 10.0 is reserved for a copy of the final Verification and Validation (V&V) Report.
- Section 11.0 lists the references used in this document.
- Section 12.0 is used for acronym definitions and a glossary of unfamiliar terms.

1.1 OVERVIEW

The AMS has been developed out of an initiative of the Tank Waste Remediation System (TWRS) Management and Technical Systems organization for support of the Hanford Waste Vitrification Plant (HWVP) Project. The HWVP will be a new facility, constructed in the 200 East Area of the Hanford Site. The HWVP will facilitate Site radioactive waste cleanup by immobilizing Site defense liquid high-level and transuranic wastes in borosilicate glass for interim storage at the Site until shipment to an approved U.S. Department of Energy (DOE) geologic repository for permanent disposal.

The HWVP Project comprises four groups of contractors along with the U.S. Department of Energy, Richland Operations Office (RL).

- The RL Vitrification Project Office is responsible for management of the HWVP Project.
- Fluor Daniel, Inc. is the Architect-Engineer contractor.
- UE&C-Catalytic, Inc. is the General Construction Contractor.
- Pacific Northwest Laboratory is the Research and Development Contractor.
- Westinghouse Hanford Company (WHC) is the Integrating Contractor.

Project participants establish numerous commitment action items. A commitment item can consist of a project milestone, action items from meetings, or other transactions between project participants. These commitments need to be managed and tracked to facilitate the successful accomplishment of the work required for the HWVP Project.

A business concern existed in the tracking of commitments. While many methods of tracking work commitments are currently in use within the project and around the Hanford Site, none was deemed as fully supporting the needs of the project. Currently, project participants maintain several individual data files, automated and manual, to support the commitment tracking activity. Because of this, a comprehensive view of the commitment actions for the entire project is difficult to obtain. The system to be developed under control of this document will be called the AMS.

The AMS will be used to accurately track and monitor commitment action items. It must be accurate and validated and yet flexible enough to allow ad hoc manipulation of data in a multitude of ways. The system will be highly interactive. The originators of the commitment records, using their own personal computers (PC), will be able to create the activities and assign individuals to act as agents. These agent designees can then, in turn, validate their role by using their PC to either accept or refuse the commitment. Lack of this interactive capability and the inability to personally access the data is what makes the current automated systems unacceptable. Users also will be able to use the system as a personal "to do" list.

1.2 PURPOSE

The purpose of this document is to describe and specify the AMS and show how it will be developed and maintained to ensure a high-quality product. This document and WHC-CM-3-10, *Software Practices*, will define the management practices to be used in developing and implementing the AMS. This document will be enhanced, and new versions issued, as more information is developed defining the AMS requirements and the direction to be taken in achieving these requirements.

1.3 SCOPE

The AMS will store commitments and track, monitor, and report the status of these commitments. This document will outline the processes the AMS will use for accessing, analyzing, reporting, data entry, validation, and storage of HWVP Project commitment action items.

This page intentionally left blank.

2.0 SYSTEM PROJECT MANAGEMENT PLANNING

This section describes the management and procedures for development of the AMS application. It is intended to be used as a guideline by the customer, management, and the developers to define the AMS scope of work and to provide a detailed plan of how this work is to be accomplished.

2.1 PROJECT DELIVERABLES

The deliverables covered under this plan consist of the following:

- The tested AMS installed on the customer's file server and end users' computers. The actual AMS computer software will be written to floppy disks and delivered to the CCB.
- This document as specified in WHC-CM-3-10, *Software Practices*.
- WHC-SD-HWV-UM-004, *Activity Management System User Reference Manual* (Gates 1994).

2.2 PROCESS MODEL

The process model for the AMS Project does not follow the standard system/software "waterfall" life-cycle model. The AMS will be developed using a prototype methodology. That is, after an initial phase of concepts definition, analogous to the standard Concept Phase, a working model of the system will be created and installed for user evaluation. After a trial period, the user, based on experience gained, will review and update the requirements, system flow, etc., and the developers will incorporate these changes into the prototype model. This will be an iterative process that will be repeated until the user is satisfied that all requirements have been captured by the model.

At this point, the Requirements Specification (found in Section 4.0 of this document) will be updated and "frozen," and the software will be modified, as necessary, to provide a satisfactory production platform. Because of the 4GL tools available in the Paradox* environment, it is entirely possible that no reprogramming will be required at this point. For this reason, this process combines the traditional Requirements, Design, and Implementation Phases.

All the functions defined in the Requirements Specification will have been implemented at this point in the process. The AMS then will be subjected to a traditional Test Phase and Installation and Checkout Phase and placed under configuration control prior to starting the Operation and Maintenance Phase.

*Paradox is a trademark of Borland International, Inc.

It is anticipated that a future development cycle will include the production of a "portable" version--one that will support the use of AMS and its data on laptop portable computers. The special requirements (mainly involving extraction and inputting of data) will be documented in a future revision of the AMS documents after this enhancement activity has been approved and funded by the CCB (see Section 8.0).

Other future phases are expected to include the ability to import data from other Site commitment tracking systems (CTS), such as QUEST.

The traditional Retirement Phase is beyond the scope of this plan.

2.3 REVIEWS

Frequent customer reviews will be used to develop an initial functional definition to be used in building the first prototype. This document will be reviewed and approved by the customer prior to start of work on the software.

Continuous customer inputs will be encouraged and solicited during use of the prototype to identify problems, additional user requirements, opportunities to better match the user interface with the customer, etc.

All project documents, test procedures, test results, etc., will be approved and reviewed by Information Resource Management (IRM) Development Support, Quality Assurance (QA), and the customer.

A presentation will be made to the Data Administration Council (DAC) early in the project to inform them of the methods to be used and functions to be performed. This organization will be given access to the prototype to perform continuous monitoring. Data names used in the codes will be reviewed with DAC personnel to ensure compliance with Site standards.

2.4 ORGANIZATIONAL STRUCTURE

2.4.1 Customer Organization

The AMS customer organization is as follows:

T. G. (Tom) Halverson	Manager of the TWRS Management and Technical Systems - WHC cognizant manager in charge of the AMS Project.
K. J. (Kim) Carello	Secretary of TWRS Management and Technical Systems - AMS administrator for prototype and production use of the AMS.
T. M. (Terri) Lutter	Data Administration liaison for TWRS - technical interface to AMS development process.

2.4.2 Performing Organization

The AMS performing organization is as follows:

R. L. (Rich) Larson	Manager of the IRM Environmental Waste Management Systems - BCS Richland, Inc. (BCSR) manager in charge of TWRS (and HWVP) support.
J. S. (Jack) Trent	Manager of the IRM Environmental Systems Planning - BCSR manager in charge of HWVP support and the AMS Project.
T. A. (Ted) Gates	IRM Administrative and Technical Project Manager of the HWVP support - project manager and software engineer on the AMS Project.
M. B. (Mike) Burdick	IRM software engineer, Environmental Systems Planning - primary developer of the AMS design and software.
E. C. (Ed) Brann	IRM software engineer - Developed AMS test procedures.

2.4.3 Other Supporting Organizations

Other organizations that support the AMS are as follows:

K. C. (Kevin) Christensen	Data Standards and Administration - Reviewed AMS data definitions for DAC.
W. J. (Bill) Jones	IRM Information Engineering Services and Standards - IRM V&V consultant. Reviews AMS documentation as an independent reviewer and provides guidance on V&V-related activities.
M. L. (Mark) Hermanson	QA - Reviews and approves the documentation on the AMS Project.

2.5 PROJECT RESPONSIBILITIES

Project responsibilities for the AMS Project are of managerial, technical, and quality-assuring nature. The general responsibilities of each team member are in the following sections.

2.5.1 Customer Organization

The customer organization is as follows:

- WHC Cognizant Manager: Responsible for interaction between the customer organization and the development team and for customer acceptance, reviews, and approvals.
- WHC AMS Administrator: Responsible for assigning user identifications (ID) to customers who are authorized to use the system and for performing the other necessary administrative functions such as generating administrative reports, editing user information, and transferring activities from one user to another.
- WHC Data Administration Liaison: Responsible for representing customer data requirements, interfacing with users of the AMS, and interfacing with the Data Standards and Administration (DSA) organization.

2.5.2 Performing Organization

The performing organization is as follows:

- IRM Administrative Project Manager: Responsible for the schedule, resource management, and personnel related to the project design and documentation. Serves as a point of contact between the user organization and the management of IRM.
- IRM Technical Project Manager: Responsible for the design, coding, implementation, testing, and overall quality of the system deliverables. Includes actual design and implementation tasks as well as input to all deliverable documents. Directs the developers in work assignments and serves as a point of contact between the user organization and the IRM development team.
- IRM Manager: Responsible for the overall development process of the AMS Project: approvals, budget, personnel, schedule, resource utilization, and WHC management contact.
- IRM Software Engineer: Responsible for the design, coding, unit testing, and documentation of the AMS.

2.5.3 Other Supporting Organizations

Other supporting organizations are as follows:

- DSA: The DSA provides communication with the DAC and assists the AMS Project in complying with the WHC Data Administration Standards.

- QA: The QA representative is responsible for reviewing and approving software documentation in accordance with WHC standard engineering practices, WHC QA, and WHC software practices.
- V&V Consultant: The V&V consultant can act as the independent reviewer because the consultant is part of a separate Level 4 organization and is not part of the AMS Project.

2.6 PROCEDURES USED

Except for the prototyping methodology and its iterations, as defined in Section 2.2, the project followed the process guidelines defined by WHC-CM-3-10, *Software Practices*.

2.7 APPROVAL DESIGNATOR

The approval designator for the AMS Project and associated documentation has been determined to be N/A.

2.8 CHANGE AUTHORITY

Change authority during the development process will be vested in the AMS Project Team, under approval of the WHC cognizant manager and the IRM project manager.

Change control after the AMS is released is described in Section 8.0.

2.9 STATUS REPORTING

Status reporting will be accomplished through frequent meetings between the cognizant manager and the AMS Project manager and Project Team, use of the prototypes by the customer, and weekly reports to the WHC cognizant manager.

2.10 ASSUMPTIONS, DEPENDENCIES, AND CONSTRAINTS

2.10.1 Assumptions

The following are the basic assumptions of the project.

- The AMS will be developed to use the Paradox base currently in use by the customer.
- Approval will be received to use Paradox, Version 4.0, to overcome the limitations on text field size.
- The budget for the AMS Project will be available.

- The current user hardware will be used by the completed system.
- Customers will use the AMS prototypes and provide their impressions.

2.10.2 Dependencies

The following states the external dependencies of the project:

- Timely receipt of the necessary Paradox 4.0 upgrades and Paradox 4.0 runtime software for the development and user file servers
- Communication between AMS server and workstations via the Hanford Local Area Network (HLAN).

2.10.3 Constraints

The following constraints apply to the AMS Project:

- Timely availability of non-project user personnel to participate in user acceptance
- Timely completion of document reviews, V&V reviews, approvals, etc.

2.11 WORK TASKS

The following work tasks are part of the AMS.

2.11.1 Prepare Concepts Definition (Customer)

The HWVP Project will conduct a consensus session and develop a concepts definition document.

2.11.2 Define Preliminary Requirements

The developers will work with the customer to develop the final preliminary functional requirements prior to writing the first operating AMS prototype.

2.11.3 Write Functional Requirements Document

A document describing the functions and features to be incorporated into the AMS will be written by the development team and approved by the customer at the end of the previous step.

2.11.4 Customer Review and Approval of Functional Requirements

The customer reviews document and forwards comments. The document is revised until the customer approves the document as adequately describing an application that fully meets the needs of the HWVP Project.

2.11.5 Code First Prototype Version

The code will be written to produce an operating prototype that can be tested by the customer to ascertain functionality.

2.11.6 In-Process Demonstration to Customer

During coding of the prototype, make several demonstrations to customer of operating functions to get "midcourse" direction on suitability of approach.

2.11.7 Install First Prototype for Customer Evaluation

Install a fully functional version of the prototype to support at least six customer users to provide an evaluation of the AMS prototype in its actual operational environment.

2.11.8 Customer Evaluation--Prototype Update

Accept comments and suggestions from customer and incorporate them into the operating prototype while adding "production support" (administrative functions, reports, etc.).

2.11.9 Prepare Small Job Documentation

Write required small job documentation as specified in WHC-CM-3-10, *Software Practices*.

2.11.10 Prepare User Manual

Write a user reference manual in two versions. The first version is to be completed in time to support the initial use of the prototype by the HWVP Project evaluation users.

2.11.11 Prepare Test Procedures

Write the procedures required to implement the test plan (see Section 9.0 of this manual) and ensure a complete user acceptance test of the AMS.

2.11.12 Perform Customer Acceptance Test

Customer personnel will perform the specified test procedures and document the results they find.

2.11.13 Perform Final V&V Review

A final review of the project, documentation, and test results will be conducted by IRM V&V Support.

2.11.14 Receive Final Project Approvals

Approval of all materials will be received by the IRM cognizant engineer, IRM cognizant manager, customer cognizant manager, IRM V&V Support, and QA.

2.12 PROJECT SCHEDULE

The HWVP AMS software development schedule is presented in Figure 2-1.

HWVP AMS SOFTWARE DEVELOPMENT

(Timescale represents Months)

Project: Activity Mgmt. System
Date: 9/20/93
Page: 1

Critical Progress Summary
Noncritical Milestone

Figure 2-1. Hanford Waste Vittrification Plant Activity Management System Software Development Schedule.

This page intentionally left blank.

3.0 ACCEPTANCE CRITERIA

Acceptance of the AMS Project will depend on the program's capability to maintain the data in the AMS database, and to deliver the data to end-function users at local terminals, as defined in Section 4.0. Final acceptance will depend on completion and approval of testing plans and V&V of the system.

This page intentionally left blank.

4.0 REQUIREMENTS SPECIFICATION

4.1 INTRODUCTION

The AMS software requirements specification (SRS) constitutes the complete system requirements for the AMS. The information in this section will be used to design and test the automated system.

The contents of this section follow the guidance in WHC-CM-3-10, *Software Practices*, Section SP-3.4 and Appendix H; and ANSI/ASME 830-1984, *IEEE Guide to Software Requirements Specifications* (ANSI/IEEE 1984).

4.1.1 Purpose

This section contains the primary document that establishes requirements for the AMS software. The purpose is to provide the customer, performing, and supporting organizations with a full description of the requirements for the AMS. The SRS also provides input to the AMS test process and is used as the basis for a significant portion of the AMS V&V procedures.

This SRS provides a concise definition of the system software requirements that serve as direct input to the system design process. By emphasizing functions to be performed (the "what"), rather than software architecture ("the how"), the SRS will not be restrictive and will allow the software developers to exercise maximum flexibility and creativity during system design.

4.1.2 Scope

The scope of this SRS is limited to defining the software requirements for the AMS. The requirements in this document pertain to the entire AMS and no distinction is made between procured software and applications software to be developed during this project.

The software products to be produced are applications software for the AMS. Host software products, (e.g., a relational database management system [RDBMS] and peripheral host products such as forms generator software to aid the programmer in building data entry screens and a report generator to aid in building reports) will be provided for systems developers.

4.1.3 Definitions

Definitions and terminology used in this document will follow ANSI/IEEE 729-1989, *Standard Glossary of Software Engineering Technology* (ANSI/IEEE 1989). Deviations or additions will be defined in Section 12.0. Five of these definitions are included here as a convenience for readers without access to ANSI/IEEE 729-1989. These are as follows:

- Hardware--Physical equipment used in data processing, as opposed to computer programs, procedures, rules, and associated documentation

- Software--Computer programs, procedures, rules, and associated documentation and data pertaining to the operation of a computer system
- Requirement--(1) A condition or capability needed by a user to solve a problem or achieve an objective; and (2) a condition or capability that must be met by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. The set of all requirements forms the basis for subsequent development of the system or system component.
- Specification--A document that prescribes, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a system or a system component.
- Requirements Specification--A specification that sets forth the requirements for a system or a system component; for example, a software or hardware configuration item. A requirements specification typically includes functional, performance, interface, and design requirements, as well as development standards.

4.1.4 Overview

The general structure of this SRS progresses from a general description of the AMS environment to the specific software requirements necessary to support the functionality of the system. Section 4.2 contains descriptive information on the AMS product function, user characteristics, constraints, assumptions, and dependencies.

Section 4.3 defines the specific requirements that the system software must satisfy. Detailed software functionality is contained in Section 4.3.1. The required performance characteristics are found in Section 4.3.2 and the external interface requirements are contained in Section 4.3.3. Additional information presented in this SRS includes design constraints, and software system attributes.

4.2 GENERAL DESCRIPTION

This section summarizes the user needs that are the basis for the development of the AMS. The primary purpose of this summary is to provide information that will make the specific requirements in the next section easier to understand. Although this summary is intended to be accurate and complete, Section 4.3 constitutes the specific requirements agreed upon by the user and the performing organization. Thus the description in this section is not to be construed as implying any specific requirements that are not explicitly stated in Section 4.3.

4.2.1 Product Perspective

The AMS will be a single, fully integrated system that replaces the existing CTS currently being used by the HWVP Project Office. To meet the

currently identified organizational needs, new data elements will be added and changes to existing data fields will be made. The AMS will be based on the relational data model and developed using Paradox 4.0 RDBMS software.

4.2.2 Product Functions

The AMS will support tracking, reporting, data entry, and data storage of activities. An activity is defined as any work-related event that the AMS user desires to document. Standard PCs connected to the HLAN will be used by the end users to access the AMS.

The AMS will accurately record any activity entered by an end user. That user then will be designated as the originator of that activity. Optionally, an AMS user can be assigned as an agent on the activity and would be responsible for carrying out the action described in the activity description. Activities can be created with the originator as agent. This type of activity is known as a "personal" item.

When an activity is first assigned to an agent, the prospective agent will be sent a cc:Mail message announcing the assignment. The agent must then validate the activity by either accepting or refusing it. If the prospective agent chooses to refuse the assignment, the agent's name will be taken off the activity and the originator will be sent a cc:Mail message stating that the agent has refused. Prospective agents are not responsible for an activity (nor can an activity be considered delinquent) until they accept it.

If the activity is a private entry, where the originator is also the agent, no cc:Mail messages are sent, and the assignment is automatically accepted so that the originator does not have to validate the agent assignment.

The originator of an activity also can allow other AMS users to view specific activities. These viewers can only view the activity after it has been accepted by an agent. Only the originator and the agent of an activity will be allowed to update that activity.

The AMS will track the date an activity is created, the date the activity status was last updated, the date the activity was completed, and the date the activity was closed. The AMS administrator will be able to move closed activities to an archive table. Utilities will permit these archived activities to be reinserted into the active system tables.

The AMS will provide a variety of report formats that the originator, agent, and authorized viewers may use. These reports can be displayed on the screen, printed to a printer, written to a text file, or mailed to any cc:Mail user.

Flexibility in the ability to retrieve and view data is very important. The AMS will provide a variety of activity retrieval methods based on various data fields in the database.

The AMS will provide functions to create and maintain lists of AMS users, lists of activities or groups of activities, and lists of cc:Mail users. The

user lists will be used to assign viewers for activities and selecting activities by specifying the originator and/or agent. Activity lists will be used to group sets of activities for convenience in retrieving the grouped activities. The lists of cc:Mail users will be used to address cc:Mail messages that contain activity reports.

The AMS will provide administrative functions to allow the AMS administrator to edit user information, and add, delete, disable, and enable users. Additional administrative functions will support moving activities to the archive table, transferring a user's activities to another user, moving activities from the archive table back into the AMS, and copying or deleting the archive table. The system administrator also will be able to create user lists for any user to reference when adding viewers to activities.

The AMS will provide an on-line help facility. This facility will make essential information available to the user while operating the AMS.

4.2.3 User Characteristics

Users of the AMS will be divided into two categories, regular users and administrator users. Regular users will be those who are able to create, edit, report, and view activities and create lists. In addition to having access to all functions available to regular users, administrator users will have access to administrative functions such as adding users to the AMS. The AMS users will be required to have a basic working knowledge of computers or have attended an elementary computer class. No knowledge of Paradox will be required.

4.2.4 General Constraints

This section provides a general description of any items that limit the developer's options in designing the system. It is not intended to state specific requirements or design constraints, but provides the basis for specific requirements included in Section 4.3.

4.2.4.1 Document Approval Designation. This project has been assigned a document approval designator of N/A according to the criteria defined by WHC-CM-3-10, *Software Practices* and WHC-CM-3-5, *Document Control and Records Management Manual*, Section 12.7, "Approval of Environmental, Safety, and Quality Affecting Documents."

4.2.4.2 Data/System Recovery. The AMS will depend on the archive and recovery processes that the HLAN employs for backing up network file server computers. If a system failure occurs, users will not experience more than a single day's worth of lost data.

4.2.5 Assumptions and Dependencies

The major assumption for this project is that a suitable network file server will be available to be used for the AMS. Failure to obtain an adequate file server will result in delays while a network file server is procured.

4.3 SPECIFIC REQUIREMENTS

The specific requirements that must be satisfied by the AMS software are stated in this section. The objective is to provide the details the software developers will need to develop the design. These requirements must be complete and unambiguous. Any requirements that are not completely defined or that appear as new requirements during the design phase will be added to the AMS SRS. Each requirement also must be stated in such a way that its achievement can be objectively verified.

4.3.1 Functional Requirements

This section describes the fundamental actions that must take place in the software. The section is divided into the major AMS functions.

4.3.1.1 Data Elements. The following data elements will be required by the AMS:

Activity ID	Numeric
Activity title	Alphanumeric (30 characters)
Activity originator	Alphanumeric (30 characters)
Activity agent	Alphanumeric (30 characters)
Activity priority	Alphanumeric (3 characters)
Activity document ID	Alphanumeric (20 characters)
Activity description	Alphanumeric (0 - >2,048 characters)
Activity status	Alphanumeric (0 - >2,048 characters)
Activity deliverable	Alphanumeric (0 - >2,048 characters)
Original due date	(mm/dd/yy)
Current due date	(mm/dd/yy)
Completion date	(mm/dd/yy)
Closure date	(mm/dd/yy)
Status date	(mm/dd/yy)
Creation date	(mm/dd/yy)
Validation status	alphanumeric (1 character)
User organization code	alphanumeric (6 characters)
User location	alphanumeric (20 characters)
User login date	(mm/dd/yy)
Activity category code	Alphanumeric (16 characters)
Activity category description	Alphanumeric (49 characters)
Activity prerequisite code	Alphanumeric (3 characters)
Activity prerequisite description	Alphanumeric (20 characters).

4.3.1.2 Querying. The user must be able to retrieve activities by group, role, match, completion, delinquency, prerequisite, and category.

The capability of retrieving activities must be limited to the originator, agent, and designated viewers. Access to open activities to which agents have been assigned but who have not accepted must be limited to the originator via the retrieve function or to the agent via the validate function. Closed activities may be viewed by the originator, agent (if any), and viewers. If a closed activity has an assigned agent who has not validated the assignment, the agent information will be deleted.

The "retrieve by group" query will permit convenient retrieval of activities that have been grouped together by the originator. The user will be able to select the activity group to be retrieved from a list displaying all the activity groups that the user has created. Closed activities will not normally be retrieved unless the user indicates specifically that they should be included.

The "retrieve by role" query will return activities based on the role the user plays in the activity. The user can select any combination of the following roles: originator, agent, personal data (where the originator is the agent), or authorized viewer. Combinations of roles are processed by an "or" operation. That is, selecting "originator" and "agent" retrieves the records where the user is either the originator or the agent. (This is the reason for the "personal data" role.) Closed activities will not be included unless the user indicates that closed activities should be included.

The "retrieve by match" query will allow for retrieval by matching the contents of a number of activity data items. This function allows the user to apply wildcard and logical operators on data fields to retrieve activities. The user will be able to specify the originator and agent of an activity by specifying either an individual AMS user or a user list.

Wildcard and logical operators can be applied to the following fields: activity ID, activity title, activity priority, current due date, and status date. Any Paradox operand may be used. A help screen will be available to list a subset of Paradox operators to assist the user in formulating these queries. Closed activities will not be included unless the user indicates that closed activities should be included.

The "retrieve by validation" query will return activities that the user has originated that match a specified validation status. The user will be able to specify any combination of the following validation statuses: waiting for validation, accepted, refused, not assigned, and closed.

The "retrieve by completion query" will return activities that have been completed. Additionally, the user must specify the role(s) to retrieve (as described in the "retrieve by role" query). Closed activities will not be included unless the user indicates that closed activities should be included.

The "retrieve by delinquency" query will retrieve activities that are delinquent. An activity is considered delinquent if it meets all the following criteria.

- The completion date field is blank.
- The current date is less than the current due date field.

- The activity has been accepted by the agent.
- The activity is not closed.

Users will specify which role they play in the activity. Only the originator or agent role may be specified.

The "retrieve by prerequisite" query will retrieve activities that contain a selected prerequisite code. The user will select the desired prerequisite code from a list of codes.

The "retrieve by category" query will retrieve activities that contain a selected category code. The user will select the desired category code from a list of codes.

After the query is performed a list of retrieved activities will be displayed to the user. The following fields from the activities will be displayed on this list: activity title, validation status, activity originator, activity agent, and current due date. The user will be able to sort this list of retrieved activities by any of the displayed fields. The user will be able to select any or all of the retrieved activities and only those selected activities will be operated on by any activity functions the user selects.

4.3.1.3 Reporting and Viewing. Users will have five different reports that can be run on activities. To run these reports the user must first run a query, then select the activities to be included on the report from the list of retrieved activities, and then run the report function. The formats of the reports are described below.

- The full format report will contain all the fields that comprise an activity.
- The summary format will be the same as the full format but will not contain the activity description and activity deliverable fields.
- The landscape brief format will contain the following fields:
 - Activity agent
 - Activity title
 - Current due date
 - Completion date
 - Activity ID
 - Activity priority
 - Validation status.

Each activity will be printed on one line and in landscape mode.

- The portrait brief format will contain the following fields:
 - Activity agent
 - Activity title
 - Current due date
 - Completion date.

Each activity will be printed on one line and in portrait mode.

Each of the above reports will be available in three sort orders: activity agent, activity title, and current due date. It will be possible to direct each of the above reports (except the landscape brief report) to a printer, text file, or cc:Mail message. Because of the constraints of the cc:Mail system, the landscape report cannot be sent in a cc:Mail message.

Reports that are directed to a printer will be available in two different paper sizes: letter size (8.5 by 11 in.) and planner size (8.5 by 5.5 in.). The planner size will be printed on a letter size piece of paper but the printed portion will only cover an 8.5- by 5.5-in. area. In addition, all the reports except the landscape brief report will be able to be viewed with data entry screens and the memo editor.

- The fifth report format will list all the authorized viewers for each selected activity. This report will contain the following fields:
 - Activity ID
 - Activity title
 - Activity originator
 - Activity agent
 - The user names for each authorized viewer (or, if there are no authorized viewers for the activity the function will say "No enabled viewers"). This report can be sent to a printer or written to a text file.

Users will be able to run the full report on all the activities for which they are the originator and which are awaiting validation. The user will not have to retrieve any activities to run this report. This report will be selected from a separate menu function and will contain all such activities. This report can only be sent to a printer or written to a text file.

The administrator will be able to run a report that will list all the unvalidated activities. The administrator will input the number of days since the unvalidated activity was created, greater than or equal to 7 days, and the report will list all activities that have been left unvalidated for a time period greater than that. The resulting report will be grouped by user organization code. The following fields will be on the report:

- User organization code
- Activity agent
- Activity title
- Creation date
- Activity originator.

This report can only be sent to a printer.

4.3.1.4 Creating Activities. Users will be able to create activities in two ways. First, a user can select the create activity function from the main menu and then enter the activity information through a data entry screen. Second, users will be able to copy any activity they are allowed to view and make any desired changes through the data entry screen. This copy process will be called "cloning" an activity.

Cloning will be available at two different points in the AMS process: (1) a user will be able to clone an activity immediately after creating it via the create activity function, and (2) a user will be able to clone existing activities by running a query, selecting the activities to be cloned, and performing the clone function.

When an activity is created or cloned the user will be able to edit the following fields:

- Activity title
- Activity agent
- Activity priority
- Activity document ID
- Original due date
- Current due date (clone only)
- Activity description
- Activity status
- Activity deliverable.

The activity ID will be generated by the system, the current due date will be the same as the original due date, and the activity originator will be the name of the user.

4.3.1.5 Editing Activities. Users will be able to edit any activity for which they are the originator or agent. Originators will be able to edit (update) the following fields:

- Activity title
- Activity agent
- Activity priority
- Activity document ID
- Original due date if the activity has not been accepted by the agent
- Current due date
- Activity description
- Activity status
- Activity deliverable
- Prerequisite code
- Category code.

Agents will be able to edit the following fields:

- Activity completion date
- Activity status.

To edit activities, the user will retrieve activities by a query, select the activities they wish to edit, and select the edit function.

4.3.1.6 Data Entry. Activities will be created, updated, or viewed through a common data entry screen. Three of the fields on this screen will be memo fields. These differ from the other data fields in that they can hold a nearly unlimited amount of data and may not show all their data on the data entry screen. To edit any of the memo fields, a full-screen editor will be used. Any functions that can change an activity's data will be disabled if the data entry screen is being used in the viewing function.

The data entry screen will display all the data fields of the activity. The fields that the user can edit will have a blue background. Those fields that a user cannot edit will have a pink background. These color designations (and the input restrictions they represent) will change depending on the role the user has in the activity--originator or agent.

A function for assigning an agent to an activity will be available from the data entry screen menu and available from the create or edit functions. Selecting this function will enable the originator to select an agent from a list of AMS users and automatically insert the selected agent's identifying data into the activity. When an agent assignment is made and the activity is saved, a cc:Mail message will be generated and sent to the selected agent.

If the originator is also assigned as the agent, thus creating a "personal" activity, no cc:Mail message is sent and the assignment is automatically validated, so that the originator does not have to perform the validation function.

Functions for selecting prerequisite and category codes will be available from the data entry screen menu and available from the create or edit functions. Selecting these functions will enable the originator to select a code from a list of predefined prerequisite or category codes depending on the function selected. When the code is selected, it is inserted into the appropriate field.

An undo function will be provided to back out any changes made by the user while the data entry screen is open. To move to a different field on the screen, the user must be able to use the cursor movement keys, tab key, or the mouse. When a user is creating or cloning (see Section 12.0) an activity, a cancel function will be available to quit the activity data input function and delete any data that already has been entered.

Three fields on the data entry form will be memo fields. The memo editor will be a full-screen text editor. The following functions will be available while editing the memo field:

- Read a text file into the memo field
- Write the contents of the memo field to a text file
- Print the contents of the memo field
- Search for text strings
- Repeat last text search
- Text search and replace
- Mark and unmark block of text
- Copy block of text
- Cut block of text
- Paste block of text

- Delete block of text
- Undo any changes made to memo field.

4.3.1.7 Adding Viewers. Access to an individual activity's data is normally limited to its originator and agent. A provision must be implemented to allow the originator of an activity to designate other AMS users as "viewers" on that activity. Such viewers will be able to look at but not update the data. The originator will be able to specify individual AMS users, or assign predefined lists of users. These may be lists created by the originator or more general lists that the system administrator has created. If a user list is specified, any user on that list will be able to view that activity. Viewers will be added to activities at the time the activity is created or later, by running a query, selecting the desired activities, and selecting the add viewers function. Viewers can be added to activities individually or can be added to multiple activities that have been selected in the query all at once.

4.3.1.8 List Management. Three different kinds of lists can be created by the user: AMS users, activities (called activity groups), or cc:Mail users. Functions will be available to create, copy, delete, and append other lists to these lists. Functions also will be provided to delete or add entries to the lists. Users will be able to add activities to activity groups by running a query, selecting specific activities, and then selecting an activity group.

The system administrator will be able to create AMS user lists that any AMS user can use. These lists can be used to group AMS users into units that are important to the overall organization using the AMS.

4.3.1.9 Help Facility. Help will be available to the user via a main menu selection. An index of topics will be displayed and the user can select one and view the information associated with that topic.

4.3.1.10 Closing and Reopening Activities. Only the originators of activities will be able to close them. Once an activity is closed for at least 2 weeks, it can be purged from the system and placed in an archive table by the AMS administrator. To close activities, retrieve the activities with a query, select the activities to be closed, and select the close function. If an activity that has not been accepted by an agent is closed, the activity agent field is cleared.

Activities that have been closed also can be reopened by the originator if they have not been purged to the archive table. If an activity has been purged, the system administrator is able to retrieve the activity with the unpurge function. To reopen activities, retrieve activities with a query, select the activities to be reopened, and select the reopen function.

4.3.1.11 Activity Validation. When a user logs into the AMS the number of any unvalidated activities for which the user is the agent will be displayed. To validate an activity, the user will select the validation function from the main menu. All the user's unvalidated activities will be retrieved.

Each activity will then be displayed, in turn, on a data entry screen. The user will be able to view the activity data, including the description and activity deliverable fields via the memo editor. The user will be able to

update the activity status using the memo editor. The user can reject or accept the activity. If the activity is accepted, the validation status changes to accepted and any authorized viewers can retrieve and view the activity. If the activity is refused, the user's name is cleared from the activity agent field, the originator is notified of the refusal with a cc:Mail message, and the validation status is set to "not assigned."

4.3.1.12 Administrative Functions.

Administrative functions will be divided into three different groups: user functions, activity functions, and miscellaneous functions. Only users that are administrative users will have access to these functions. User functions will primarily work on attributes of AMS users. The following user functions will be available.

- Add User - This function will add users to the AMS. The administrator will select the user from a copy of the cc:Mail user list that is maintained within the AMS database. The cc:Mail user data, including the name, location, etc., is automatically moved to the AMS user record. Additionally, other user data is entered by the administrator. This user data consists of the following:
 - Payroll number
 - Organization
 - Initial password
 - User type (administrative or regular).
- Delete User - This will mark a user for removal from the system. Only users who are not originators or agents of any activities that are not closed can be deleted. The user will actually be removed by the purge function.
- Disable User - This will prevent a user from logging into the AMS or being identified as an agent of an activity.
- Enable User - This will allow a user who has been disabled to log into the AMS and/or be chosen as an agent of an activity. The enable user can also be used to reactivate a deleted user if run prior to the purge.
- Edit User - This will change information about a user.
- Change User ID - This will change a user's ID.
- List Users On-line - This will list any users who are currently logged into the AMS.

Activity functions will primarily work on AMS activities. The following activity functions will be available.

- Unvalidated Report - This will print the report that lists all unvalidated activities.

- **Transfer Activities** - This will transfer one user's activities to another user.
- **Purge Activities** - This will remove closed activities from the AMS. The user will enter the number of weeks an activity has to have been closed to be purged. The minimum number of weeks will be two. All activities that have been closed for at least the number of weeks the user entered will be copied to an archive table. Any users who have been marked for removal and are no longer originators or agents of any activities still in the system will be removed from the system at this time. The archive table will be named after the current year; for 1993 it will be named ams1993. When a new year occurs, or if the archive table does not exist, the purge function will automatically create a new table.
- **Unpurge Activities** - This will reinsert activities that have been purged from the AMS. The user will select which archive table to choose from and then the activities within that archive table will be displayed. The user will then select which activities to unpurge.
- **Remove Archive Tables** - This will allow the user to copy or delete any archive tables that exist.

Miscellaneous functions will perform any additional administrative work that is needed. The following miscellaneous functions will be available.

- **Organization Change** - This will change every occurrence of an organization code to another organization code.
- **Update cc:Mail List** - This will update the list of cc:Mail users.
- **Global User Lists** - This will let the user create and maintain any global user lists.

Paradox will allow AMS administrators with the full Paradox software (not the runtime version) running on their computers to enter Paradox and retrieve data (with read-only privileges) to perform Paradox commands.

4.3.1.13 User Password. Users will be assigned a password when they are first entered into the system. Users must be able to change their own password. A function will be provided for that purpose. The user will type in the current password and then enter the new password twice. If the current password typed in matches the current password in the system and the two copies of the new password match, the password will be changed. From that point, the new password will be required for the user to log into the AMS.

4.3.2 Performance Requirements

Number of Concurrent Users Supported. The application must be able to support a minimum of five concurrent users. Access to the AMS will be made by PCs connected to a file server via HLAN.

Disk Storage. Disk capacity will be required for the following components:

- Database management software
- Applications code development libraries
- Production code libraries
- User database.

Printing. Local or network laser printers will be used for reports. The Hewlett-Packard LaserJet III^{*} will be considered to be the standard printer. All reports should be able to be printed on this printer.

Response Times. It is recognized that Paradox applications do not effectively use the network. Response times will vary depending on the current traffic of the HLAN and the current activity on the network file server. Other factors are the type of network file server used and the type of PC at the user's location. Because of this, no response time specification will be stated.

4.3.3 External Interface Requirements

External interfaces are those interfaces outside the application software product that is being developed. These include the following:

- The user
- The hardware
- Other software products.

User interfaces describe the characteristics of the human interaction with the system, such as how to access the AMS using an IBM-PC^{**} connected to the HLAN. Hardware interfaces describe the interface of the application software with the hardware, such as devices to be supported protocols to be used. Software interfaces describe the application software interfaces to the vendor-supplied software products, such as the use of the RDBMS.

4.3.3.1 User Interfaces. User interfaces consists of the inputs, outputs, and displays on a display terminal used or seen by an AMS user. All user data interchange with the AMS will be via menu, dialogue box, and/or data entry screen. Users will be able to use the keyboard or the mouse to select menu choices and enter data. Each screen displayed will be structured for ease of use. Error messages, along with any possible resolution options that the user can be expected to understand, will be displayed on the screen when the AMS detects an error.

^{*}Hewlett-Packard and LaserJet are trademarks of the Hewlett-Packard Corporation.

^{**}IBM is a trademark of International Business Machines, Inc.

The AMS will employ the following menu and screen organization. Menus will enable the choice of the following functions:

- All administration functions
- Create activity
- Activity queries
- Validate activities
- Maintain lists
- Run unvalidated activity report
- Help
- Change password.

List management functions will use dialogue boxes for list content selection and data item entry. Each of the three types of list management will employ similar dialogue boxes to ensure a consistent interface.

After an activity query is run, the retrieved activities screen and menu will be displayed. This menu will allow selection of the following functions:

- Edit activities
- Clone activities
- Report and view activities
- Add activities to a group
- Add viewers to activities individually or as a group
- Run activity viewer report
- Close activities
- Reopen activities
- Sort by:
 - Title
 - Validation status
 - Originator
 - Agent
 - Due date
- Rerun query
- Return to main menu.

When the report and view activities function is selected from the retrieved activities screen, a dialogue box will be used to let the user select all the report options.

4.3.3.2 Hardware Interfaces. The AMS host hardware will be an HLAN network file server. The PCs will run the AMS at the user's end. The network file server will look and act like a hard drive to the user's computer. This interface will be transparent to the user. The user can execute the Paradox software from the local PC for better performance or from the file server (with decreased performance) if there is not enough local disk space available.

4.3.3.3 Communications Interfaces. The only communications interface is the HLAN. End users and development staff will access the AMS file server through the normal HLAN/Ethernet connection. The AMS login procedures and system usage are addressed in WHC-SD-HWV-UM-004, *Activity Management System User Reference Manual* (Gates 1994).

4.3.4 Design Constraints

4.3.4.1 Standards Compliance. The system must be compatible with applicable site standards. Data-naming standards for those items that have site standard names will be maintained. Site programming standards will be adhered to. Modularity, program identification, program documentation, and program constructs will be used consistent with the capabilities of the Paradox environment.

4.3.4.2 Hardware Limitations. The inefficient way that a Paradox application uses the network may mean that system response is not as fast as desired.

4.3.5 Attributes

This section describes system attributes that place specific requirements on the software. These include: ease of use, availability, security, maintainability, and transferability/conversion.

4.3.5.1 Ease of Use. To facilitate the system's ease of use, the following capabilities will be provided.

- **Menu Driven** - Menus will be provided for the user interface to the system.
- **Dialogue Boxes** - Interactive dialogue boxes will be provided for the user interface to the system.
- **Data Input** - The system will use a mouse and/or keyboard for the user interface to the system. A mouse shall not be required to run the AMS.
- **Error Messages** - Error messages will be unique and useful to the user. Every effort should be made to produce error messages that will help the user to recover from abnormal conditions that might occur while inputting or reporting on the data.
- **Update Messages** - Messages will be provided that clearly indicate the successful entry of data into the system.

The AMS will provide, at the time of entry, automatic checks and verification (edits) of the data entered with established acceptance criteria. These checks and data verification will be identified during detailed design.

The system will provide a user interface for data entry and information display which is easily understood with minimal training and does not arbitrarily impede use of the system.

4.3.5.2 Availability. The system will be available for user access anytime except during scheduled HLAN maintenance. Because some of the administrative functions require exclusive access to the database, there will be some time periods where regular user excess will not be permitted while these functions are running.

4.3.5.3 Security. Access to AMS data will be strictly controlled. Additional codes or passwords will be established to control update, reporting, and inquiry against activity records. In addition, certain data fields and data tables within an activity shall be protected against update access. This decision will be made during the design phase.

4.3.5.4 Maintainability. Analyst support will be provided during all phases of the project including the maintenance or operational phase of the project for user support functions.

Modifications to AMS in the form of system changes or reprogramming will be made as prescribed by the configuration management plan (see Section 8.0 of this document).

Specific data types and units, as supported by Paradox, will be determined during detailed design specifications. Procedures, screens, and global change architecture and methodology will be determined during the detailed design process.

4.3.6 Other Requirements

This section outlines other general requirements. Because of their unique nature, these requirements do not fit in any of the topics previously presented.

4.3.6.1 Database. Transactions will be applied to the database as they are entered, thereby presenting an up-to-date database to all users. In other words, transactions to the data will not be collected and applied periodically.

It will be possible to select records based on the contents of selected data fields or a combination of selected data fields. This capability will exist for reports and on-line data access.

Alternate access keys based on non-key data fields will be used to enhance system performance.

A system dictionary will be established that defines all codes and data elements.

4.3.6.2 Operations. The system will be available for normal operations except for scheduled HLAN maintenance and when administrative functions are being run by an administrative user.

All data will be backed up automatically by the HLAN file servers as part of the normal archiving procedure.

4.4 AMS MENUS

The AMS functions will be activated by making selections from on-screen pull-down menus whose titles identify the function (or function group) they initiate. These menus are listed below, grouped by the screen menu bar in

which they appear, with the hierarchical structure denoted by indentation. The pertinent section in the requirements description above is shown in brackets.

4.4.1 AMS Main Screen Menu Selections

1. System Menu
 - 1.1 Change Password [4.3.1.13]
 - 1.2 Default (not implemented)
2. Activity Menu
 - 2.1 Create [4.3.1.4]
 - 2.2 Validate [4.3.1.11]
 - 2.2.1 Validate (function) [4.3.1.11]
 - 2.2.2 Report [4.3.1.3]
 - 2.3 Retrieve (activity) [Section 4.3.1.2]
 - 2.3.1 By Group [4.3.1.2]
 - 2.3.2 By Role [4.3.1.2]
 - 2.3.3 By Match [4.3.1.2]
 - 2.3.4 By Validation [4.3.1.2]
 - 2.3.5 By Completion [4.3.1.2]
 - 2.3.6 By Delinquency [4.3.1.2]
3. Lists Menu [4.3.1.8]
 - 3.1 User Lists [4.3.1.8]
 - 3.2 Activity Groups [4.3.1.8]
 - 3.3 Mail Lists [4.3.1.8]
4. Help [4.3.1.9]
 - 4.1 Help Index [4.3.1.9]
5. Admin Menu [4.3.1.12]
 - 5.1 User Functions [4.3.1.12]
 - 5.1.1 Add User [4.3.1.12]
 - 5.1.2 Delete User [4.3.1.12]
 - 5.1.3 Disable User [4.3.1.12]
 - 5.1.4 Enable User [4.3.1.12]
 - 5.1.5 Edit User [4.3.1.12]
 - 5.1.6 Change User ID [4.3.1.12]
 - 5.1.7 List Users Online [4.3.1.12]
 - 5.2 Activity Functions [4.3.1.12]
 - 5.2.1 Unvalidated Report [4.3.1.12]
 - 5.2.2 Transfer Activities [4.3.1.12]
 - 5.2.3 Purge Activities (to archive table) [4.3.1.12]
 - 5.2.4 Unpurge Activities [4.3.1.12]
 - 5.2.5 Remove Archive Tables [4.3.1.12]
 - 5.3 Miscellaneous Functions [4.3.1.12]
 - 5.3.1 Org Change [4.3.1.12]
 - 5.3.2 Update cc:Mail List [4.3.1.12]
 - 5.3.3 Global User Lists [4.3.1.12]
 - 5.4 Paradox [4.3.1.12]
6. Exit
 - 6.1 Yes ("do it")

4.4.2 AMS Data Entry Form Menu Selections

1. MemoEdit [4.3.1.6]
 - 1.1 Description [4.3.1.6]
 - 1.2 Status [4.3.1.6]
 - 1.3 Deliverable [4.3.1.6]
2. Agent [4.2.2, 4.3.1.4, 4.3.1.5]
 - 2.1 Find Agent [4.2.2, 4.3.1.4, 4.3.1.5]
 - 2.2 Clear (existing agent) [4.2.2, 4.3.1.4, 4.3.1.5]
3. Previous
 - 3.1 Yes ("do it")
 - 3.2 Add Viewers [4.3.1.7]
4. Next
 - 4.1 Yes ("do it")
 - 4.2 Add Viewers [4.3.1.7]
5. Clone
 - 5.1 Yes ("do it")
 - 5.2 Add Viewers [4.3.1.7]
6. Exit
 - 6.1 Yes (exit to screen from which this screen was called)
 - 6.2 Add Viewers [4.3.1.7]
 - 6.3 Main Menu (exit to main menu screen)
7. Cancel
 - 7.1 Yes (exit to screen from which this screen was called)
 - 7.2 Main Menu (exit to Main Menu screen)
8. Undo (delete all changes made to the activity during this session)
 - 8.1 Undo ("do it")
 - 8.2 Cancel (return to current data entry screen)

4.4.3 AMS MemoEdit Field Menu Selections

1. Edit (NOTE: these are all standard Paradox MemoEdit field functions) [4.3.1.6]
 - 1.1 WriteFile [4.3.1.6]
 - 1.2 ReadFile [4.3.1.6]
 - 1.3 Print [4.3.1.6]
 - 1.4 Find [4.3.1.6]
 - 1.5 FindNext [4.3.1.6]
 - 1.6 Search and Replace [4.3.1.6]
 - 1.7 Cut [4.3.1.6]
 - 1.8 Copy [4.3.1.6]
 - 1.9 Paste [4.3.1.6]
2. Exit
 - 2.1 Save
 - 2.2 Undo

4.4.4 AMS Validation Screen Menu Selections

1. Options [4.3.1.11]
 - 1.1 Accept [4.3.1.11]
 - 1.2 Refuse [4.3.1.11]
 - 1.3 DoNothing [4.3.1.11]

2. ViewMemo
 - 2.1 Description (read-only access)
 - 2.2 Status (may update) [4.3.1.11]
 - 2.3 Deliverable (read-only access)
3. Previous
4. Next
5. Exit

4.4.5 AMS Retrieved Activities Screen Menu Selections

1. Functions
 - 1.1 Edit [4.3.1.5]
 - 1.2 Clone [4.3.1.4]
 - 1.3 Report and View [4.3.1.3]
 - 1.4 Group [4.3.1.8]
 - 1.5 Viewers [4.3.1.7]
 - 1.5.1 Individually [4.3.1.7]
 - 1.5.2 Group [4.3.1.7]
 - 1.5.3 Report [4.3.1.7]
 - 1.6 Close [4.3.10]
 - 1.7 Reopen [4.3.10]
2. Sort [4.3.1.2]
 - 2.1 by Title [4.3.1.2]
 - 2.2 by Status [4.3.1.2]
 - 2.3 by Originator [4.3.1.2]
 - 2.4 by Agent [4.3.1.2]
 - 2.5 by Due Date [4.3.1.2]
3. Check [4.3.1.2]
 - 3.1 All [4.3.1.2]
 - 3.2 Current [4.3.1.2]
4. Uncheck [4.3.1.2]
 - 4.1 All [4.3.1.2]
 - 4.2 Current [4.3.1.2]
5. Reselect [4.3.1.2]
 - 5.1 Yes ("do it")
6. Exit
 - 6.1 Yes ("do it")

5.0 ALTERNATIVES ANALYSIS

5.1 ALTERNATIVES CONSIDERED

Several possible alternatives were considered before arriving at the chosen solution. The purpose of the alternatives analysis is to justify the choice made by comparing it to the alternative solutions.

5.1.1 Criteria

The following criteria are the factors that affect the selection of the best alternative. For this system, the criteria include the items in the following subsections. In selecting the optimum approach, a balance must be struck between satisfying the first criteria and minimizing the costs associated with the other three criteria.

5.1.1.1 Satisfy the SRS. The solution must satisfy the SRS detailed in Section 4.0. The best alternative will satisfy a requirement at a higher level of performance; for example, it may complete a function faster or in a more user-friendly manner than an inferior solution.

5.1.1.2 Minimize Hardware and Software Cost. The cost of procuring new hardware and software is a consideration. A constraint of the SRS is to use as much existing hardware and software as possible.

5.1.1.3 Minimize System Development Cost. The cost of developing new software and connecting the system together is an important factor. Development cost usually is related to system complexity, so a simpler solution generally is more desirable.

5.1.1.4 Minimize Operating Cost. Once the system is operational, the cost of maintaining it must be minimized. Significant operating costs result from salaries for operating personnel, hardware maintenance, mainframe computer time, and backups. The amount of time required to perform some of the routine system operations, such as data updates and backups, also affects operating costs.

5.2 ALTERNATIVES

Possible alternatives discussed include the "no-action" alternative. Four design alternatives are examined: (1) buy commercial system, (2) use existing system, (3) enhance existing system, and (4) create new system. "Enhance existing system" is the chosen alternative.

5.2.1 Buy Commercial System

5.2.1.1 Description. A commercially available system that satisfies the SRS would be purchased and put into production.

5.2.1.2 Advantages and Disadvantages. The most attractive feature of this alternative is that the costs of implementing this system might potentially be less than developing a system in house.

At this time, no commercially available CTS has been identified that satisfies the AMS requirements sufficiently to be considered as a viable solution.

5.2.2 Use Existing System

5.2.2.1 Description. Find a CTS currently in use at the Hanford Site that satisfies the SRS and place this system into production.

5.2.2.2 Advantages and Disadvantages. This alternative could be very cost effective because little or no development and associated processes would be necessary.

Currently, there are approximately 68 CTS's in use at the Hanford Site. The majority of these systems are clones of one another. Approximately 20 of these systems are unique. Of these 20, about one-half are not multiuser systems which makes these unacceptable from the start. The remaining systems do not satisfy the SRS closely enough to make this a viable alternative.

5.2.3 Enhance Existing System

5.2.3.1 Description. Identify a CTS currently in use by the customer that (1) closely matches the AMS requirements, and (2) is written with an application development tool that facilitates cost-effective enhancement. Modify this system to fit the SRS.

5.2.3.2 Advantages and Disadvantages. Depending on the amount of modifications required, this alternative could entail less cost than developing a new system. Portions of the existing code and documentation could be reused, resulting in cost savings and quicker development time.

If the system chosen for enhancement does not match the SRS closely enough or requires use of expensive, difficult-to-use tools, this alternative could result in longer development time and more cost than developing a completely new system.

While it did not match the requirements as closely as might be desirable, it was decided to build the AMS on the existing HWVP CTS. This permitted the use of the existing Paradox software (upgraded to Paradox Version 4.0 to accommodate large text fields and provide enhanced performance) and as much as possible of the existing Paradox Application Language (PAL) code.

5.2.4 Create New System

5.2.4.1 Description. Develop a system from scratch that matches the SRS.

5.2.4.2 Advantages and Disadvantages. This alternative would have the advantage of exactly implementing the SRS, as it would not compromise in order to use the existing system's structure and code. The development tools could be selected for best suitability and least cost.

If there were an existing system that could provide a substantial base for enhancement to implement the desired functionality, this alternative would have a higher development cost.

5.2.5 No-Action Alternative

5.2.5.1 Description. The project participants would continue to track and monitor commitments with methods they are currently using.

5.2.5.2 Advantages and Disadvantages. This is the least expensive but is unacceptable because it does not satisfy the SRS. Continued use of the existing systems is unacceptable because project-wide commitment information is not easily obtainable.

This page intentionally left blank.

6.0 SDD

6.1 INTRODUCTION

The AMS SDD describes the physical architecture of the system described in the AMS SRS.

The AMS SDD formally partitions the elements of the system described in the AMS SRS into design entities and describes the key properties of and relationships between the design entities. The AMS SDD can be described as a detailed blueprint for implementation activities.

The AMS SDD is reviewed formally in order to comply with the QA program and WHC-CM-4-2, *Quality Assurance*, QI 3.2. This document will be reviewed against the AMS SRS and revised as necessary to reflect changes in the AMS requirements.

Content of this document follows the guidance in WHC-CM-3-10, *Software Practices*, and IEEE 1016-1987, *Software Design Descriptions* (IEEE 1987).

6.1.1 Purpose

Each function in the AMS SRS has been allocated to a design entity which can be traced to a design solution in the AMS SDD. The design solution identified is the one judged best able to meet the AMS SRS objectives within the identified constraints documented in Section 6.1.2. The purpose of this document is to describe the content and relationships of the design entities to which AMS SRS system functions have been allocated. The AMS SDD is used as the primary medium for communicating software design information.

6.1.2 Scope

The scope of this document is limited to a description of the software design for the AMS. This document is used as the primary medium for communicating software design information.

The software product to be produced is the applications software for the AMS. Host hardware and operating software, terminal devices, communications hardware and software, and host software products will be provided for systems developers.

6.1.3 SDD Structure

Each reader of this SDD may have a different view of what is considered necessary information to describe essential aspects of the software design. The general structure of this SDD supports the idea of multiple design views by organizing information about design entities and their associated attributes into separate design views. Each design view represents a separate concern about the software system. Both logical and physical representations of the design are presented. The AMS SDD is organized into six sections.

Section 6.1 provides an introduction to the document.

Section 6.2 provides references to related project documentation.

Section 6.3 describes the division of the software system into design entities. This chapter describes the way the system has been constructed and the purpose and function of each entity.

Section 6.4 describes the relationships among the design entities. This chapter defines the interaction among design entities and provides the information needed to easily perceive why, how, where, and at what level system actions occur.

Section 6.5 describes system interfaces. There is currently only one automated external interface. The human-machine interface is also addressed in this section. The internal interface design is expressed through a menu structure.

Section 6.6 describes the internal details and attributes of design entities. The chapter provides the developers with specific programming instructions.

6.2 DECOMPOSITION DESCRIPTION

The AMS consists of several design entities. This chapter describes the purpose and function of each design entity. The intent of this chapter is to provide a high-level understanding of the system.

The AMS will replace the existing CTS.

6.2.1 Module Decomposition

The AMS functional areas have been separated into individual design entities which are structurally and functionally distinct. Design entities result from the decomposition of system requirements. The objective of functional decomposition is to divide the system into separate components that can be considered, implemented, changed, and tested with minimal effect on the other design entities.

6.2.1.1 AMS Processing Modules. The AMS processing modules encompass the documenting and tracking of any work-related activities the user desires. The processing modules are called the following:

- User Login
- Main Menu
- Create Activity
- Data Entry
- Memo Editor
- Validate Activities
- Retrieve Activities
- Edit Activities
- Clone Activities

- Report and View Activities
- Add Viewers
- Close Activities
- Reopen Activities
- List Management
- Help System
- Administrative Functions.

6.2.1.1.1 User Login. The user login screen will provide access to the AMS. Users will enter their login ID and password. If these do not match, the user will be given an opportunity to try again. If the user is still unsuccessful after three attempts, the program is terminated. Upon successful login, the module will search for and display the number of unvalidated activities to which the user has been assigned as agent. It will then call the Main Menu module.

6.2.1.1.2 Main Menu. The Main Menu screen acts as the gateway to the other screens which have been identified as design entities. Pull-down menus will be used to list the various modules available for the user to choose. The modules available from the Main Menu are as follows:

- Create Activity
- Validate Activities
- Retrieve Activities
- Help System
- List Management
- Administrative Functions.

6.2.1.1.3 Create Activity. The Create Activity module contains the functions needed to create an activity. An activity ID is generated by the system and Data Entry and Memo Editor modules are used to input the activity data. If an activity agent is identified, a cc:Mail message is generated by the system to notify the agent of the activity. The Clone Activities module can be called by the user to copy the new activity and create an additional new activity. The Add Viewers module can be called by the user to assign viewers for the activity.

6.2.1.1.4 Data Entry. The Data Entry screen is used by the user to enter or modify activity data. The user enters data by the keyboard and uses the keyboard or the mouse to navigate around the screen.

6.2.1.1.5 Memo Editor. The Memo Editor module is used to enter or modify activity data that resides in memo fields. There are three fields in the AMS that are memo fields: activity description, activity status, and activity deliverable. The user enters data by the keyboard and uses the keyboard or the mouse to navigate around the screen. The Memo Editor module also includes basic editing functions such as search and replace.

6.2.1.1.6 Validate Activities. The Validate Activities module is used by a user to accept or refuse activities that have been created with that user as the agent. The Data Entry screen is used to display the activity in read-only mode. The memo editor is used to display the activity description and activity deliverable memo fields in read-only mode. The memo editor is used to edit the activity status. The user can either accept or refuse the

activity. If the activity is refused, a cc:Mail message is generated by the system to notify the agent of the activity refusal. If the activity is accepted, the val status field is updated.

6.2.1.1.7 Retrieve Activities. The Retrieve Activities module is used by the user to retrieve activities and perform some operation upon them. The user will choose the query to run, enter query criteria, run the query, and then the activities retrieved by the query are displayed. The user then will select the activities to be operated upon. The following modules can be called at this point: Edit Activities, Clone Activities, Report and View Activities, Add Viewers, Close Activities, and Reopen Activities.

6.2.1.1.8 Edit Activities. The Edit Activities module is used by the user to edit activities that have been retrieved by a query and then selected. The Data Entry and Memo Editor modules are used to perform the changes to the activities. The Add Viewers module can be called by the user to assign viewers for the activity.

6.2.1.1.9 Clone Activities. The Clone Activities module is used by the user to create activities by copying an existing activity to create a new activity. This module is called from the Retrieve Activities module and operates upon selected activities. The Data Entry and Memo Editor modules are called to perform any changes the user desires. The Add Viewers module can be called by the user to assign viewers for the activity.

6.2.1.1.10 Report and View Activities. The Report and View Activities module contains a dialogue box to display all the possible options. The Data Entry and Memo Editor modules are used to view activities in read-only mode.

6.2.1.1.11 Add Viewers. The Add Viewers module is used by the originator of an activity to allow other AMS users to view the activity in read-only mode. The originator will pick viewers from lists of AMS users that they or the administrator created.

6.2.1.1.12 Close Activities. The Close Activities module is used to close activities. Once an activity is closed, it remains in the system for 2 weeks, after which it can be removed by the Administrative Purge function.

6.2.1.1.13 Reopen Activities. The Reopen Activities module is used to reopen activities that have been closed. An activity cannot be reopened once it has been removed from the system by the Administrative Purge function.

6.2.1.1.14 List Management. The List Management module is used by the user to create and maintain lists. There are three different kinds of lists:

- Lists of AMS users
- Lists of activities or activity groups
- Lists of cc:Mail users.

6.2.1.1.15 Help System. The Help System module is used to display information about the AMS to the user. Text extracted from WHC-SD-HWV-UM-004, *Activity Management System User Reference Manual* (Gates 1994), will be included in the Help System.

6.2.1.1.16 Administration Functions. The Administration Functions module provides screens to maintain the AMS operating environment. The module consists of a variety of administrative tools.

6.2.2 Concurrent Process Decomposition

The processing necessary for each design entity to perform its function is included in each detailed design description.

6.2.3 Data Entity Decomposition

The records and tables comprising the AMS database will be implemented in relational table structure using Paradox products. The individual data design entities are defined in detail in Table 6-1.

6.3 DEPENDENCY DESCRIPTION

The section defines the interaction among design entities and provides the information needed to easily perceive why, how, where, and at what level system actions occur. The dependency description will be expressed through models. The set of models presented are data flow diagrams (DFD) of the AMS functional processes. Subsequent models define the relationship between the entities, physical data structure, and network descriptions.

6.3.1 DFDs

The DFDs present a hierarchical view of the processes, data flows, data stores, and external interfaces supported by the AMS application system. Detailed DFDs are found in Section 6.7.

The DFDs describe data flow between process entities. Data flows are represented by arrowed lines with the name of the data entity. Rectangles on the diagrams represent "process bubbles." A process takes inputs, performs any required transformation, and produces outputs. A process can be an administrative procedure or part of a computer program.

Open-ended oblongs represent data stores. A data store can be implemented as a computer file or paper file.

Both processes and data stores contain reference numbers. The circular graphics are entities which are sources or destinations of data. The DFDs are hierarchical; each level in the diagram describes more detail about each process.

6.3.1.1 Context Diagram (Figure 6-1). The context diagram for AMS presents a high-level view of the interfaces of users and system administrator. The interfaces to the systems administration activities of AMS are as follows.

Table 6-1. Data Design Entities. (sheet 1 of 2)

Field name	Definition
actagentid	User ID of activity agent
actdelivtxt	Deliverable of the activity
actdesc	Description of activity
actdocid	Document ID of document associated with activity
actid	Unique identifier of activity
actnum	Next available activity identifier
actorigid	User ID of activity originator
actprioritycd	Priority of activity
actstatustxt	Status of activity
acttitle	Title of activity
closedt	Date activity is closed
completedt	Date activity is completed
createdt	Date activity is created
currduedt	Current due date of activity
groupid	Unique identifier of activity group
grouptitle	Title of activity group
helpid	Unique identifier for help topic
mailid	Unique identifier for mail list
mailloc	Location of cc:Mail user
mailname	Name of cc:Mail user
mlistid	Unique identifier for mail list
mlisttitle	Title of mail list
origduedt	Original due date of activity
statusdt	Date the activity status was last modified
text	Text of help topic
title	Title of help topic
useractiveflg	Flag used to determine if the user is currently using the system
userid	Unique identifier for user
userloc	Location of user

Table 6-1. Data Design Entities. (sheet 2 of 2)

Field name	Definition
userlogindt	Date user last logged in to AMS
username	Name of user
userorgcode	Organization code of user
userpasswdtxt	Password for user
userrolecd	Determines whether the user is an administrative user
userstatuscd	Used to disable user from using AMS
valst	Determines the validation status of activity
vlistid	Unique identifier for user list
vlistuserid	User ID of user in a user list
vtile	Title of user list

AMS = Activity Management System
ID = Identification

AMS Users. The AMS users are those persons authorized access to AMS who perform the day-to-day maintenance of the database data, request reports and queries, and perform the functional activities supported by the system.

Activity Agent. The Activity Agent is the AMS user who is designated as the agent of an activity. An agent is the person who actually performs the activity. The agent must validate an activity before it becomes a valid activity. Validation is the process of either accepting or refusing an activity. If the activity is accepted, it becomes a valid activity and can be viewed by any assigned viewers. If the activity is refused, then the agent's name is removed from the activity.

AMS Admin User. The AMS Admin User is an AMS user who is designated as an administrative user. The administrative user will have access to special functions to perform system maintenance and other necessary operations.

Activity Management System. The AMS represents all the data stores and processes that make up the AMS.

6.3.1.2 Level 1 AMS Diagram (Figure 6-2). The Level 1 data flow explodes the context diagram into individual processes. The individual processes of this level are as follows.

AMS Login. Before accessing the AMS, the potential user must successfully log onto the AMS. If the logon procedure succeeds, the system then looks for and displays the number of unvalidated activities on which the user is the agent.

Main Menu. After successful logon, the user is presented with the main menu screen. This screen contains menus with functions that the user can select.

6.3.1.3 Level 2 Main Menu Diagram (Figure 6-3). The Level 2 data flow explodes the Main Menu process into its individual subordinate processes. The individual processes of this level are as follows.

Create Activity. This process creates new activities. The user will key all relevant information about the activity through a data entry screen. After an activity is created, additional users can be granted view access and the activity can be copied or cloned. When agents are identified for an activity, they are automatically sent a cc:Mail message notifying them of the new activity.

Help System. Displays help information. The user chooses the desired help topic from a list of help topics and the information about that topic is displayed.

Retrieve Activity. Activities are retrieved by a data query and then can be operated by various functions.

List Management. Activity groups, viewer lists, and mail lists are created and maintained by this process.

Validate Activity. Users can create reports of, or validate, any unvalidated activities on which they have been assigned as the agent. The agent-user validates an activity by either accepting or refusing it. The status of the activity also can be updated when the user validates it. If the user refuses an activity, the originator of the activity is notified automatically by cc:Mail.

Administration Functions. Users who are also administrative users can run the set of functions that perform system maintenance and other necessary operations.

6.3.1.4 Level 2.1 Create Activity Diagram (Figure 6-4). The Level 2.1 data flow explodes the Create Activity process into individual processes. The individual processes of this level are as follows.

Generate Activity ID. This process obtains a unique activity identifier from the System table for the new activity.

Activity Data Entry. The user enters activity information through data entry screens.

Add Viewer to Activity. Adds viewers to an activity by selecting from among the set of viewing lists created by the user or global viewing lists created by the AMS administrator.

Notify Agent of New Activity. Notifies the activity agent of the activity through a cc:Mail message.

Clone Activity. Creates a new activity by copying the current activity.

6.3.1.5 Level 2.1.2 Data Entry Diagram (Figure 6-5). The Level 2.1.2 data flow explodes the Data Entry process into individual processes. The individual processes of this level are as follows.

Data Entry Screen. User enters activity information through a data entry screen.

Find Agent. User selects agent by selecting a name from a list of AMS users. The userid of the agent is then inserted into the Actagentid field. The activity can also not have an agent by selecting "Clear agent" from the menu. If no agent is selected, the Actagentid field is set to a userid of "000000" which corresponds to a user titled "No Agent" in the user table. Only the originator of an activity can change or assign an agent to an activity.

Memo Editor. The three memo fields Actdesc, Actstatustxt, and Actdelivtxt are updated using the Paradox memo editor. The memo editor is essentially a full-screen text editor. It has primitive operands that support functions such as text searching.

6.3.1.6 Level 2.3 Retrieve Activity Diagram (Figure 6-6). The Level 2.3 data flow explodes the Retrieve Activity process into individual processes. The individual processes of this level are as follows.

Query for Activities. This retrieves activities by using a data query employing various search criteria the user selects and inputs.

List Retrieved Activities. A list of activities retrieved by the query is displayed. The list of activities can be sorted by the user into different orders. The user selects the activities on which operations will be performed.

6.3.1.7 Level 2.3.2 List Retrieved Activities Diagram (Figure 6-7). The Level 2.3.2 data flow explodes the List Retrieved Activities process into individual processes. The individual processes of this level are as follows.

Activity Function Menu. Displays the operations the user is allowed to perform on the retrieved activities. When an operation is selected, it passes the operation a list of the selected operations.

Report and View Activities. Allows the user to view or run reports on the selected activities.

Edit Activity. Lets the user edit the selected activities. Only the originator or agent of an activity can edit it.

Clone Activities. Creates new activities by copying existing activities.

Close Activities. Closes the selected activities for which the user is the originator.

Reopen Activities. Reopens selected activities for which the user is the originator.

Add Viewer to Activity. Adds viewers to selected activities for which the user is the originator.

Group Activities. Adds selected activities to an existing activity group the user has previously created.

6.3.1.8 Level 2.3.2.3 Edit Activity Diagram (Figure 6-8). The Level 2.3.2.3 data flow explodes the Edit Activities process into individual processes. The individual processes of this level are as follows.

Activity Data Entry. The user enters activity information through data entry screens.

Add Viewer to Activity. Adds viewers to an activity by selecting personal viewing lists that the user previously created or global viewing lists the administrator created.

6.3.1.9 Level 2.3.2.4 Clone Activities Diagram (Figure 6-9). The Level 2.3.2.4 data flow explodes the Clone Activities process into individual processes. The individual processes of this level are as follows.

Generate Activity ID. This process obtains a unique activity identifier from the System table for the new activity.

Activity Data Entry. The user enters activity information through data entry screens.

Add Viewer to Activity. Adds viewers to an activity by selecting personal viewing lists that the user created or global viewing lists the administrator created.

Notify Agent of New Activity. Notifies the activity agent of the activity assignment through a cc:Mail message.

6.3.1.10 Level 2.4 List Management Diagram (Figure 6-10). The Level 2.4 data flow explodes the List Management process into individual processes. The individual processes of this level are as follows.

User List Management. Create and maintain viewer lists.

Activity Group Management. Create and maintain activity groups.

Mail List Management. Create and maintain mail lists.

6.3.1.11 Level 2.5 Validate Activity Diagram (Figure 6-11). The Level 2.5 data flow explodes the Validate Activity process into individual processes. The individual processes of this level are as follows.

Unvalidated Activity Report. Runs report on unvalidated activities for which the user is agent.

Perform Activity Validation. User accepts or refuses activity.

Memo Editor. User can edit the activity status.

Notify Originator of Refusal. If activity is refused, a cc:Mail message is automatically sent to the originator of the activity.

6.3.1.12 Level 2.6 Administration Functions Diagram (Figure 6-12). The Level 2.6 data flow explodes the Administration Functions process into individual processes. The individual processes of this level are as follows.

Admin User Functions. Functions that affect user data.

Admin Activity Functions. Functions that affect activity data.

Admin Misc Functions. Miscellaneous functions.

6.3.1.13 Level 2.6.1 Administrator User Functions Diagram (Figure 6-13). The Level 2.6.1 data flow explodes the Administration User Functions process into individual processes. The individual processes of this level are as follows.

Add User. Adds a user to the AMS. Administrator selects the user name from the names in cc:Mail. Information not available from cc:Mail is then entered through the keyboard.

Delete User. Marks a user for deletion from the AMS and terminates user's ability to login to the AMS. A user must not be the originator or agent of any activities that are not closed. Once marked, the user record is actually deleted by the purge function.

Disable User. Disables a user from using the AMS and from being designated as an agent of any new activities.

Enable User. Enables a disabled user to use the AMS and be designated as an agent of any new activities. Also unmarks any user marked for deletion who has not already been removed by the purge function.

Edit User. Changes the existing information about the user. The user's userid cannot be changed by this function.

Change User ID. Changes a user's userid.

List Users Online. Lists the users who are currently logged onto the system.

6.3.1.14 Level 2.6.2 Administration Activity Functions Diagram (Figure 6-14). The Level 2.6.2 data flow explodes the Administration Activity Functions process into individual processes. The individual processes of this level are as follows.

Unvalidated Report. Runs a report on all the unvalidated activities in the AMS. The user enters the minimum number of weeks an activity has to have been unvalidated to appear on the report.

Transfer Activities. Transfers a user's activities to another user. If any of the activities are unvalidated, a cc:Mail message is sent for each unvalidated activity notifying the user of the new activity.

Purge Activities. Moves activities from the Activity table into the AMS1993 table. The user enters the minimum number of weeks an activity has to have been closed to be purged. Then any users who have been marked for deletion, and who are not the originator or agent of any activities in the Activity table, are deleted.

6.3.1.15 Level 2.6.3 Administration Miscellaneous Functions Diagram (Figure 6-15). The Level 2.6.3 data flow explodes the Administration Miscellaneous Functions process into individual processes. The individual processes of this level are as follows.

Org Change. This changes all occurrences of one organization code in the User table to another organization code.

Update cc:Mail List. Runs a C program with Paradox Engine calls that updates the AMS cc:Mail table from the HLAN cc:Mail file. This is necessary to be able to add new users to the AMS who have just been added to cc:Mail.

Global User Lists. Lets the administrator create and maintain viewing lists that any user can use when adding viewers to an activity.

6.3.2 Data Dependencies

Computer databases are made up of tables containing associated groups of data. The association of data is made at two basic levels, "entity" and "database."

Individual data items (or data elements) that support a common function or activity are normally stored together. This grouping is commonly known as a record or table. In its early design stage, these groups define what is known as an "entity."

Groups of related entities are brought together to form a "database." A database may be made up of data about a single entity or many entities linked together.

The linking of multiple entities is accomplished by defining the relationship between entities. The entity-relationship diagram is the resulting description of the database's entities and relationships.

6.3.2.1 Entity-Relationship Model. The AMS entity-relationship model consists of a series of rectangular boxes (representing entities) and lines (representing the relationship between two entities) (see Figure 6-16).

6.3.3 Entity Descriptions

6.3.3.1 ACTIVITY Table. The purpose of this table is to store information about an activity. To create an activity, the user as a minimum must input a title and a current due date. These fields are mandatory and cannot be blank. Once an activity, is accepted the original due date cannot be changed.

The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Actid	N*	Key field for this table. Generated by the system.
Actorigid	A10	Foreign key for User table. Identifies originator of activity. Generated by the system.
Actagentid	A10	Foreign key for user table. Identifies agent of activity.
Acttitle	A30	Title of activity. Mandatory field.
Actprioritycd	A3	Priority of activity.
Actdocid	A20	ID of document associated with activity.
Actdesc	M30	Description of activity.
Actstatustxt	M30	Status of activity.
Actdelivtxt	M30	Deliverable of activity.
Origduedt	D	Original due date of activity.
Currduedt	D	Current due date of activity. Mandatory field.
Completedt	D	Completion date of activity.
Closedt	D	Date activity closed. Generated by the system.
Createdt	D	Date activity created. Generated by the system.
Statusdt	D	Date Actstatustxt field was last updated. Generated by the system.
Valst	A1	Validation status of activity. Generated by the system. It can only contain the following values: "a" - Activity accepted. "w" - Activity waiting validation. "r" - Activity rejected. "c" - Activity closed. "n" - Activity not assigned an agent.
Actcatcd	A16	Foreign key for Catlist table. Identifies category code of activity.

ActPrereqcd A3 Foreign key for Prelist table.
Identifies prerequisite code of
activity.

Secondary indices for table ACTIVITY are:

- . Valst
- . Actorigid
- . Actagentid

6.3.3.2 ACTVIEW Table. The purpose of this table is to identify users who are authorized to view an activity by referencing an activity and a viewer list. The Actid field identifies the activity to which to add viewers. The Userid and Vlistid field identify the viewing list containing the users with viewing access.

The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Actid	N*	Part of compound key for this table. Foreign key to Activity table. System generated.
Userid	A10*	Part of compound key for this table. Foreign key to Viewer table. System generated.
Vlistid	N*	Part of compound key for this table Foreign key to Viewer table. System generated.

6.3.3.3 AMS1993 Table. The purpose of this table is to archive activities. When an activity is purged, information about the activity is copied to this table and the activity is deleted from the Activity table. The name of the table will change every year. In 1994, the table will be named AMS1994 and so on. Besides the userid of the originator and the agent, their names are also stored for easy identification in case the user has been deleted from the user table. An activity can be put back into the system from this table by an administration function.

The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Actid	N*	Key field for this table. Generated by the system.
Actorigid	A10	Identifies originator of activity.
Actagentid	A10	Identifies agent of activity.
Acttitle	A30	Title of activity.

Actprioritycd	A3	Priority of activity.
Actdocid	A20	ID of document associated with activity.
Actdesc	M30	Description of activity.
Actstatustxt	M30	Status of activity.
Actdelivtxt	M30	Deliverable of activity.
Origduedt	D	Original due date of activity.
Currduedt	D	Current due date of activity.
Completedt	D	Completion date of activity.
Closedt	D	Date activity closed.
Createdt	D	Date activity created.
Statusdt	D	Date Actstatustxt field was last updated.
Valst	A1	Validation status of activity.
Actoriname	A30	User name of activity originator.
Actagentname	A30	Agent name of activity originator.

6.3.3.4 CATLIST Table. The purpose of this table is to store all the activity categories. This table is maintained by the AMS Administrator by an administrative function.

The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Actcatcd	A16	Key field for this table. Unique identifier for each category.
ActcatDesc	A49	Description of category.

6.3.3.5 CCMAIL Table. The purpose of this table is to identify all the users of cc:Mail. This table is created and updated by an administrative function and uses a C program with Paradox Engine calls that parses through the file c:\ccmail\lancal.dir and extracts the necessary information and enters it into the table. This table is used to enter users into the system; users cannot be entered into the system unless their name is in this table. This table also is used to address cc:Mail messages.

The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Mailid	N*	Key field for this table. System generated.
Mailname	A30	Name of cc:Mail user.
Mailloc	A20	Location of cc:Mail user.

6.3.3.6 GROUPS Table. The purpose of this table is to store activity groups that the user creates and maintains. An activity group is a list of activities. This table just creates the activity group; Grplist table actually populates the activity group. An activity group can be empty.

The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Userid	A10*	Part of compound key. Userid of user who creates the group. System generated.
Groupid	N*	Part of compound key. Unique identifier for activity group. System generated.
Grouptitle	A15	Title of activity group.

6.3.3.7 GRPLIST Table. The purpose of this table is to populate activity groups that the user has created. The Userid and Groupid fields identify the activity group and the Actid field identifies the activity in the group.

The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Userid	A10*	Part of compound key. Userid of user who creates the group. System generated.
Groupid	N*	Part of compound key. Unique identifier for activity group. System generated.
Actid	N*	Activity ID of activity.

6.3.3.8 HELP Table. The purpose of this table is to store help information. Excerpts from WHC-SD-HWV-UM-004, *Activity Management System User Reference Manual* (Gates 1994), will be inserted into this table.

The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Helpid	N*	Key field for the table.
Title	A20	Title of help topic.
Text	M20	Text of help topic.

6.3.3.9 MAIL Table. The purpose of this table is store mail lists that the user creates and maintains. A mail list is a list of cc:Mail users. This table just creates the mail list; the Maillist table actually populates the mail list. A mail list can be empty.

The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Userid	A10*	Part of compound key. Userid of user who creates the mail list. System generated.
Mlistid	N*	Part of compound key. Unique identifier for mail list. System generated.
Mlisttitle	A15	Title of mail list.

6.3.3.10 MAILLIST Table. The purpose of this table is to populate mail lists that the user has created. The Userid and Mlistid fields identify the mail list and the mailname contains the name of the cc:Mail user.

The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Userid	A10*	Part of compound key. Userid of user who creates the mail list. System generated.
Mlistid	N*	Part of compound key. Unique identifier for mail list. System generated.
Mailname	A30*	Name of cc:Mail user.

6.3.3.11 PRELIST Table. The purpose of this table is to store all the activity Prerequisites. This table is maintained by the AMS Administrator by an administrative function. The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Actprecd	A3	Key field for this table. Unique identifier for each prerequisite.
Actpredesc	A20	Description of prerequisite.

6.3.3.12 SYSTEM Table. The purpose of this table is to store the next available unique activity identifier. When an activity is created, a full lock is obtained for this table, and the value of the Actnum field is read and used for the new activity. The value of the Actnum field is then incremented and the full lock is released.

The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Actnum	N*	Key field for table. Value of next available activity identifier.

6.3.3.13 USER Table. The purpose of this table is to store information about AMS users. This information is created when a user is added to the AMS. The Username and Userloc fields are copied from the cc:Mail table when the user is added to the AMS.

The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Userid	A10*	Key field for table. Unique identifier for user. This consists of a company identifier and the user's payroll number.
Username	A30	Name of user.
Userorgcode	A6	Organization code of user.
Userloc	A20	Location of user.
Userpasswdtxt	A10	Password for user.
Userlogindt	D	Date user last logged into the AMS.
Userrolecd	A1	Role user plays in the AMS. Used to identify administrative and regular users.
Userstatuscd	A1	Status of user. Used to disable user from logging into the AMS and being identified as an agent for an activity.
Useractiveflg	A1	Flag to determine if the user is currently logged into the AMS.

6.3.3.14 VIEWER Table. The purpose of this table is to store viewer lists. Viewer lists are used to identify users with view access to activities and to specify the activity originator and/or agent for one of the queries. A viewer list is a list of AMS users. This table only creates the viewing list; the Viewlist table actually populates the viewing list. A viewer list can be empty.

The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Userid	A10*	Part of compound key. Userid of user who creates the viewer list.
Vlistid	N*	Part of compound key. Unique identifier for viewing list.
Vtitle	A30	Title of viewing list.

6.3.3.15 VIEWLIST Table. The purpose of this table is to populate viewing lists that the user has created. The userid and vlistid fields identify the viewing list and the vlistuserid field identifies the user in the list.

The structure of this table is as follows:

<u>Field name</u>	<u>Field type</u>	<u>Description</u>
Userid	A10*	Part of compound key. Userid of user who creates the viewer list.
Vlistid	N*	Part of compound key. Unique identifier for viewing list.
Vlistuserid	A10*	Userid of user in viewing list.

6.3.4 Conversion Dependencies

Because the customer does not wish to convert any data from the existing system there are no conversion dependencies.

6.4 INTERFACE DESCRIPTION

This section describes the interfaces between AMS and those persons who have need to AMS data. Interfaces are divided into external, internal, and human/machine.

6.4.1 External Interfaces

To generate cc:Mail messages, the AMS will use the cc:Mail Import System (CCIS). This system monitors a directory on the file server and when a file

in a special format is written to that area, CCIS will parse through the file and generate a cc:Mail message. For further information, contact the cc:Mail product manager.

To obtain a list of cc:Mail users, the AMS will use a file that is currently part of the LANCAL program. This file, s:\ccmail\lancal.dir, is parsed through by a C program with Paradox Engine calls to populate the cc:Mail table. Because the format of this file can change, this program will need to be modified as necessary. In the case that this file becomes no longer available, alternative means may need to be developed to obtain this information.

6.4.2 Internal Interfaces

The Paradox relational database management system provides the management of the AMS data. Version 4.0 of PAL is the principal programming language.

C is used to obtain cc:Mail data.

6.4.3 Human/Machine Interfaces

The interface between the human operator and the AMS is critical to effective use of the system. This section describes the nature of that interface.

6.4.3.1 Developer Interface. The AMS development staff consists of software developers whose responsibility is to define, create, implement, and maintain the PAL application modules that materialize the user's requirements in a computer application. The nature of these responsibilities requires that developers have a high-level access authority to the AMS during system development.

When changes to the AMS are authorized by the AMS CCB, developers must have the authority to update the database structure, perform global data manipulation, maintain the data dictionary, modify the application software, maintain and access a test database, and access the production database.

6.4.3.2 User Interface. The end users of the AMS are authorized access to specific portions of the database. This access is accomplished by means of a series of on-line transactions and reports.

6.4.3.3 Administrative User Interface. The administrative users of the AMS are authorized to perform any administrative functions necessary to the maintenance and running of the AMS. This interface is accomplished by means of a series of on-line transaction screens and reports.

6.5 DETAILED DESIGN

This section documents the internal details of the programming modules contained in the application procedure library code for the AMS.

6.5.1 Procedure Descriptions

This section will discuss each programming module contained in the AMS. The AMS is started by executing the batch file runams.bat. This batch file starts Paradox running and then executes the PAL script ams.sc. The ams.sc script makes the network connections necessary to run the AMS, changes the working directory to the network drive containing the application, and starts calling procedures from the procedure library ams_lib.

Script: AMS.SC--Makes two network connections and starts the application by calling the login and top_menu procedures

Procedure: ADD_USER_DLG_PROC--Controls the dialogue box used in the add_user procedure

Procedure: ADD_USER--Adds a new user to the AMS

Procedure: AMS_ERROR--Displays a dialogue box containing an error message and error code that are parameters for the procedure

Procedure: AMS_LOGIN--Verifies that a user is a valid user before allowing use of the AMS and sets the appropriate database locks needed to limit access to the data

Procedure: ATTACH_LIST3--Attaches a viewing list for a group of activities

Procedure: ATTACH_LIST--Attaches a viewing list for an activity

Procedure: BAD_QUERY--Notifies the users that the search criteria they entered for the retrieve by match query is not valid

Procedure: BLANK_ACTIVITY--If users create or clone an activity and try to save it without a title or original due date, this procedure is called to display a warning to the users and let them either delete the activity or continue inputting activity data.

Procedure: BLANK_DATE--Called when a user edits an activity and tries to save the activity without a current or original due date. A dialogue box tells the user that a current or original due date must be entered and returns to the edit data entry screen.

Procedure: BLANK_TITLE--Called when a user edits an activity and tries to save the activity without a title. A dialogue box tells the user that a title must be entered and returns to the edit data entry screen.

Procedure: BRIEF1_QUERY--Runs a query that retrieves the activity information necessary for the Tandscape brief report

Procedure: BRIEF2_QUERY--Runs a query that retrieves the activity information necessary for the portrait brief report

Procedure: CATLOOK_DLG_PROD--Controls the dialogue box used in the catlook procedure

Procedure: CATLOOK--Lets the user select a category from a list of predefined categories

Procedure: CCMail_DLG_PROC--Controls the dialogue box that the user uses to address a cc:Mail message that contains an AMS report

Procedure: CHANGE_DEFAULTS--Displays a dialogue box telling the user that the selected function is not implemented yet

Procedure: CHANGE_PASSWORD_DLG_PROC--Controls the dialogue box used to change the user's password

Procedure: CHANGE_PASSWORD--Displays the dialogue box to change the user's password and then changes the password

Procedure: CHANGE_UID_DLG_PROC--Controls the dialogue box that the AMS administrator uses to change a user's ID number

Procedure: CHANGE_UID--Changes a user's ID by using a dialogue box for data entry and changing the user ID

Procedure: CHOOSE_PRINTER--Lets the user select which printer port to use to print a report

Procedure: CLONE_ACTIVITY--Creates new activities by copying existing activities

Procedure: CLOSE_ACTIVITY--Closes activities. If an activity to be closed has a validation status of "w," the activity agent field is cleared.

Procedure: DEL_LIST_CONFIRM--Asks the user to confirm that the selected list should really be deleted

Procedure: DEL_USER--Marks a user for deletion. The user actually is deleted by the purge procedure.

Procedure: DETACH_LIST3--Detaches a viewing list for a group of activities

Procedure: DETACH_LIST--Detaches a viewing list for an activity

Procedure: DISABLE_USER--Prevents a user from using the AMS and being designated an agent of a new activity

Procedure: EDIT_ACTIVITIES--Lets the user edit activities

Procedure: EDIT_ACT_WAIT_PROC--Controls data entry when the user is creating, cloning, or editing activities

Procedure: EDIT_CAT_DLG_PROC--Controls the dialogue box used in the edit_cat procedure

Procedure: EDIT_CAT--Used by the administrator to maintain the predefined list of categories

Procedure: EDIT_MEMO_MENU--Displays the memo editor menu

Procedure: EDIT_MENU--Displays the menu used for creating, cloning, or editing activities

Procedure: EDIT_PRE_DLG_PROC--Controls the dialogue box used in the edit_pre procedure

Procedure: EDIT_PRE--Used by the administrator to maintain the predefined list of prerequisites

Procedure: EDIT_USER_DLG_PROC--Controls the dialogue box used for editing a user's information

Procedure: EDIT_USER--Lets the administrator edit a user's information

Procedure: ENABLE_USER--Enables a disabled user or unmarks a user marked for deletion

Procedure: ENTER_ACTIVITY--Creates a new activity

Procedure: ENTER_PARADOX--Lets an administrative user who has a copy of Paradox enter the Paradox interactive environment from AMS

Procedure: ERROR_HANDLER--Handles any internal Paradox errors that may occur

Procedure: EXIT_CONFIRM--Asks the user for confirmation before exiting the list management functions

Procedure: FINDCATNAME--Returns the description of a category that corresponds to a category code

Procedure: FINDPRENAME--Returns the description of a prerequisite that corresponds to a prerequisite code

Procedure: FULL_QUERY--Performs a query that retrieves all information necessary for the full report

Procedure: GET_ACTIVITY--Retrieves the next or previous activity when the user is editing activities

Procedure: GET_FILENAME--Prompts the user for a filename when one is needed for reading or writing

Procedure: GET_FILE_DLG_PROC--Controls the dialogue box used to prompt the user for a filename

Procedure: GET_TASK_ID--Generates a new activity ID number

Procedure: GET_UNVAL--Moves to the next or previous activity during activity validation

Procedure: GLOBAL_LIST--Lets the administrative user create and maintain global viewing lists

Procedure: GROUP_ACTIVITY--Adds selected activities to an existing activity group

Procedure: GROUP_LIST_DLG_PROC--Controls the dialogue box that is used to select the activity group to which selected activities will be added

Procedure: GROUP_LIST--Lets the user create and maintain activity groups

Procedure: HELP_SYSTEM_DLG_PROC--Controls the dialogue box used to display the help topics

Procedure: HELP_SYSTEM_WAIT_PROC--Controls the data entry screen used to display the help topic text

Procedure: HELP_SYSTEM--Displays help information

Procedure: HIDE_TEXT--Encrypts and decrypts a text string

Procedure: HIGHLIGHT_ROW--Highlights an activity in the retrieve_action procedure

Procedure: LIST_ACTIVITY2--Lets a user attach or detach viewing lists for an activity after creating, cloning, or editing

Procedure: LIST_ACTIVITY3_WAIT_PROC--Controls the data entry screen used to attach or detach viewing lists for a group of activities

Procedure: LIST_ACTIVITY3--Lets a user attach or detach viewing lists for a group of activities

Procedure: LIST_ACTIVITY_WAIT_PROC--Controls the data entry screen used to attach or detach viewing lists for an activity

Procedure: LIST_ACTIVITY--Lets a user attach or detach viewing lists for an activity

Procedure: LIST_USERS--Displays a list of users currently using the AMS

Procedure: LOGIN_DLG_PROC--Controls the dialogue box used to log into the AMS

Procedure: LOW_MEM_ERROR--Displays an error message to the user when the AMS runs out of memory and cannot complete an operation

Procedure: MAIL_LIST_DLG_PROC--Controls the dialogue box used to create and maintain mail lists

Procedure: MAIL_LIST--Lets the user create and maintain mail lists

Procedure: MATCH_QUERY--Runs the retrieve-by-match query

Procedure: MEMO_FIND--Locates a text string within the memo editor

Procedure: MEMO_REPLACE--Performs a text search and replace function within the memo editor

Procedure: MEMO_WRITE--Writes the contents of the memo field to a file from within the memo editor

Procedure: NAMELOOK_DLG_PROC--Controls the dialogue box used in the namelook procedure

Procedure: NAMELOOK--Lets the user search for an AMS user's name and returns their userid to AMS

Procedure: NOTIFY_AGENT--Generates a cc:Mail message notifying users that they are the agents of a new activity

Procedure: NOTIFY_ORIGINATOR--Generates a cc:Mail message notifying the originator of an activity that the agent has refused that activity

Procedure: ORG_CHANGE_DLG_PROC--Controls the dialogue box used to globally change an organization code

Procedure: ORG_CHANGE--Globally changes an organization code within the user table

Procedure: PRELOOK_DLG_PROD--Controls the dialogue box used in the prelook procedure

Procedure: PRELOOK--Lets the user select a category from a list of predefined prerequisites

Procedure: PURGE--Moves activities from the activity table to the archive table. It also removes any users who have been marked as deleted.

Procedure: QUERY_OP_HELP--Displays query operators that can be used in the retrieve-by-match query

Procedure: QUIT_AMS--Terminates a user's AMS session

Procedure: REMOVE_ARCHIVE_DLG_PROC--Controls the dialogue box used to copy or delete an archive table

Procedure: REMOVE_ARCHIVE--Lets the administrative user copy or delete an archive table

Procedure: REOPEN_ACTIVITY--Reopens any closed activities

Procedure: RETRIEVE_ACTION--Displays any activities retrieved by a query and lets the user pick an operation to run on any selected activities

Procedure: RETRIEVE_ACT_PROC--Controls the data entry screen used for the retrieve_action procedure

Procedure: RETRIEVE_CAT--Runs the retrieve-by-category query

Procedure: RETRIEVE_COMPLETED--Runs the retrieve completed activities query

Procedure: RETRIEVE_DEL_DLG_PROC--Controls the dialogue box used for the retrieve delinquent activities query

Procedure: RETRIEVE_DEL--Runs the retrieve delinquent activities query

Procedure: RETRIEVE_GROUP--Runs the retrieve by activity group query

Procedure: RETRIEVE_MATCH_DLG_PROC--Controls the dialogue box for the retrieve-by-match query

Procedure: RETRIEVE_MATCH--Runs the retrieve-by-match query

Procedure: RETRIEVE_PRE--Runs the retrieve-by-prerequisite query

Procedure: RETRIEVE_ROLE_DLG_PROC--Controls the dialogue box for the retrieve-by-role query

Procedure: RETRIEVE_ROLE--Runs the retrieve-by-role query

Procedure: RETRIEVE_VAL_DLG_PROC--Controls the dialogue box for the retrieve-by-validation status query

Procedure: RETRIEVE_VAL--Runs the retrieve-by-validation status query

Procedure: RUN_REPORT_WAIT_PROC--Controls the data entry screen for the run_report procedure

Procedure: RUN_REPORT--Lets the user run activity reports or view activities

Procedure: RUN_REP_DLG_PROC--Controls the dialogue box for the run_report procedure

Procedure: RUN_UNVAL_REP--Runs a report on unvalidated activities for administrative users

Procedure: SETUP_REPORT--Sets the destination for an activity report

Procedure: SUM_QUERY--Runs a query that retrieves all the information necessary for the summary report

Procedure: TOP_MENU--Displays the main screen and main menu

Procedure: TRANSFER_ACTIVITIES--Transfers all of one user's activities to another user

Procedure: UNDO_WARNING--Warns the user about the effect of the undo function and asks for confirmation before proceeding

Procedure: UNPURGE_DLG_PROC--Controls the dialogue box used for the unpurge function

Procedure: UNPURGE--Moves activities from the archive table back into the Activity table

Procedure: UPDATE_CCMAIL--Runs a C program, mail_imp.c, that will update the cc:Mail table

Procedure: VALIDATE_ACT--Lets the user validate activities

Procedure: VALIDATE_REP--Runs the full report for any unvalidated activities for which the user is the agent

Procedure: VALIDATE_WAIT_PROC--Controls the data entry screen for the validate_act procedure

Procedure: VAL_LOOKUP--Displays the number of unvalidated activities for which the user is the agent

Procedure: VAL_MENU--Displays the menu for activity validation

Procedure: VIEWER_REP--Runs a report that lists all the users with viewing access for selected activities

Procedure: VIEWING_LIST--Lets the user create and maintain viewing lists

Procedure: VIEW_LIST_DLG_PROC--Controls the dialogue box used to create and maintain viewing lists

Procedure: WARNDUEDT--Displays a warning that the current or original due date is less than the current date.

6.5.2 Procedure Dependencies

Tables 6-2 and 6-3 detail the procedure dependencies for the AMS. Table 6-2 lists each procedure that a procedure calls. Table 6-3 lists every procedure that calls a specific procedure.

Table 6-2. Each Procedure that a Procedure Calls. (sheet 1 of 7)

Procedure name	Calls procedure
Ams Script	ams_error ams_login error_handler top_menu val_lookup
add_user	add_user_dlg_proc ams_error
add_user_dlg_proc	low_mem_error
ams_error	None
ams_login	ams_error login_dgl_proc quit_ams
attach_list	None
attach_list3	None
bad_query	help_system
blank_activity	None
blank_date	None
blank_title	None
brief1_query	None
brief2_query	None
catlook_dlg_proc	None
catlook	catlook_dlg_proc low_mem_error
ccmail_dlg_proc	low_mem_error
change_defaults	None
change_password	ams_error change_password_dlg_proc
change_password_dlg_proc	None
change_uid	change_uid_dlg_proc namelook
change_uid_dlg_proc	None
choose_printer	None

Table 6-2. Each Procedure that a Procedure Calls. (sheet 2 of 7)

Procedure name	Calls procedure
clone_activity	ams_error edit_act_wait_proc edit_menu get_task_id
close_activity	ams_error
del_list_confirm	None
del_user	namelook
detach_list	None
detach_list3	None
disable_user	ams_error namelook
edit_act_wait_proc	blank_activity blank_date blank_title edit_memo_menu edit_menu get_activity get_filename list_activity2 memo_find memo_replace memo_write nameLook notify_agent undo_warning warnduedt
edit_activities	edit_act_wait_proc edit_menu
edit_cat_dlg_proc	None
edit_cat	edit_cat_dlg_proc low_mem_error
edit_memo_menu	None
edit_menu	None
edit_pre_dlg_proc	None
edit_pre	edit_pre_dlg_proc low_mem_error
edit_user	ams_error edit_user_dlg_proc nameLook

Table 6-2. Each Procedure that a Procedure Calls. (sheet 3 of 7)

Procedure name	Calls procedure
edit_user_dlg_proc	None
enable_user	ams_error namelook
enter_activity	ams_error edit_act_wait_proc edit_menu get_task_id
enter_paradox	None
error_handler	None
exit_confirm	None
findcatname	None
findprename	None
full_query	None
get_activity	None
get_file_dlg_proc	None
get_filename	get_file_dlg_proc
get_task_id	ams_error
get_unval	None
global_list	viewing_list
group_activity	low_mem_error
group_list	group_list_dlg_proc low_mem_error
group_list_dlg_proc	ams_error del_list_confirm exit_confirm low_mem_error
help_system	help_system_dlg_proc help_system_wait_proc low_mem_error
help_system_dlg_proc	None
help_system_wait_proc	None
hide_text	None
highlight_row	None
list_activity	list_activity_wait_proc

Table 6-2. Each Procedure that a Procedure Calls. (sheet 4 of 7)

Procedure name	Calls procedure
list_activity2	ams_error list_activity_wait_proc
list_activity3	list_activity3_wait_proc
list_activity3_wait_proc	attach_list3 detach_list3
list_activity_wait_proc	attach_list detach_list
list_users	None
login_dgl_proc	None
low_mem_error	None
mail_list	low mem error mail_list_dlg_proc
mail_list_dlg_proc	ams_error del_list confirm exit_confirm low_mem_error
match_query	bad_query
memo_find	None
memo_replace	None
memo_write	get_filename
namelook	namelook_dlg_proc
namelook_dlg_proc	low_mem_error
notify_agent	None
notify_originator	None
org_change	org_change_dlg_proc
org_change_dlg_proc	None
prelook_dlg_proc	None
prelook	low mem error prelook_dlg_proc
purge	None
query_op_help	None
quit_ams	None

Table 6-2. Each Procedure that a Procedure Calls. (sheet 5 of 7)

Procedure name	Calls procedure
remove_archive	remove_archive_dlg_proc unpurge_dlg_proc
remove_archive_dlg_proc	None
reopen_activity	ams_error
retrieve_act_proc	highlight_row
retrieve_action	clone_activity close_activity edit_activities group_activity list_activity list_activity3 reopen_activity retrieve_act_proc run_report viewer_rep
retrieve_cat	low_mem_error
retrieve_completed	retrieve_action retrieve_role_dlg_proc
retrieve_del	retrieve_action retrieve_del_dlg_proc
retrieve_del_dlg_proc	None
retrieve_group	low_mem_error retrieve_action
retrieve_match	low_mem_error match_query retrieve_action retrieve_match_dlg_prog
retrieve_match_dlg_prog	low_mem_error query_op_help
retrieve_pre	low_mem_error
retrieve_role	retrieve_action retrieve_role_dlg_proc
retrieve_role_dlg_proc	None
retrieve_val	retrieve_action retrieve_val_dlg_proc
retrieve_val_dlg_proc	None
run_rep_dlg_proc	None

Table 6-2. Each Procedure that a Procedure Calls. (sheet 6 of 7)

Procedure name	Calls procedure
run_report	brief1_query brief2_query full_query run_rep_dlg_proc run_report_wait_proc setup_report sum_query
run_report_wait_proc	edit_memo_menu memo_find memo_write setup_report
run_unval_rep	None
setup_report	ccmail_dlg_proc get_filename low_mem_error
sum_query	None
top_menu	add_user change_defaults change_password change_uid del_user disable_user edit_user enable_user enter_activity enter_paradox global_list group_list help_system list_users mail_list org_change purge quit_ams remove_archive retrieve_completed retrieve_del retrieve_group retrieve_match retrieve_role retrieve_val run_unval_rep transfer_activities unpurge update_ccmail

Table 6-2. Each Procedure that a Procedure Calls. (sheet 7 of 7)

Procedure name	Calls procedure
top_menu (cont)	validate_act validate_rep viewing_list
transfer_activities	ams_error namelook notify_agent
undo_warning	None
unpurge	unpurge_dlg_proc
unpurge_dlg_proc	get_task_id low_mem_error namelook
update_ccmail	quit_ams
val_lookup	None
val_menu	None
validate_act	ams_error val_menu validate_wait_proc
validate_rep	choose_printer
validate_wait_proc	edit_memo_menu get_filename get_unval memo_find memo_replace memo_write notify_originator undo_warning val_menu
view_list_dlg_proc	ams_error del_list_confirm exit_confirm low_mem_error
viewer_rep	get_filename
viewing_list	low_mem_error view_list_dlg_proc
warnduedt	None

Table 6-3. Each Procedure that Calls a Specific Procedure.
(sheet 1 of 7)

Procedure name	Is called by procedure
add_user	top_menu
add_user_dlg_proc	add_user
ams_error	Ams Script add_user ams_login change_password clone_activity close_activity disable_user edit_user enable_user enter_activity get_task_id group_list_dlg_proc list_activity2 mail_list_dlg_proc reopen_activity transfer_activities validate_act view_list_dlg_proc
ams_login	Ams Script
attach_list	list_activity_wait_proc
attach_list3	list_activity3_wait_proc
bad_query	match_query
blank_activity	edit_act_wait_proc
blank_date	edit_act_wait_proc
blank_title	edit_act_wait_proc
brief1_query	run_report
brief2_query	run_report
catlook_dlg_proc	catlook
catlook	edit_act_wait_procd
ccmail_dlg_proc	setup_report
change_defaults	top_menu
change_password	top_menu
change_password_dlg_proc	change_password
change_uid	top_menu

Table 6-3. Each Procedure that Calls a Specific Procedure.
(sheet 2 of 7)

Procedure name	Is called by procedure
change_uid_dlg_proc	change_uid
choose_printer	validate_rep
clone_activity	retrieve_action
close_activity	retrieve_action
del_list_confirm	group_list_dlg_proc mail_list_dlg_proc view_list_dlg_proc
del_user	top_menu
detach_list	list_activity_wait_proc
detach_list3	list_activity3_wait_proc
disable_user	top_menu
edit_act_wait_proc	clone_activity edit_activities enter_activity
edit_activities	retrieve_action
edit_cat_dlg_proc	edit_cat
edit_cat	top_menu
edit_memo_menu	edit_act_wait_proc run_report_wait_proc validate_wait_proc
edit_menu	clone_activity edit_act_wait_proc edit_activities enter_activity
edit_pre_dlg_proc	edit_pre
edit_pre	top_menu
edit_user	top_menu
edit_user_dlg_proc	edit_user
enable_user	top_menu
enter_activity	top_menu
enter_paradox	top_menu
error_handler	Ams Script

Table 6-3. Each Procedure that Calls a Specific Procedure.
(sheet 3 of 7)

Procedure name	Is called by procedure
exit_confirm	group_list_dlg_proc mail_list_dlg_proc view_list_dlg_proc
findcatname	clone_activity edit_activities edit_act_wait_proc enter_activity get_activity get_unval validate_act
findprename	clone_activity edit_activities edit_act_wait_proc enter_activity get_activity get_unval validate_actNone
full_query	run_report
get_activity	edit_act_wait_proc
get_file_dlg_proc	get_filename
get_filename	edit_act_wait_proc memo_write setup_report validate_wait_proc viewer_rep
get_task_id	clone_activity enter_activity unpurge_dlg_proc
get_unval	validate_wait_proc
global_list	top_menu
group_activity	retrieve_action
group_list	top_menu
group_list_dlg_proc	group_list
help_system	bad_query top_menu
help_system_dlg_proc	help_system
help_system_wait_proc	help_system

Table 6-3. Each Procedure that Calls a Specific Procedure.
(sheet 4 of 7)

Procedure name	Is called by procedure
hide_text	add_user ams_login change_password change_password_dlg_proc edit_user
highlight_row	retrieve_act_proc
list_activity	retrieve_action
list_activity2	edit_act_wait_proc
list_activity3	retrieve_action
list_activity3_wait_proc	list_activity3
list_activity_wait_proc	list_activity list_activity2
list_users	top_menu
login_dgl_proc	ams_login
low_mem_error	add_user_dlg_proc ccmail_dlg_proc group_activity group_list group_list_dlg_proc help_system mail_list mail_list_dlg_proc nameLook_dlg_proc retrieve_group retrieve_match retrieve_match_dlg_proc setup_report unpurge_dlg_proc view_list_dlg_proc viewing_list
mail_list	top_menu
mail_list_dlg_proc	mail_list
match_query	retrieve_match
memo_find	edit_act_wait_proc run_report_wait_proc validate_wait_proc
memo_replace	edit_act_wait_proc validate_wait_proc

Table 6-3. Each Procedure that Calls a Specific Procedure.
(sheet 5 of 7)

Procedure name	Is called by procedure
memo_write	edit_act_wait_proc run_report_wait_proc validate_wait_proc
namelook	change_uid del_user disable_user edit_act_wait_proc edit_user enable_user transfer_activities unpurge_dlg_proc
namelook_dlg_proc	namelook
notify_agent	edit_act_wait_proc transfer_activities
notify_originator	validate_wait_proc
org_change	top_menu
org_change_dlg_proc	org_change
prelook_dlg_proc	prelook
prelook	edit_act_wait_proc
purge	top_menu
query_op_help	retrieve_match_dlg_prog
quit_ams	ams_login top_menu update_ccmail
remove_archive	top_menu
remove_archive_dlg_proc	remove_archive
reopen_activity	retrieve_action
retrieve_act_proc	retrieve_action
retrieve_action	retrieve_completed retrieve_del retrieve_group retrieve_match retrieve_role retrieve_val
retrieve_cat	top_menu
retrieve_completed	top_menu

Table 6-3. Each Procedure that Calls a Specific Procedure.
(sheet 6 of 7)

Procedure name	Is called by procedure
retrieve_del	top_menu
retrieve_del_dlg_proc	retrieve_del
retrieve_group	top_menu
retrieve_match	top_menu
retrieve_match_dlg_prog	retrieve_match
retrieve_pre	top_menu
retrieve_role	top_menu
retrieve_role_dlg_proc	retrieve_completed retrieve_role
retrieve_val	top_menu
retrieve_val_dlg_proc	retrieve_val
run_rep_dlg_proc	run_report
run_report	retrieve_action
run_report_wait_proc	run_report
run_unval_rep	top_menu
setup_report	run_report run_report_wait_proc
sum_query	run_report
top_menu	Ams Script
transfer_activities	top_menu
undo_warning	edit_act_wait_proc validate_wait_proc
unpurge	top_menu
unpurge_dlg_proc	remove_archive unpurge
update_ccmail	top_menu
val_lookup	Ams Script
val_menu	validate_act validate_wait_proc
validate_act	top_menu
validate_rep	top_menu

Table 6-3. Each Procedure that Calls a Specific Procedure.
(sheet 7 of 7)

Procedure name	Is called by procedure
validate_wait_proc	validate_act
view_list_dlg_proc	viewing_list
viewer_rep	retrieve_action
viewing_list	global_list top_menu
warnduedt	edit_act_wait_proc

6.6 DFDs

Figure 6-1. Activity Management System Context Diagram.

Figure 6-2. Activity Management System Level 1
Activity Management System Data Flow Diagram.

Figure 6-3. Activity Management System Level 2
Main Menu Data Flow Diagram.

**Figure 6-4. Activity Management System Level 2.1
Create Activity Data Flow Diagram.**

Figure 6-5. Level 2.1.2 Activity Data Entry Data Flow Diagram.

Figure 6-6. Level 2.3 Retrieve Activity Data Flow Diagram.

Figure 6-7. Level 2.3.2 List Retrieved Activities Data Flow Diagram.

Figure 6-8. Level 2.3.2.3 Edit Activity Data Flow Diagram.

Figure 6-9. Level 2.3.2.4 Clone Activity Data Flow Diagram.

Figure 6-10. Level 2.4 List Management Data Flow Diagram.

Figure 6-11. Level 2.5 Validate Activity Data Flow Diagram.

Figure 6-12. Level 2.6 Administration Functions Data Flow Diagram.

Figure 6-13. Level 2.6.1 Administration User Functions Data Flow Diagram.

Figure 6-14. Level 2.6.2 Administration Activity Functions Data Flow Diagram.

Figure 6-15. Level 2.6.3 Administration Miscellaneous Functions Data Flow Diagram.

Figure 6-16. Activity Management System Entity-Relationship Diagram.

6.7 SCREEN FLOWS

The screen flows are shown in Figure 6-17.

Figure 6-17. Activity Management System Screen Flow. (sheet 1 of 9)

6-61

Figure 6-17. Activity Management System Screen Flow. (sheet 3 of 9)

Figure 6-17. Activity Management System Screen Flow. (sheet 4 of 9)

Figure 6-17. Activity Management System Screen Flow. (sheet 5 of 9)

Figure 6-17. Activity Management System Screen Flow. (sheet 6 of 9)

Figure 6-17. Activity Management System Screen Flow. (sheet 7 of 9)

Figure 6-17. Activity Management System Screen Flow. (sheet 8 of 9)

Figure 6-17. Activity Management System Screen Flow. (sheet 9 of 9)

7.0 SYSTEM USER DOCUMENTATION

User documentation for the AMS application is contained in WHC-SD-HWV-UM-004, *Activity Management System User Reference Manual* (Gates 1994).

This page intentionally left blank.

8.0 SOFTWARE CONFIGURATION CONTROL

8.1 PURPOSE

This section contains the SCMP for the AMS. It is intended to be used as a controlling document to ensure that the integrity of the AMS is preserved during future development (enhancement), maintenance, and production.

8.2 SCOPE

Configuration management is defined as a discipline applying technical and administrative direction and surveillance to:

- Identify and document the functional and physical characteristics of a configuration item
- Control changes to those characteristics
- Record and report change processing and implementation status
- Verify compliance with specified requirements (IEEE 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology* [IEEE 1990])

This SCMP covers software configuration management (SCM) activities for the AMS as of October 1, 1993.

The SCM activities include the following:

- Identification and establishment of baselines
- Review, approval, and control of changes
- Tracking and reporting of such changes
- Audits and review of the evolving software product
- Control of interface documentation (IEEE 823-1983, *Software Configuration Plans* [IEEE 1983]).

8.3 MANAGEMENT

8.3.1 Organization

There are three types of organizations relative to this SCMP: customer, performing, and supporting.

The customer base is expected to grow as new users are added to the initial HWVP AMS and as new organizations begin using AMS independently. This means that there will be a variety of customers interfacing with the AMS.

The extent of the ultimate customer base is unknown at this time. These customer organizations will be defined in Section 8.6 as they become known. The AMS users (regardless of organization) fall into the following categories (each organization will remain independent, with its own AMS database, server, and software):

- AMS users who originate activities
- AMS users who are classified as agents
- AMS users who are allowed to look at activity data by being classified as "viewers" by the originator
- AMS administrative users who assign users, maintain the database, and execute simple Paradox reports.

The performing organization is Information and Scientific Systems, Environmental Systems Planning, AMS team. This organization is a subdivision of BCSR, IRM, and performs the development and maintenance work on the AMS.

The supporting organizations assist the performing organization in its duties with QA, verifying and validating, data standards, and network management issues.

8.3.2 Responsibility

The responsibilities of the customer organizations are as follows.

- Users--The customer population, through selected representation, is responsible for defining the system requirements and acceptance criteria, participating in testing, and reviewing document revisions and test results.
- Cognizant manager--Customer management is responsible for the approval, the Work Authorization, and funding for the Work Authorization.
- AMS administrator(s)--Each AMS installation will have at least one AMS administrator who is responsible for assigning new users, generating administrative reports, running AMS utilities for archiving old data, and generating administrative and other special reports.

The responsibilities of the performing organization are as follows.

- AMS project manager--The AMS Support project manager is responsible for the design, coding, implementation, testing, and overall quality of the system, including actual software coding tasks as well as input to all deliverable documents. This project manager directs other developers in work assignments and serves as the primary point of contact between the customer, supporting, and performing organizations.

- Environmental System Planning Manager--The Environmental Systems Planning manager is responsible for the approval, budget, personnel, schedule, and resource utilization for the AMS.
- Software engineers--Software engineers are assigned to the AMS project as required to respond to customer needs. Their responsibilities are design, coding, and unit testing of the software and revision of the documentation.

The responsibilities of the supporting organizations are as follows.

- V&V Consultants--The V&V consultant acts as the independent reviewer. This individual is part of a separate organization and is not part of the AMS Project, but is qualified to perform/approve the V&V activities as specified in WHC-CM-3-10, *Software Practices*, SP-3.2, Section 6.7.
- DSA--The DSA group provides communication with the DAC and assists the AMS Project in complying with WHC Data Administration Standards.
- Hanford Area Network Technical Advisory Group--This group provides a forum for the review and discussion of networking issues affecting the Hanford Area Network and is a vehicle for the AMS Project to communicate network needs and impacts and request assistance.

8.3.3 Interface Control

The CCB is the focal point to the AMS interface. There will be a variety of user organizations with different needs that wish to use the AMS. Any such users or prospective users that require modification or enhancements to the AMS will need to prepare the following for submission to the CCB:

- A completed AMS Change Request form, as specified in WHC-CM-3-10, *Software Practices* (see Figure 8-1)
- A Work Authorization as required by WHC-CM-3-10, *Software Practices*, SP-3.1, Paragraph 5.2; and WHC-CM-4-2, *Quality Assurance*, Section QR 19.0, Paragraph 4.3.1
- Requirements
- Acceptance criteria.

A basic level of ongoing maintenance may be funded by open Work Authorizations. Any change that will modify the released AMS configuration to be performed under such authorization requires, as a minimum, specific CCB authorization. For more complex enhancements, the requesting organization must prepare all the items specified previously.

Without the Work Authorization and the CCB authorization (the approved Change Request form) to support development or maintenance activities beyond the basic level covered by ongoing CCB funding, no such work will be performed.

Organizations that desire to use the AMS must submit a request to the CCB for acceptance. Once their application has been approved, and after they have become an active AMS user, organizations are eligible, under conditions specified by the CCB, for membership on the CCB. New users will be expected to provide their share of the cost of CCB-funded ongoing maintenance.

The AMS CCB is responsible for notifying the AMS administrators about any changes to the AMS interface. The AMS administrators are, in turn, responsible for notifying their users about any AMS interface changes.

Except for emergency fixes of problems with no workaround that are preventing use of the system, all maintenance and enhancement activities will be inserted into the AMS in regularly scheduled releases. The first of these will be a minor revision performed not less than 90 days after initial system release. Subsequent releases will be made not less than every 90 days for the first year and not less than every 6 months after that.

Major revisions, if required, will be made at not less than 6-month intervals the first year after initial release and at not less than 12-month intervals in following years. These constraints may be altered through special agreement between the CCB and the performing organization.

If a user's organization develops Paradox procedures that they wish to have incorporated into the AMS, that organization must submit the following to the CCB:

- The proper documentation (as required by WHC-CM-3-10), which has been reviewed and approved by the performing organization for incorporation into the AMS
- The CCB authorization (the approved Change Request form)
- An approved work authorization.

NOTE: Any user-developed procedures must be subjected to the full set of AMS testing, including such additions as may be necessary to test the new procedure being added. This is to ensure the CCB that no regressions have been introduced that will affect the performance of those using the AMS.

8.3.4 Implementation

This SCMP covers upgrades of procured software packages, revisions of the AMS, and additions to the AMS.

8.3.5 Policies and Procedures

The AMS complies with WHC-CM-3-10, *Software Practices*; WHC-CM-4-2, *Quality Assurance*, QR 19.0; WHC-CM-6-1, *Standard Engineering Practices*; and WHC-CM-2-6, *Data Administration Standards*. Any additions or modifications to the AMS also will be made in compliance to these standards.

8.4 SOFTWARE CONFIGURATION ACTIVITIES

8.4.1 Configuration Identification

The documentation for the AMS uses WHC-CM-6-1, *Standard Engineering Practices*, Section EP-1.12, Paragraph 5.1, to uniquely identify supporting documents. The AMS software revision identification will be of the form R.r, where an increment is added to capital "R" for major revisions and small "r" for minor revisions (see WHC-CM-3-10, Section SP-6.2, Paragraph 5.1.2). The uniquely identified minor and major revisions are stored on the performing organization's file server and on backup floppy disks.

8.4.2 Configuration Control

A CCB will be used to approve and prioritize change requests to the AMS. These change requests are used to initiate action regarding a perceived new requirement, enhancement, or correction to the AMS; and for applications for use of AMS by new user organizations. The change request procedure provides for the initiation, tracking, review, evaluation, authorization, implementation, testing, and closeout of each required change to the AMS.

The CCB will operate under the terms of the charter included in Section 8.7.

8.4.3 Configuration Status Accountability

The AMS performing organization keeps accountability for system creation through the use of logs of file extraction and execution. These logs point to specific files and revisions in the online and/or archived AMS programs. When a new revision of the production system is created, a system configuration report is produced showing the hardware and software of the target AMS. The performing organization is responsible for maintaining a permanent copy of each released version of the AMS.

The performing organization will complete a Software Revision Record, as specified in WHC-3-10, for each new release of the AMS. (See Figure 8-2.)

8.4.4 Audits and Reviews

To ensure that changes to documentation, as well as the software, are documented, reviews will be performed. Each new version will be subject to a V&V review and both new feature and regression testing. The initial set of test cases used for initial AMS testing will be maintained, updated with cases to test new functions, and run against any new release. Each test case will contain the acceptance criteria for that test item.

8.4.5 Access Control

Access control will be maintained at two levels:

- Level 1 is restricted to AMS administrators. Ability to perform these administrator functions will be limited to specifically designated personnel and controlled by user ID and password.
- Level 2 is the level allowed to the performing organization software engineer, who will be able to modify software, change database structure, etc. Access to this level is restricted by additional passwords, need to have development tools, etc.

Additionally, AMS provides internal facilities to connect to its server via HLAN. This process is normally not visible to AMS users.

8.4.6 Backup and Retrieval

The AMS software will have two backups made after each software revision. These backups will be documented as a supporting document, the AMS software listings release. This release contains the current information on the AMS software (including procured software), documentation, and hardware status at the time of the release. These backups will be stored in two separate physical locations.

Additionally, the AMS database and master software, which resides on network servers, is backed up nightly by Network Administration. In case of a server failure, a maximum of one day's inputs may be lost.

8.4.7 Disaster Recovery

Users of the AMS do not require a level of availability higher than that provided from normal HLAN service. In case of a loss of data on the file server, Network Administration will restore the files within 24 hours. Loss of software or hardware at the user's location is handled through the AMS administrator and/or the End User Center. If a user's computer fails, it can be swapped for one with the required AMS prerequisites and the AMS administrator can reinstall the local AMS software.

8.5 TOOLS, TECHNIQUES, AND METHODOLOGIES

There are no specific software development tools, techniques, or methodologies that will be used in accomplishing this plan.

8.6 RECORDS COLLECTION AND RETENTION

Records handling will conform to the requirements of WHC-CM-3-10, *Software Practices*.

8.7 CCB CHARTER

8.7.1 Purpose

The purpose of this charter is to establish the AMS CCB and describe the responsibilities of the Board and its members.

8.7.2 Scope

This charter applies to all changes made to the AMS and supporting AMS documents, including, but not limited to, network connections, LANCAL, cc:Mail, or other systems. It also applies to the use of AMS by other organizations. The computer hardware and software configurations required to use the AMS software also fall under the jurisdiction of the AMS CCB.

8.7.3 Membership

The membership of the CCB is as follows:

- Member and chairperson: Thomas G. Halverson, Manager, TWRS Management and Technical Systems
- Member (Secretary): Terri M. Lutter, TWRS DSA
- Member: TBD
- Advisor: Performing organization cognizant engineer
- Advisor: DAC representative
- Advisor: Network Engineering representative
- Future members: Other organization representatives as requested.

Following initial release of the AMS, the AMS CCB will have the authority to revise its membership at any time, subject to a majority approval of the membership.

8.7.4 Membership Responsibilities

A member of the CCB must sponsor a change recommendation.

The board chairperson (or other CCB member designated by the chairperson) shall be responsible for the following:

- Presiding over the board meetings
- Forwarding approved change requests to the performing organization

- Overseeing the CCB recommendations (chairperson has final disposition authority)
- Interfacing with AMS developers and others as required.

The secretary shall be responsible for the following:

- Ensuring that all members and identified participants receive adequate notice of meetings
- Distributing copies of change requests to be reviewed to each member at least 48 hours before each meeting (except where the chairperson has designated a change as an emergency item)
- Publishing and distributing meeting minutes to all CCB members and advisors and any other affected individuals within 2 working days following the board meeting
- Maintaining the CCB charter.

The CCB members are obligated to evaluate changes with consideration of the impact the change would have on the total population of users. Individual preferences should not be allowed to dominate the AMS structure.

The CCB is responsible for the following:

- Reviewing the change request package(s) to ensure all affected functions are appropriately represented and that no unanticipated, undesirable impact to existing users will occur
- Ensuring the adequacy of the description, justification, and impact (if not approved)
- Ensuring cost/benefit of the proposed change(s) to the users is accurately reflected in the change package
- Verifying appropriate technical reviews have been accomplished
- Documenting funding sources on the change request form
 - Giving a priority to pending changes relative to their importance to program when necessary
- Ensuring appropriate distribution of approved/disapproved changes
- Surveying the user community to determine the acceptance of proposed and newly implemented changes
- Approving the requests of new, non-AMS user organizations to become AMS users.

8.7.5 Change Disposition

All change requests will be dispositioned as follows:

- Approved
 - As submitted
 - As amended
- Disapproved
 - With reason(s) stated
- Deferred
 - With reason(s) stated and schedule for resubmittal.

8.7.6 CCB Meetings

The CCB will be scheduled by the Board secretary as needed in consultation with the CCB chairman and Board members.

Figure 8-1. Change Request and Problem Report Guidelines. (sheet 1 of 2)

Change Request and Problem Report

	CR/PR Number: _____
1. Software/Document Identification (Name): _____	Rev: _____
2. Prepared by: _____	Date: _____
System Name: _____	TPCN, W/O: _____
3. CR/PR Type: ☐ Change Request ☐ Problem Report	Requested Completion Date: _____
4. Description: _____	
5. Justification If Change Request: _____	
6. Submitter's Priority ☐ High ☐ Medium ☐ Low	
7. Change Authority: _____ ☐ Accept ☐ Modify ☐ Reject ☐ Defer Until: _____	
8. Assigned to: _____	Planned Release Date: _____
9. Solution Comments	Cost/Schedule Estimate: _____ / _____
10. Software/Documents Affected: _____	
11. Approvals Indicate CR is Complete or PR is Resolved.	
Software Developer: _____	Date: _____
Cognizant Manager: _____	Date: _____
CR or PR Preparer: _____	Date: _____
Other: _____	Date: _____

Figure 8-1. Change Request and Problem Report Log. (sheet 2 of 2)

CHANGE REQUEST AND PROBLEM REPORT LOG

SYSTEM NAME _____

CUSTODIAN NAME _____

CR/PR # DATE SUBMITTED

SUBMITTED BY

PRIORITY

TITLE

STATUS/RESOLUTION

Figure 8-2. Software Revision Record Guidelines. (sheet 1 of 2)

Software Revision Record

1. Software ID (Name) _____ Rev _____
 2. ☐ Release ☐ Change
 3. Prepared by _____ Date _____
 4. Description (reference Change Request and Problem Report Numbers)
 5. Software File Identification (or attach directory listing) Name Rev
 6. Record Copy of Software Files - Storage Media Identification and Location
 7. Installation Identification Name Location
 8. Documentation Identification Name Rev
 9. Environment Identification Name Rev
Hardware
Operating System
Language
 10. The software identified on this form is approved for ☐ Operation ☐ Testing ☐ Other(specify)
- Software Developer _____ Date _____
- Cognizant Manager _____ Date _____
- Other _____ Date _____

Figure 8-2. Software Revision Record Guidelines. (sheet 2 of 2)

These instructions provide guidelines for preparing the software revision record. If more space is needed, use blank pages and attach to the software revision record. This will be the record of the software released or changed.

1. Record software identification or name and revision number.
2. Indicate whether this is a release or a change to existing software.
3. Record the name of the person preparing the information and the date prepared.
4. Provide a brief description of the reason for the change. Reference all change requests and problem reports that are satisfied by this revision of the software.
5. Record the name and other identifying information for each of the software files that will be included in the release or change, or attach a directory listing with the identifying information.
6. Record the identification, location, and directory names of the copy of the software (source code) files for this release or change.
7. Record the identification, name, and location where each installation of the software will be made.
8. Record the document identification, name, and revision of applicable documentation.
9. Record the operating environment (including the hardware, operating system, and programming language names and revisions) used to operate the software identified on this form.
10. Indicate whether the approvals are for operation, testing, or some other purpose, and record approvals.

This page intentionally left blank.

9.0 TEST PLAN

This section presents the plan for testing and validating the AMS application, its associated database, user documentation, installation procedure, and hardware platform(s).

The following documents will provide the basis for defining correct operation and testing procedures:

- AMS SRS (Section 4.0 of this document)
- AMS SDD (Section 6.0 of this document)
- *Activity Management System User Reference Manual* (WHC-SD-HWV-UM-004) (Gates 1994).

9.1 AMS MODULES

The AMS is a screen-driven system that will be tested within its normal Paradox and MS-DOS* operating environment. As of September 1993, there are:

- 26 menu screens
- 36 dialogue screens
- 10 data entry/display screens
- 2 help screens
- 2 miscellaneous screens
- Error message screens.

Each of these is driven by a code module that has been written in the PAL. A development tool called Paint Pal was used to generate the PAL code that implements the many AMS dialogue boxes.

9.2 USER PROCEDURES

The online operating procedures contained in WHC-SD-HWV-UM-004, *Activity Management System User Reference Manual*, will be tested and certified correct to the AMS SRS (Section 4.0).

9.3 FEATURES TO BE TESTED

The following system features will be tested:

- Menu functions
- Database file activities (add/change/delete)
 - Activity data
 - Activity group data
 - Viewer list data
 - Mail list data
 - User password

*MS-DOS is a trademark of the Microsoft Corporation.

- Display of selected database information
- Reporting selected activity information
 - To printer
 - To screen
 - To files
 - To cc:Mail
- Administrative functions.

9.4 FEATURES NOT TO BE TESTED

The features of AMS that will not be addressed as part of this test plan include HLAN, computer-related hardware, the MS-DOS operating system, and the Paradox database software. These components are not part of the AMS design and will not be tested separately.

9.5 AMS TESTING APPROACH

The AMS has already undergone intensive user testing by virtue of the prototype development approach. Because the same PAL, Paint Pal, and other development tools were used to develop the prototypes as have been used for the production version, it was not necessary to reimplement the application at the end of prototype testing. Several versions of the prototype (the final being very close to the production version) have been installed at the HWVP users' locations and used by 16 or more users in a "production" environment. Despite this thorough, informal testing, a formal test must be developed and run to ensure adequate testing of the AMS functions and the creation of a baseline for regression testing future revisions of the application.

The performing organization will use WHC-SD-HWV-UM-004, *Activity Management System User Reference Manual*, and the SRS (Section 4.0) to design and create test procedures. The AMS test data will approximate real-world scenarios based on HWVP Project input and documented system requirements.

Testing will be performed by one or more employees of the HWVP Project Office, or other qualified persons who have not been involved with the AMS development.

Before starting AMS testing, the file server and user computer acceptance test areas are to be empty, with no databases or programs in residence. Movement of programs, scripts, or data to the acceptance test areas will be strictly controlled by the AMS test team. All elements that are to be loaded will be documented and the document will be retained with other testing documentation. Each separate software component will be uniquely identified by name, version, creation date, file size, and any other relevant data.

A complete set of test scripts will be prepared to test all the system functions that are defined in Section 4.0. A matrix will be prepared to show which scripts test which function. Each script will be identified by the name of the tester, start time, completion time, and the result (pass/fail). The scripts will be designed to check individual application components and the integration of these components.

If a test script ends in failure, the testing for that function will be halted at that point until a new version of the software, with error corrections, is installed and a complete regression test is executed (rerunning of all completed test scripts).

All testing will be performed at the customer site, away from the development area.

9.5.1 Job Stream Testing

Job stream testing is not applicable to AMS.

9.5.2 Interface Testing

Execution of the test scripts will test the interface with HLAN and cc:Mail. The human interface will be tested as a result of the testers following the test scripts and using the keyboard and mouse to navigate through the various AMS functions.

9.5.3 Security Testing

All AMS access privileges and protection features will be tested through execution of the scripts.

9.5.4 Regression Testing

Regression testing is the retest process for features that fail test criteria (and, in the future, for testing new releases of the AMS software). Retests will be performed as soon as corrections have been accomplished and the corrected software has been moved to the test directory. Results of the regression testing will be recorded. This testing is performed to ensure that modifications to the application have not corrupted other areas of the AMS.

9.5.5 User/Machine Testing

Each of the system features described in WHC-SD-HWV-UM-004, *Activity Management System User Reference Manual* (Gates 1994), will have at least one associated test performed.

9.6 ITEM PASS/FAIL CRITERIA

The following criteria determine when a test item passes the established requirements defined by the AMS SRS.

- Menu screens provide the prescribed system navigational capabilities.

- Update screens properly function as defined in WHC-SD-HWV-UM-004, *Activity Management System User Reference Manual*, and the AMS SRS. Database elements are correctly updated. New data records/elements are created and stored in their proper files in the database.
- Display screens produce the required output in the prescribed format in accordance with the data selection criteria.
- Report screens produce the required output in the prescribed format in accordance with the data selection criteria.
- User IDs and passwords function according to WHC-SD-HWV-UM-004, *Activity Management System User Reference Manual*, and the AMS SRS, and restrict unauthorized entry into the system and unauthorized access to, and updating of, activity data.

9.7 TEST PROCEDURES

The test detailed in the following sections will be incorporated into WHC-SD-HWV-TC-003, *Activity Management System Software Test Procedures* (Burdick 1993), which will contain detailed scripts to lead the testers step-by-step through the entire set of tests. It also will define formats to be used for documenting results, problems, etc. This document will contain a table to cross-reference the task numbers shown below to the specific test scripts that perform the tests.

The areas to be tested are arbitrarily subdivided into the following:

Section 9.7.1, Database Certification
Section 9.7.2, Menus
Section 9.7.3, Update Screens
Section 9.7.4, Display Screens
Section 9.7.5, Report Screens
Section 9.7.6, Security
Section 9.7.7, Admin Functions.

9.7.1 Database Certification

<u>Task No.</u>	<u>Description</u>
DB001	Verify that no Paradox tables, PAL procedures, or other software exists in the testing area of the file server or the user computer systems.
DB002	Copy the necessary software to the file server and user computers.
DB003	Verify that all necessary Paradox tables have been created with the correct characteristics.
DB004	Certify the AMS database at the end of the test procedures.

9.7.2 Menus

<u>Task No.</u>	<u>Description</u>
MS001	Menu navigation capabilities. Select each item on the master menu and module menus to test navigation through the system from the menus. Use the keyboard and the mouse for this process. Test all paths.

9.7.3 Update Screens

<u>Task No.</u>	<u>Description</u>
US001	Create new activities, retrieve the data, and check each field for format and content.
US002	Input erroneous data to verify edit checking and error message capabilities. (Do not use for fields that do not edit input data.)
US003	Retrieve an activity, change the data in all permitted fields, and update the database. Retrieve the activity and certify that all updates took place correctly. This will require cooperation between testers designated as originator, agent, and viewers on the activities.
US004	Repeat this test as many times as necessary to ensure that the maintenance functions are performing as specified.

9.7.4 Display Screens

<u>Task No.</u>	<u>Description</u>
DS001	Execute each display screen and check the results. For those displays that have variable inputs, execute multiple times with different input values.

9.7.5 Report Screens

<u>Task No.</u>	<u>Description</u>
RS001	Execute each report screen and check the results. For those displays that have variable inputs, execute multiple times with different input values. Direct output to screens, printer, files, and cc:Mail and verify that all results are correct.

9.7.6 Security

<u>Task No.</u>	<u>Description</u>
SC001	Verify that user IDs and passwords are correctly checked and verified. Verify that unregistered user IDs and passwords are rejected.
SC002	Verify data security by attempting to update fields that are not authorized based on the tester's role on the activity.
SC003	Verify that the change password function works and that the tester can log in with the new data.

9.7.7 Admin Functions

<u>Task No.</u>	<u>Description</u>
AD001	Test all user functions. Add a new user, delete a user, disable and enable a user, and test that the command works. Edit the user data and change a user's ID. Test that the "list users online" operation works correctly.
AD002	Test all activity functions. Execute the Unvalidated Activity Report and verify its correctness. Transfer an originator's activities to a new user. Test the purge, unpurge, and remove archive tables functions.
AD003	Test the miscellaneous functions. Execute an organization change, perform a cc:Mail list update, and create update, and delete a global user list. Verify that all functions produced the correct results.
AD004	Test that the user can enter Paradox and is prevented from modifying any AMS data without the correct access passwords.

10.0 V&V FINAL REPORT

A final V&V review will be conducted by an independent reviewer. The final report of this V&V activity will be retained in the project files.

This page intentionally left blank.

11.0 REFERENCES

- ANSI/IEEE, 1984, *IEEE Guide to Software Requirements Specifications*, ANSI/IEEE 830-1984, American National Standards Institute and the Institute of Electrical and Electronics Engineers, New York, New York.
- ANSI/IEEE, 1989, *Standard Glossary of Software Engineering Technology*, ANSI/IEEE 729-1989, American National Standards Institute and the Institute of Electrical and Electronics Engineers, New York, New York.
- Burdick, M. B., 1993, *Activity Management System Software Test Procedures*, WHC-SD-HWV-TC-003, Rev. 0, Westinghouse Hanford Company, Richland, Washington.
- Gates, T. A., 1994, *Activity Management System User Reference Manual*, WHC-SD-HWV-UM-004, Rev. 1, Westinghouse Hanford Company, Richland, Washington.
- IEEE, 1983, *Software Configuration Plans*, IEEE 823-1983, Institute of Electrical and Electronics Engineers, New York, New York.
- IEEE, 1987, *Software Design Descriptions*, IEEE 1016-1987, Institute of Electrical and Electronics Engineers, New York, New York.
- IEEE, 1990, *IEEE Standard Glossary of Software Engineering Terminology*, IEEE 610.12-1990, Institute of Electrical and Electronics Engineers, New York, New York.
- WHC-CM-2-6, *Data Administration Standards*, Westinghouse Hanford Company, Richland, Washington.
- WHC-CM-3-10, *Software Practices*, Westinghouse Hanford Company, Richland, Washington.
- WHC-CM-4-2, *Quality Assurance*, Westinghouse Hanford Company, Richland, Washington.
- WHC-CM-6-1, *Standard Engineering Practices*, Westinghouse Hanford Company, Richland, Washington.

This page intentionally left blank.

12.0 ACRONYMS AND GLOSSARY

12.1 ACRONYMS

This is a list of the abbreviations, acronyms, and initialisms that are used in this document.

AMS	Activity Management System
ANSI	American National Standards Institute
BCSR	BCS Richland, Inc.
CCB	Configuration Control Board
CCIS	cc:Mail Import System
CTS	Commitment Tracking System
DAC	Data Administration Council
DFD	data flow diagram
DSA	Data Standards and Administration
HLAN	Hanford Local Area Network
HWVP	Hanford Waste Vitrification Plant .
ID	identification
IEEE	Institute of Electrical and Electronics Engineers
IRM	Information Resource Management
PAL	Paradox application language
PC	personal computer
QA	Quality Assurance
RDBMS	relational database management system
RL	U.S. Department of Energy, Richland Operations Office
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SDD	Software Design Description
SRS	software requirements specification
TWRS	Tank Waste Remediation System
V&V	verification and validation
WBS	work breakdown structure

12.2 GLOSSARY

Agent. AMS user identified to perform activity.

Column. A subdivision of a table with a column name and a specific data type.

Commit. The process of storing new, changed, or deleted records from the work space to a table in the database.

Database. The disk space that Paradox sets aside for storing information in tables.

Design. The process of defining the software architecture, components, modules, interfaces, test approach, and data for a software system to satisfy requirements.

Design entity. An element (component) of a design that is structurally and functionally distinct from other elements and that is separately named and referenced.

Dialogue box. A framed object on the screen, usually containing OK and Cancel buttons. One interacts with a dialogue box by typing information into a text box or by choosing from a pick list.

Entity attribute. A named characteristic or property of a design entity. It provides a statement of fact about the entity.

Field. In a table, the information stored at the intersection of a row and a column.

Forms. A set of fields and constant text that Paradox displays on a terminal device so users can see and change data that is stored in database tables.

Hardware. Physical equipment used in data processing, as opposed to computer programs, procedures, rules, and associated documentation.

Interface. (1) A shared boundary; (2) to interact or communicate with another system component.

Mandatory. A field characteristic denoting a field whose value may not be null.

Memo field. A variable-length field that can contain any alphanumeric characters. A memo field can be up to 64 megabytes in size.

Module. (1) A program unit that is discrete and identifiable with respect to compiling, combining with other units and loading (American National Standards Institute); (2) a logically separable part of a program.

Originator. AMS user who creates an activity.

Program. A computer program.

Program architecture. The structure and relationship among components of a computer program. The program architecture also may include the program's interface with its operational environment.

Record. A unit of information in the Paradox work space that corresponds to a row in the database table.

Requirement. (1) A condition or capability needed by a user to solve a problem or achieve an objective; (2) a condition or capability that must be met by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. The set of all requirements forms the basis for subsequent development of the system or system component.

Row. One set of fields in the database.

Software. Computer programs, procedures, rules, and associated documentation and data pertaining to the operation of a computer system.

Software Design Description (SDD). A representation of a software system created to facilitate analysis, planning, implementation, and decision making. A blueprint or model of the software system. The SDD is used as the primary medium for communication software design information.

Subprogram. A program unit that may be invoked by one or more other program units. Examples are procedure, function, and subroutine.

System Architecture. The structure and relationships among components of a system. The system architecture also may include the system's interface with its operational environment.

Table. A named collection of related information stored in the Paradox database.

This page intentionally left blank.