Safety and Fitness Electronic Records (SAFER) System Master Test Plan

November 30, 1996

Prepared for:

Federal Highway Administration

Prepared by:

The Johns Hopkins University
Applied Physics Laboratory

Please note that this is a Draft Issue

It is important to note that this is **a** draft document. The document is incomplete and may contain sections that have not been completely reviewed internally. The material presented herein will undergo several iterations of review and comment before a baseline version is published.

DRAFT

Safety and Fitness Electronic Records (SAFER) System Master Test Plan

Table of Contents

DRAFT

Safety and Fitness Electronic Records (SAFER) System Master Test Plan

1. Purpose

The purpose of this plan is to establish a formal set of guidelines and activities to be adhered to and performed by JHU/APL and the developer to ensure that the SAFER System has been tested successfully and is fully compliant with the SAFER System requirements.

The initial release of this document, submitted in draft form, provides a general framework for establishing the testing environment and provides general guidelines for performing unit, integration, system, and acceptance testing of the SAFER System. Several draft versions of this document will be issued, following review and comment by the developer, as details regarding the software design evolve.

2. Reference Documents

Software Development and Documentation, Military Standard, MIL-STD-498, 5 December 1994, AMSC NO. N7069.

IEEE Standard for Software Quality Assurance Plans, ANSI/IEEE Std 730-1984, June 14, 1984

POR-5804, Trident II Data processing Plan, Vol. 1, Johns Hopkins University/Applied Physics Laboratory, January 1991

3. Definitions

The information presented below represents a definition of terms used throughout this document.

- Approval. Written notification by an authorized representative of the acquirer that a developer’s plans, design, or other aspects of the project appear to be sound and can be used as the basis for further work. Such approval does not shift responsibility from the developer to meet contractual requirements.

- Computer program. A combination of computer instructions and data definitions that enable computer hardware to perform computational or control functions.

Computer Software Configuration Item (CSCI). An aggregation of software that satisfies an end use function and is designated for separate configuration management by the acquirer. CSCIs are selected based on tradeoffs among software function, size, host or target computers, developer, support concept, plans for reuse, criticality, interface considerations, need to be separately documented and controlled, and other factors.

– Developer. An organization that develops software products ("develops" may include new development, modification, reuse, reengineering, maintenance, or any other activity that results in software products.

Pass/Fail Criteria. Decision rules used to determine whether a software item or a software feature passes or fails a test.

Software development file (SDF). A repository for material pertinent to the development of a particular body of software. Contents typically include (either directly or by reference) considerations, rationale, and constraints related to requirements analysis, design, and implementation; developer-internal test information; and schedule and status information.

Software Feature. A distinguishing characteristic of a software item, e.g., performance, portability, functionality.

Software Item. Source code, object code, job control code, control data, or a collection of these items.

Software test environment. The facilities, hardware, software, firmware, procedures, and documentation needed to perform qualification, and possibly other, testing of software. Elements may include but are not limited to simulators, code analyzers, test case generators, and path analyzers, and may also include elements used in the software engineering environment.

– Software unit. An element in the design of a CSCI; for example, a major subdivision of a CSCI, a component of that subdivision, a class, object, module, function, routine, or database. Software units may occur at different levels of a hierarchy and may consist of other software units. Software units in the design may or may not have a one-to-one relationship with the code and data entities (routines, procedures, databases, data files, etc.) that implement them or with the computer files containing those entities.

Test item. A software item which is an object of testing.

– Test Log. A chronological record of relevant details about the execution of the tests.

Test Summary Report. A document or set of documents summarizing testing activities and results.

Testing. The process of analyzing a software item to detect the differences between existing and required conditions, i.e., bugs, and to evaluate the features of the software item.

4. Scope

This test plan covers general guidelines for performing unit, integration, system, and acceptance testing of the SAFER System. These guidelines will be expanded to include specific test design, case, and procedure specifications as details regarding the software design evolve. Testing will ultimately cover operator and user procedures, as well as programs and processing control. In addition to comprehensively testing multi-process functionality including multi-threading, inter-process communications and multi-processor utilization, external interfaces, security, recovery and performance will also be evaluated.

5. Features to be Tested

The SAFER System, as specified in the SAFER Physical Architecture Document, is partitioned into seven CSCIs. These are:

- Input Message Handler
- Administrative Manager
- Subscriber Processor
- External Request Processor
- Safety Data Manager
- Output Message Handler
- OPCON Manager

The features of each CSCI will be fully defined and documented in the SAFER Detailed Software Design Document. In subsequent versions of the Master Test Plan, the specific features of each CSCI and their inter-relationships will be explicitly identified for testing purposes.

6. Features Not to be Tested

Features of the SAFER System which are not to be tested initially are TBD.

7. Approach

7.1 Program Phases

The SAFER Project will be conducted in two phases:

SAFER 96 Development & Test

SAFER 97 Development, Test & Production

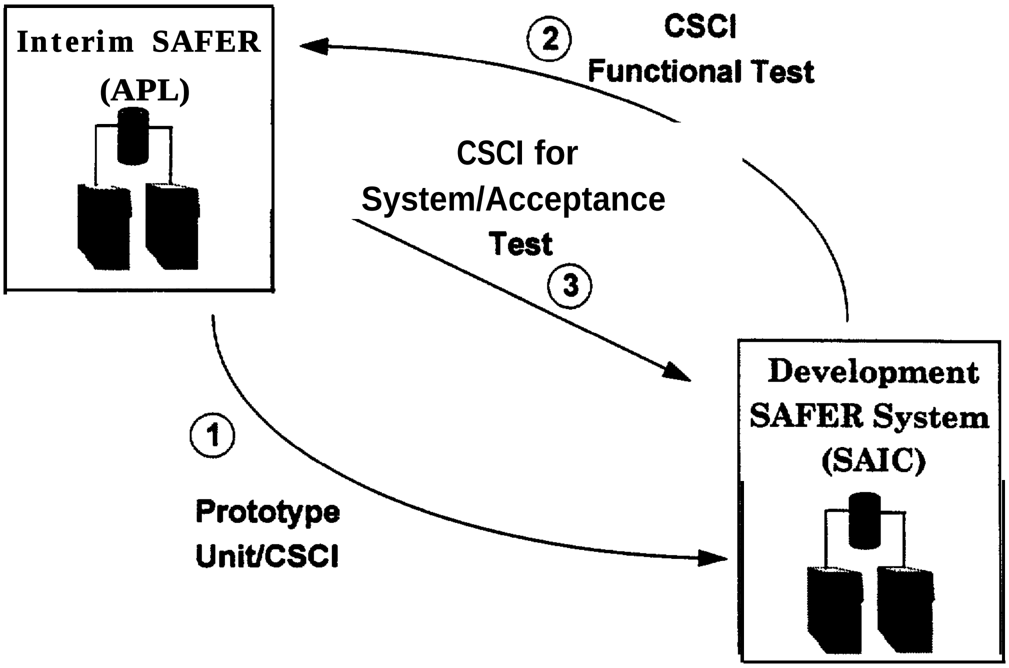
In the SAFER 96 Development & Test Phase, the SAFER 96 System will be designed, developed and tested. Deployment and operation and maintenance activities will be initiated in December of 1996. Deployment will be completed, i.e., the System deployed to support no fewer than 200 MCSAP sites, by June 1997. A preliminary system analysis and design of the second build of the SAFER System, termed SAFER 97, will also be performed.

In the SAFER 97 Development, Test & Production Phase, the SAFER 97 System will be developed based on additional requirements identified during the development effort of the SAFER 96 and 100/200 MCSAP Site Projects while the SAFER 96 System supports production operations. The SAFER 97 System will be deployed to support the 100/200 MCSAP Site Project no later than December 1997.

7.2 General Approach

Software development and testing shall be a collaborative effort between the JHU/APL and the developer. A graphical depiction of the 1996 development and test approach is shown in Figure 1.

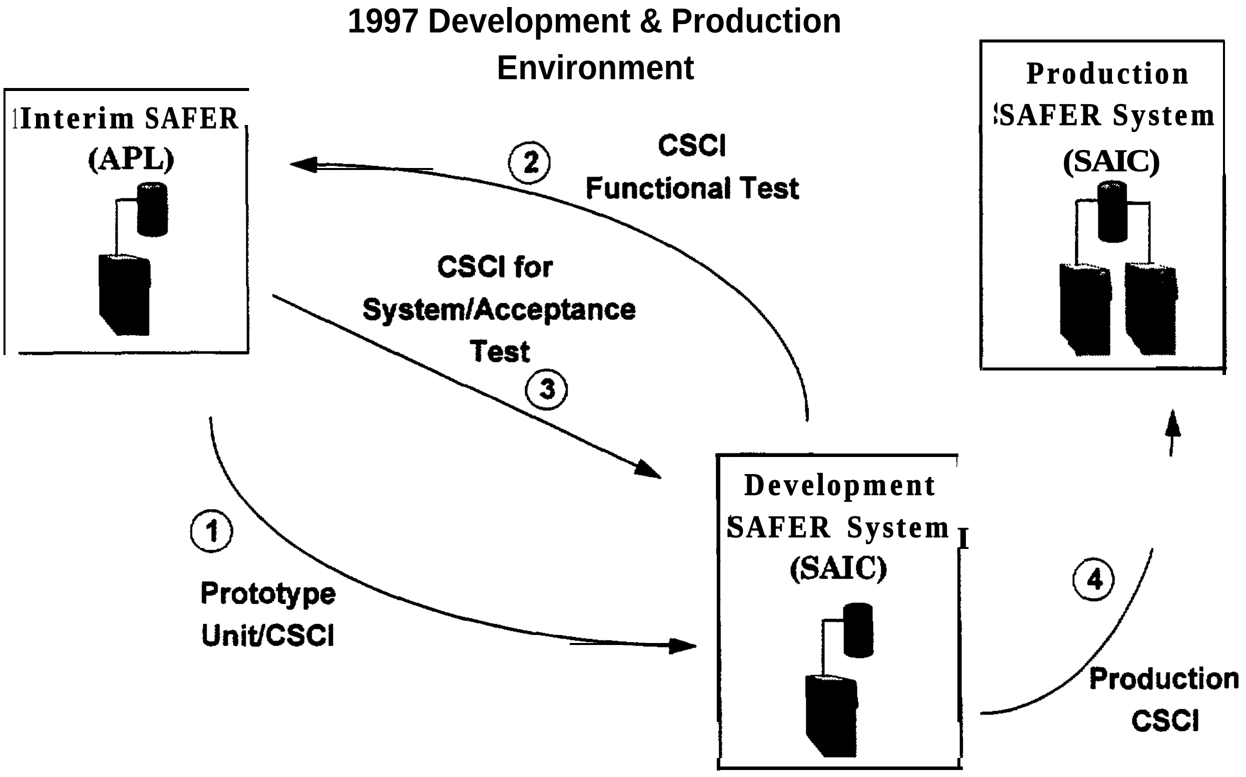
Figure 1. SAFER 96 Software Development & Testing Approach



During the 1996 development and test cycle, JHU/APL shall be responsible for developing and maintaining an interim version of the SAFER System at its facility. This system is henceforth referred to as the Interim SAFER System and shall be used for software prototype development and testing. Prototype software shall be developed by JHU/APL, in accordance with the specifications of the SAFER Logical and Physical Architecture, and supplied to the developer.

The developer shall be responsible for developing and maintaining a development version of the production SAFER System, henceforth referred to as the SAFER System. The developer shall review and modify, where applicable, prototype software supplied by JHU/APL and incorporate it into the SAFER System. The developer shall provide JHU/APL with builds (releases) of the production software, at the CSCI level or above, to be installed on the Interim SAFER System, replacing prototype code, for testing in accordance with the Master Test Plan. Concurrent testing shall also be conducted at the developer's facility. Software deficiencies, detected during testing, shall be corrected by the developer and the corrected code re-installed on the Interim SAFER System for further testing. This iterative cycle shall continue until all detected deficiencies have been eliminated.

Figure 2. SAFER Software Development & Testing Approach


1997 Development & Production Environment
Interim SAFER (APL)
2
CSCI Functional Test
Production SAFER System (SAIC)
CSCI for System/Acceptance Test
3
1
Prototype Unit/CSCI
Development SAFER System (SAIC)
4
Production CSCI

During the 1997 development and test cycle, production operations must also be supported at the SAIC facility. This environment is graphically depicted in Figure 2. At the start of the 1997 development and test cycle, one of the Interim SAFER processors will be migrated to SAIC to serve as their development processor while their original development system will become the production SAFER System.

APL and the developer will use the system documentation to prepare all test design, case, and procedure specifications to support unit, integration, system, and acceptance testing. This approach will verify the accuracy and comprehensives of the information in the documentation in those areas covered by the tests.

7.3 Unit Testing

Unit testing means ensuring that all aspects of each software unit's detailed design are comprehensively tested.

7.3.1 Preparing for unit testing

The developer shall establish test cases (in terms of inputs, expected results, and evaluation criteria), test procedures, and test data for testing the software corresponding to each software unit. The test cases shall cover all aspects of the unit's detailed design. The developer shall record this information in the appropriate software development files (SDFs).

7.3.2 Performing unit testing.

The developer shall test the software corresponding to each software unit. The testing shall be in accordance with the unit test cases and procedures.

7.3.3 Revision and retesting.

The developer shall make all necessary revisions to the software, perform all necessary retesting, and update the software development files (SDFs) and other software products as needed, based on the results of unit testing.

7.3.4 Analyzing and recording unit test results

The developer shall analyze the results of unit testing and shall record the test and analysis results in appropriate software development files (SDFs).

7.4 Integration Testing

The developer shall perform integration testing in accordance with the following requirements.

Note 1: Integration testing means integrating the software corresponding to two or more software units, testing the resulting software to ensure that it works together as intended, and continuing this process until all software in each CSCI is integrated and tested.

Note 2: If a CSCI is developed in multiple builds, integration testing of that CSCI will not be completed until the final build. Integration testing in each build should be interpreted to mean integrating software developed in the current build with other software developed in that and previous builds, and testing the results.

7.4.1 Preparing for integration testing

The developer shall establish test cases (in terms of inputs, expected results, and evaluation criteria), test procedures, and test data for conducting integration testing. The test cases shall cover all aspects of the CSCI architectural design. This information shall be recorded by the developer in the appropriate software development files (SDFs).

7.4.2 Performing integration testing

The developer shall perform integration testing. The testing shall be in accordance with the integration test cases and procedures. JHU/APL shall also perform independent functional tests of each CSCI.

7.4.3 Revision and retesting.

The developer shall make all necessary revisions to the software, perform all necessary retesting, in conjunction with JHU/APL, and update the software development files (SDFs) and other software products as needed, based on the results of integration testing.

7.4.4 Analyzing and recording integration test results.

The developer shall analyze the results of integration testing. Testing and analysis results shall be recorded in the Integration Test Results Document by the developer and be reviewed and approved by JHU/APL.

7.5 System Testing

The developer shall participate in System testing activities in accordance with the following requirements.

Note 1: System testing means integrating CSCIs with interfacing CSCIs, testing the resulting groupings to determine whether they work together as intended, and continuing this process until all CSCIs in the system are integrated and tested.

Note 2: If a system or CSCI is developed in multiple builds, system testing may not be complete until the final build. System testing in each build should be interpreted to mean integrating the current build of each CSCI with the current build of other and testing the results to ensure that the system requirements to be implemented in that build have been met.

7.5.1 Preparing for system testing

The developer and JHU/APL shall participate in developing and recording test cases (in terms of inputs, expected results, and evaluation criteria), test procedures, and test data for conducting system testing. The test cases shall cover all aspects of the system-wide and system architectural design. The developer shall record software-related information in appropriate software development files (SDFs).

7.5.2 Performing system testing.

The developer and JHU/APL shall participate in system testing. The testing shall be in accordance with the system test cases and procedures.

7.5.3 Revision and retesting.

The developer shall make necessary revisions to the software, participate in all necessary retesting, in conjunction with JHU/APL, and update the appropriate software development files (SDFs) and other software products as needed, based on the results of system testing.

7.5.4 Analyzing and recording system test results.

JHU/APL shall be responsible for analyzing the results of system testing. JHU/APL shall document analysis and test results in the System Test Results Document.

7.6 Acceptance Testing

The developer shall participate in system acceptance testing in accordance with the following requirements.

Note: Acceptance testing is performed to demonstrate to the Federal Highway Administration Office of Motor Carriers (FHWA/OMC) that all system requirements have been met.

7.6.1 Preparing for system acceptance testing.

The developer and JHU/APL shall participate in developing and recording the test preparations, test cases, test procedures and test data to be used for acceptance testing and the traceability between the test cases and the system requirements.

7.6.2 Performing acceptance testing

The developer and JHU/APL shall participate in acceptance testing. This participation shall be in accordance with the acceptance test cases and procedures.

7.6.3 Independence in acceptance testing

The person(s) responsible for performing acceptance testing shall not be the persons who performed detailed design or implementation of software in the system. This does not preclude persons who performed detailed design or implementation of software in the system from contributing to the process, for example, by contributing test cases that rely on knowledge of the system's internal implementation

7.6.4 Revision and retesting

The developer shall make necessary revisions to the software, provide JHU/APL advance notice of retesting, participate in all necessary retesting, in conjunction with JHU/APL, and update the software development files (SDFs) and other software products as needed, based on the results of acceptance testing.

7.6.5 Analyzing and recording acceptance test results

JHU/APL shall be responsible for analyzing and recording the results of acceptance testing. The results shall be documented in the Acceptance Test Results Document.

8. Item Pass/Fail Criteria

Requirements for determining item pass/fail criteria are TBD.

9. Test Suspension and Resumption Criteria

Requirements for determining test suspension and resumption criteria are TBD.

10. Environmental Requirements

The developer shall establish, control, and maintain a software test environment to perform unit, integration, system and acceptance testing of software. The developer shall ensure that each element of the environment performs its intended functions.

11. Responsibilities

Specific testing responsibilities assigned to JHU/APL and the developer for unit, integration, functional CSCI, system, and acceptance testing are summarized in the table below.

Test Type	Participant		Facility	
	APL	SAIC	APL	SAIC
Unit Test		x		x
Integration Test		x		x
CSCI Functional Test	x		x	
System Test	x	x		x
Acceptance Test	x	x		x

12. Deliverables, Milestones & Schedules

JHU/APL has responsibility for the following software testing deliverables and milestones:

Phase 1 Testing Deliverables:

Dec 1995	Master Test Plan
Oct 1996	System Test Results Document
Nov 1996	Acceptance Test Results Document

Phase 2 Testing Deliverables:

Oct 1997	System Test Results Document
Nov 1997	Acceptance Test Results Document

The developer has responsibility for the following software testing deliverables and milestones:

Phase 1 Testing Deliverables:

May 1996	Completion of Software Coding
Sep 1996	Completion of Unit, Integration & System Testing
Sep 1996	Integration Test Results Document
Oct 1996	Completion of Field Acceptance Testing

Phase 2 Testing Deliverables:

May 1997	Completion of Software Coding
Sep 1997	Completion of Unit, Integration & System Testing
Sep 1997	Integration Test Results Documents
Oct 1997	Completion of Field Acceptance Testing

13. Problem Reporting and Corrective Action

Problem reporting and corrective action issues were addressed in the SAFER Quality Assurance Plan.

14. Tools, Techniques, and Methodologies

Issues related to tools, techniques, and methodologies were addressed in the SAFER Quality Assurance Plan.

15. Approvals

Approval for the satisfactory completion of unit testing is the responsibility of the developer with the concurrence of JHU/APL. Approval for the satisfactory completion of integration, system, and acceptance testing must be obtained from JHU/APL and FHWA/OMC prior to the release of the SAFER System for production processing.

Appendix A

SAFER System

Acceptance Test Plans and Test Reports

Acceptance. Test Plan

Acceptance ID: Web Access
ATP ID: WEB-WA-001
Version Control ID: APL V01-B02
Affected CSCI: WEB
Acceptance Function:

1. Access the SAFER Web Server via the Internet or direct dial-up using MicroSoft's Remote Access Service (RAS) via the SAFER 800#

Reauirement to be validated:

- Access the SAFER Web Server via and Internet Service provide (ISP)
- Access the SAFER Web Server using direct RAS dial-up via the SAFER 800# service
- Access is accomplished via the following Web browsers:
  - Netscape Navigator Version 3.xx
  - Netscape Navigator Gold Version 3.xx
  - Microsoft's Internet Explorer Version 3.xx
- Access is restricted to defined SAFER users holding a valid SAFER userid and password
- Access is also supported for an unrestricted environment when the userid and password requirement is removed

Test Tools, Drivers, or Special Conditions:
Testing via each of the browsers specified above.

Input Data:
The correct SAFER Uniform Resource Locator (URL): \\isafer2\jhuapl.edu

Expected Outcome:

Successful connection to the SAFER Web Server and display of the SAFER Home Page

Test Procedure Steps:

- Direct dial the SAFER 800# service (800-404-3946) using RAS or access the Internet via an ISP
- Once connected, execute each of the browsers listed above
- Supply the correct SAFER URL at the http: prompt
- Supply a valid SAFER userid and password; this step is only required for the restricted environment case

Analysis Procedures: None

Acceptance Test Report

Acceptance ID: SAFER Web
ATP ID: WEB-WA-001
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: N.D. Crouch Test Date: 11/25/96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Direct dial of the SAFER 800# service (800-404-3946) using RAS was successfully completed
- In the restricted environment, access was only granted to a user with a valid SAFER userid and password; in the unrestricted environment, no userid or password were required
- Once connected, each browser listed above was executed and supplied the correct SAFER URL
- For each browser, the SAFER Home Page was successfully accessed and displayed
- The above results were successfully duplicated using an ISP access method

Reviewed By (SAIC): ________________ Date: ____________

Approved By (JHU/APL): ________________ Date: ____________

Acceptance Test Plan

Accep tance ID: Web Home Page
ATP ID: WEB-WHP-001
Version Control ID: APL V01-B02
Affected CSCI: WEB
Acceptance Function:

1. Proper display and function of the SAFER Web Home Page

Requirement to be validated:

- SAFER logo is displayed properly
- Home page text is displayed properly and can be scrolled vertically and horizontally, if necessary
- The SAFER Database Queries link properly links to the SAFER Query Page
- The browser anomalies example link properly links to the browser anomalies example page
- When the user's browser is properly configured for e-mail support, the SAFER Deployment Coordinator link properly sends e-mail to the SAFER Deployment Coordinator

Test Tools, Drivers, or Special Conditions:
Testing via each of the browsers specified in test WEB-WA-001

Input Data:
Appropriate e-mail configuration data

Expected Outcome:

Successful validation of each of the specified requirements

Test Procedure Steps:
After accessing and displaying the SAFER Home Page, validate that each of the specified requirements has been satisfied

Analysis Procedures: None

Acceptance Test Report

Acceptance ID: Web Home Page
ATP ID: WEB-WHP-001
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: N.D. Crouch Test Date: 11/25/96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The SAFER logo was displayed properly
- The Home page text was displayed properly and could be scrolled vertically and horizontally, if necessary
- The SAFER Database Queries link properly linked to the SAFER Query Page
- The browser anomalies example link properly linked to the browser anomalies example page
- When the user's browser was properly configured for e-mail support, the SAFER Deployment Coordinator link properly sent e-mail to the SAFER Deployment Coordinator

Reviewed By (SAIC): ________________ Date: ____________

Approved By (JHU/APL): ________________ Date: ____________

Acceptance Test Plan

Acceptance ID: Web Query Page
ATP ID: WEB-WQP-001
Version Control ID: APL V01-B02
Affected CSCI: WEB
Acceptance Function:

1. Proper display and function of the SAFER Web Query Page on initial entry

Requirements to be validated:

- The SAFER Query Page consists of a selection and a data display frame and that each is displayed properly
- The radio buttons in the Search Type area of the selection frame operate properly and allow the user to select either or but not both the DOT and ICC/MC number types
- The ID Number area of the selection frame properly accepts numeric and non-numeric entries with the following results upon query submission:
  - Valid DOT and ICC/MC numbers return the correct carrier snapshot
  - Invalid DOT and ICC/MC numbers generate a user error message
  - Alpha-numeric entries generate a user error message
  - Empty entries generate a user error message
- The Search button submits a query to the SAFER/Carrier System based on the value entered in the ID Number area
- The Reset button properly resets, i.e., blanks, the ID Number area
- The explanatory text is properly displayed in the data display frame when the page is first entered

Test Tools. Drivers, or Special Conditions:
Testing via each of the browsers specified in test WEB-WA-001

Input Data:

- Selection of each radio button via mouse input
- Input of valid DOT and ICC/MC numbers
- Input of non-existent DOT and ICC/MC numbers
- Input of alphanumeric entries
- Input of blank (empty) entry

Expected Outcome:

Successful validation of each of the specified requirements

Test Procedure Steps:

After accessing and displaying the SAFER Query Page, validate that each of the specified requirements has been satisfied

Analvsis Procedures: None

Acceptance Test Report

Acceptance ID: SAFER Web
ATP ID: WEB-WQP-001
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: N.D. Crouch Test Date: 11/25/96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The SAFER Query Page consisted of a selection and a data display frame and each was displayed properly
- The radio buttons in the Search Type area of the selection frame operated properly and allowed the user to select either or but not both the DOT and ICC/MC number types
- The ID Number area of the selection frame properly accepted numeric and non-numeric entries with the following results upon query submission:
  - Valid DOT and ICC/MC numbers returned the correct carrier snapshot
  - Invalid DOT and ICC/MC numbers generated a user error message
  - Alpha-numeric entries generated a user error message
  - Empty entries generated a user error message
- The Search button submitted a query to the SAFER/Carrier System based on the value entered in the ID Number area
- The Reset button properly reset, i.e., blanked, the ID Number area
- The explanatory text was properly displayed in the data display frame when the page was first entered

Reviewed By (SAIC): ________________ Date: ______________

Approved By (JHU/APL): ________________ Date: ______________

Acceptance Test Plan

Accep tance ID: Web Query Page
ATP ID: WEB-WQP-002
Version Control ID: APL V01-B02
Affected CSCI: WEB
Acceptance Function:

1. Proper display and function of the SAFER Web Query Page when displaying a carrier snapshot

Requirements to be validated:

. After retrieving a carrier snapshot:
  - the correct snapshot record is returned based on the input query data
  - the snapshot data is displayed properly in the data display frame of the query page
  - the selection frame is displayed properly and is functional, i.e., will accept data for a new query
  - the data display frame is vertically and horizontally scrollable, if necessary

Test Tools. Drivers, or Special Conditions:
Testing via each of the browsers specified in test WEB-WA-001

Input Data: None

Expected Outcome:

Successful validation of each of the specified requirements

Test Procedure Steps:
After submitting and retrieving a valid carrier snapshot, validate that each of the specified requirements has been satisfied

Analysis Procedures: None

Acceptance Test Report

Accep tance ID: Web Query page
ATP ID: WEB-WQP-002
Version Control ID: APL V01-B02
Test Conductor: N.D. Crouch

Test Seq Number: **01**
Test Date: 11/25/96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- After retrieving a carrier snapshot:
  - the correct snapshot record was returned based on the input query data
  - the snapshot data was displayed properly in the data display frame of the query page
  - the selection frame was displayed properly and was functional, i.e., would accept data for a new query
  - the data display frame was vertically and horizontally scrollable, when necessary

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Acceptance Test Plan

Acceptance ID: Web Query Page
ATP ID: WEB-WQP-003
Version Control ID: APL V01-B02
Affected CSCI: WEB
Accep tance Function:

1. Proper display and function of the SAFER Web Query Page data display frame when displaying a carrier snapshot

Reauirements to be validated:

- The ID/Operations I Inspections/Accidents Safety Rating links are displayed properly and are functional, i.e., on selection, positions the user to the appropriate section of the snapshot record
- The carrier information text is displayed properly and the FHWA WWW Site link functions properly, i.e., on selection links to the FHWA home page
- The date when the snapshot record was last updated is displayed properly, is correct and is based on the snapshot's Date of Last Update
- The HELP text is displayed properly and the general help link, when selected, links the **user to** the top of the HELP page
- For the *Carrier, DBA Name, Physical Address, Phone, Mailing Address, USDOT#, ICC/MC Numbers, Entity Type, Power Units, State Carrier ID#, DUNS Number,* **Status,** *Drivers, Work Class, Carrier OP., Shipper Op., Cargo Carried, Hazmat Status, Inspections, and Carrier safety Rating* fields:
  - the field name and its corresponding data, if available, are displayed properly and are correct
  - double-clicking the underscored field name will link the user to the section of the HELP page in which information about the selected field is displayed
- For the non-underscored fields of *Out of Service, Out of Service %, National Average, Fatal, Injury, Tow, and Total Accidents, Rating Date, and Rating*
  - the field name and its corresponding data, if available, are displayed properly and are correct
- Inspections and accidents are reported 24 months prior to the snapshot File Create Date, this date is properly displayed for each type of data and is correct
- The safety rating currency date is displayed properly and is correct, i.e., is based on the File Create Date of the snapshot

Test Tools, Drivers, or Special Conditions:
Testing via each of the browsers specified in test WEB-WA-001

Input Data: None

Expected Outcome:

Successful validation of each of the specified requirements

Test Procedure Steps:
Validate that each of the specified requirements has been satisfied

Analysis Procedures: None

Acceptance Test Report

Accep tance ID: Web Query page
ATP ID: WEB-WQP-003
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: N.D. Crouch Test Date: 11/25/96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The ID/Operations I Inspections/Accidents | Safety Rating links were displayed properly and were functional, i.e., on selection, positioned the user to the appropriate section of the snapshot record
- The carrier information text was displayed properly and the FHWA WWW Site link functioned properly, i.e., on selection linked to the FHWA home page
- The date when the snapshot record was last updated was displayed properly, was correct and was based on the snapshot's Date of Last Update
- The HELP text was displayed properly and the general help link, when selected, linked the user to the top of the HELP page
- For the *Carrier, DBA Name, Physical Address, Phone, Mailing Address, USDOT#, ICC/MC Numbers, Entity Type, Power Units, State Carrier ID#, DUNS Number, Status, Drivers, Work Class, Carrier OP., Shipper Op., Cargo Carried, Hazmat Status, Inspections, and Carrier safety Rating* fields:
  - the field name and its corresponding data, if available, were displayed properly and were correct
  - double-clicking the underscored field name linked the user to the section of the HELP page in which information about the selected field was displayed
- For the non-underscored fields of *Out of Service, Out of Service %, National Average, Fatal, Injury, Tow,* and *Total Accidents, Rating Date, and Rating*
  - the field name and its corresponding data, if available, were displayed properly and were correct
- Inspections and accidents *were* reported 24 months prior to the snapshot File Create Date, this date was properly displayed for each type of data and was correct
- The safety rating currency date was displayed properly and was correct, i.e., was based on the File Create Date of the snapshot

Reviewed By (SAIC): ________________ Date: ______________

Approved By (JHU/APL): ________________

Appendix B

SAFER System

System Test Plans and Test Reports

System Test Plan

System ID: Carrier Snapshot
SYS ID: SYS-CS-001
Version Control ID: APL V01-B02
Affected CSCI: IMH, SDM, OMH
System Function:

1. Detect, accept and process a single mailbox request for a carrier snapshot using a USDOT# as input in pseudo-ED1 format;
2. Send the snapshot to a given single requester's mailbox.

Reauirement to be validated:

- IMH must read a snapshot request from the input mailbox
- IMH must recognize it as a carrier snapshot request
- IMH must parse the ED1 format request to obtain the DOT number
- IMH must fill in the request structure with the DOT number
- IMH must call the appropriate SDM routine to continue processing
- SDM must receive the packet header, request header, and request properly filed in with data types and the desired DOT number.
- SDM must call OMH with the packet-header, request header, and send request properly filled in.
- OMH must receive the proper carrier snapshot request from SDM_retrieve_safety_data.
- OMH must query the database and retrieve a carrier snapshot.
- OMH must send the snapshot to the requester's mailbox in text form.

Test Tools, Drivers, or Special Conditions:

The EudoraLite mail package was used to send the request. The SDB Library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:
An input 61e in EDI format was sent to IMH via the Eudora mail package:

```
ST*285*000010010
BGN*28*54321*95 1117*0800*GM
NM1*41*1*ECKEL*JOHN*MORRIS***57*CSI
NX1*MC
N9*21* 1
LM*FH
```

LQ*T07*Q 106
LM*FH
LQ*T10*Ql
LM*FH
LQ*T10*Q200
REF*MCI*0000004

Expected Outcome:

The carrier snapshot for DOT number 4 should be received in text form in the requester's mailbox.

Test Procedure Steps:

Set up the message in EudoraLite. Send it to the IMH input mailbox.

Read the snapshot for the desired DOT number from the requester's mailbox. Use the SQLPlus utility to view the carrier snapshot in the database and verify that it is the same data received in the mail message.

Analysis Procedures:

•

System Test Report

System ID: Carrier Snapshot
SYS ID: SYS-CS-001
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/l//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The desired snapshot, for DOT number 4, was received in the requester’s mailbox, and was read using the EudoraLite mail package.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Carrier Snapshot
SYS ID: SYS-CS-002
Version Control ID: APL V01-B02
Affected CSCI: IMH, SDM, OMH
System Function:

1. Detect, accept and process five mailbox requests for carrier snapshot using USDOT numbers as input in pseudo-ED1 format;
2. Send results to a single requester's mailbox.

Requirement to be validated:

- IMH must read a snapshot request from the input mailbox
- IMH must recognize it as a carrier snapshot request
- IMH must parse the ED1 format request to obtain the DOT number
- IMH must fill in the request structure with the DOT number
- IMH must call the appropriate SDM routine to continue processing
- IMH must loop through the above sequence of steps four more times provided that five snapshot requests are successively queued in the input mailbox with no other intervening messages.

For each of five successive carrier snapshot requests:

- SDM must receive the packet header, request header, and request properly filled in with data types and the desired DOT number.
- SDM must call OMH with the packet-header, request header, and send request properly filled in.
- SDM must perform the above steps five times for five successive valid carrier snapshot requests forwarded to it by IMH.

For each of five successive requests:

- OMH must receive the proper carrier snapshot request from SDM_retrieve_safety_data.
- OMH must query the database and retrieve a carrier snapshot, provided that the DOT number is a valid one.
- OMH must send the snapshot to the requester's mailbox in text form.
- OMH must loop through the above steps four additional times provided that a total of five carrier snapshots are queued in the input mailbox and sent by IMH to OMH.

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:

Five mail messages, with valid carrier snapshot requests.

Expected Outcome:

The carrier snapshot for the five DOT numbers should be received in text form in the requester's mailbox.

Test Procedure Steps:

Start IMH process.
Set up the messages in Eudora. Send them to the IMH input mailbox.
Read the snapshots for the desired DOT numbers from the requester's mailbox and verify their accuracy.

Analvsis Procedures:

System Test Report

System ID: Carrier Snapshot
SYS ID: SYS-CS-002
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/l//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

The five requested snapshot were received in the requester's mailbox and were read using EudoraLite.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Carrier Snapshot
SYS ID: SYS-CS-003
Version Control ID: APL V01-B02
Affected CSCI: IMH, SDM, OMH
System Function:

1. Detect, accept and process five mailbox requests for a carrier snapshot using USDOT numbers as input in pseudo-ED1 format when five requests are already queued in the input mailbox;
2. Send results to requester's mailbox.

Requirement to be validated:

- IMH must read a snapshot request from the input mailbox
- IMH must recognize it as a carrier snapshot request
- IMH must parse the ED1 format request to obtain the DOT number
- IMH must fill in the request structure with the DOT number
- IMH must call the appropriate SDM routine to continue processing
- IMH must loop through the above sequence of steps a total of ten times when the first five requests are already queued in the input mailbox before IMH is started, and then five additional requests are sent immediately after IMH processing begins.
- SDM must receive the packet header, request header, and request properly filled in with data types and the desired DOT number.
- SDM must call OMH with the packet-header, request header, and send request properly filled in.
- SDM must perform the above steps five times for five successive valid carrier snapshot requests forwarded to it by IMH.
- OMH must receive the proper carrier snapshot request from SDM_retrieve_safety_data.
- OMH must query the database and retrieve a carrier snapshot, provided that the DOT number is a valid one.
- OMH must send the snapshot to the requester's mailbox in text form.
- OMH must loop through the above steps four additional times provided that a total of five carrier snapshots are queued in the input mailbox and sent by IMH to OMH.

Test Tools, Drivers, or Special Conditions:

The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:

Ten mail messages, with valid carrier snapshot requests.

Expected Outcome:
The ten requested carrier snapshots should be received in the requester’s mailbox.

Test Procedure Steps:
Set up the ten messages in EudoraLite. Without starting IMH, send the first five to the IMH input mailbox.
Start the IMH process.
Immediately, use EudoraLite to send the next five messages to the IMH input mailbox.
Read the ten snapshots from the requester’s mailbox. Verify that the information in the text snapshot sent to the mailbox matches the information in the database.

Analysis Procedures:

System Test Report

System ID: Carrier Snapshot
SYS ID: SYS-CS-003
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/l//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The ten snapshots were received in the requester's mailbox for the ten desired USDOT numbers.

Reviewed By (SAIC): ____________ Date: ____________

Approved By (JHU/APL): ____________ Date: ____________

System Test Plan

System ID: Carrier Snapshot
SYS ID: SYS-CS-004
Version Control ID: APL V01-B02
Affected CSCI: IMH. SDM, OMH
System Function:

1. Detect, accept and process a carrier snapshot request using a single digit USDOT #.
2. Detect, accept and process a carrier snapshot request using a seven digit USDOT #.
3. Detect, accept and process a carrier snapshot request using a USDOT # not contained in the database.
4. Detect, accept and process a carrier snapshot request using an erroneous USDOT # containing alphabetic characters.
5. Send results to the requester's mailbox.

Requirement to be validated:

- IMH must read the snapshot request from the input mailbox
- IMH must recognize it as a carrier snapshot request
- IMH must parse the ED1 format request to obtain the DOT number
- IMH must fill in the request structure with the DOT number
- IMH must call the appropriate SDM routine to continue processing
- SDM must receive the packet header, request header, and request with the fields filled in properly for each test request. It must forward the correct packet header, request header, and request to OMH for further processing.
- OMH must receive the proper carrier snapshot request from SDM_retrieve_safety_data.
- OMH must query the database and retrieve a carrier snapshot, provided that the DOT number is a valid one.
- OMH must send the snapshot to the requester's mailbox in text form.

Test Tools, Drivers. or Special Conditions:
The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:
Mail messages, with carrier snapshot requests using a single digit USDOT#, a seven digit USDOT#, a USDOT# not contained in the database, and an erroneous USDOT# containing alphabetic characters.

Expected Outcome:

Carrier snapshots will be successfully retrieved for the single digit and seven digit USDOT numbers and will be received in the requester’s mailbox. For the USDOT number not contained in the database, and the erroneous one containing alphabetic characters, the database retrieval will fail, and a failure message will be received in the requester’s mailbox.

Test Procedure Steps:

Set up the messages in Eudora. Send them to the IMH input mailbox.

Read the four replies from the requester’s mailbox. Verify that carrier snapshots were received for the two valid USDOT numbers and that failure messages were received for the two invalid ones.

Analvsis Procedures:

-

System Test Report

System ID: Carrier Snapshot
SYS ID: SYS-CS-004
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/l//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The single digit USDOT# test case succeeded. The correct carrier snapshot was received in the requester's mailbox.
- The seven digit USDOT# test case succeeded. The correct carrier snapshot was received in the requester's mailbox.
- The USDOT# that did not exist in the database resulted in a failure message.
- The USDOT# containing alphabetic characters resulted in a failure message.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Carrier Snapshot
SYS ID: SYS-CS-005
Version Control ID: APL V01-B02
Affected CSCI: IMH, SDM, OMH
System Function:

1. Process a carrier snapshot request that is pure nonsense (that does not begin with 'R', 'C', or 'S').
2. Process a carrier snapshot request containing 'R' followed by nonsense.
3. Process a carrier snapshot request containing 'C' followed by nonsense.
4. Process a carrier snapshot request containing 's' followed by nonsense.

Requirement to be validated:

- The system must process erroneous incoming requests by rejecting them without crashing, and by informing the requester of an error if appropriate.

Test Tools, Drivers, or Special Conditions:

The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle **v.7.2** database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:

Mail messages, with carrier snapshot requests containing pure nonsense, 'R' followed by nonsense, 'S' followed by nonsense, and 'C' followed by nonsense.

Expected Outcome:

The system will not query the database for a pure nonsense, since it is not a request at all. An invalid file type message should print to the screen, but no mail message will be generated.
The system will attempt to update the database with a 'C' followed by nonsense carrier snapshot update request. Since the update will fail, subscribers will not receive carrier snapshots.
'R' or 'S' followed by nonsense will cause retrieval failure messages to be sent to the requester's mailbox..

Test Procedure Steps:

Set up the messages in Eudora. Send them to the IMH input mailbox.

View output screen for possible error messages.

Check requester's mailbox for retrieval failure messages.

Analvsis Procedures:

-

System Test Report

System ID: Carrier Snapshot
SYS ID: SYS-CS-005
Version Control ID: APL V01-B02
Test Conductor: Grace McGonnigal

Test Seq Number: 01
Test Date: 10/l//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- For the pure nonsense request, a message was printed to the console that it was not a valid message type and the system did not process it further.
- For the ‘C’plus nonsense message, an attempt was made to update the database. It failed. Failure messages were sent to subscribers. This was a bug that was subsequently corrected, because nothing should have been sent to the subscriber list.
- For the ‘R’and ‘S’plus nonsense messages, the retrievals failed and failure messages were sent to the requester’s mailbox.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Carrier Snapshot
SYS ID: SYS CS-006
Version Control ID: APL V01-B02
Affected CSCI: IMH, SDM, OMH
System Function:

1. Send, accept, and process a single mailbox request for a carrier snapshot using a USDOT# as input in Aspen format (D=USDOT#).
2. Send results to requester's mailbox.

Requirement to be validated:

- IMH must read a snapshot request from the input mailbox
- IMH must recognize it as a carrier snapshot request in Aspen format
- IMH must parse the Aspen (D=#) format request to obtain the DOT number
- IMH must fill in the request structure with the DOT number
- IMH must call the appropriate SDM routine to continue processing
- SDM must receive the packet header, request header, and request properly filled in with data types and the desired DOT number.
- SDM calls OMH with the packet header, request header, and send request properly filled in.
- OMH must receive the proper carrier snapshot request from SDM_retrieve_safety_data.
- OMH must query the database and retrieve the snapshot, provided that the DOT number is valid.
- OMH must send the snapshot to the requester's mailbox in text form.

Test Tools, Drivers, or Special Conditions:

The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:

Mail message containing the following two lines:
RCS
D=4

Expected Outcome:

The carrier snapshot for DOT number 4 should be received in text form in the requester's mailbox.

Test Procedure Steps:

Set up the message in Eudora. Send it to the IMH input mailbox.

Read carrier snapshot for the USDOT number in the requester's mailbox.

Analvsis Procedures:

System Test Report

System ID: Carrier Snapshot
SYS ID: SYS CS-006
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/1/96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The correct carrier snapshot was received in the requester's mailbox.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Carrier Snapshot
SYS ID: SYS CS-009
Version Control ID: APL V01-B02
Affected CSCI: IMH, SDM, OMH
System Function:

1. **Send,** accept, and process mailbox requests for a carrier snapshot using USDOT numbers as inputs at various rates to determine system responsiveness.
2. Determine saturation point- first point at which mailbox is filling faster than it is emptying.
3. Before doing this, determine the minimum Sleep statement that will prevent POP3 error messages.
4. Send results to requesters' mailboxes.

Reauirement to be validated:

- IMH must read a snapshot request from the input mailbox
- IMH must fill in the request structure with the DOT number
- IMH must call the appropriate SDM routine to continue processing
- IMH must perform the above steps repeatedly under any reasonable level of system load.
- Spurious POP3 error messages should not be printed to the output screen.
- SDM must receive the packet header, request header, and request properly tilled in with data types and the desired DOT number.
- SDM must call OMH with the packet header, request header, and send request properly filled in.
- OMH must receive the proper carrier snapshot request from SDM_retrieve_safety_data.
- OMH must query the database and retrieve a carrier snapshot, provided that the DOT number is a valid one.
- OMH must send the snapshot to the requester's mailbox in text form.
- Rapidly processing a large number of requests in sequence will not cause a crash or processing overload for OMH.

Test Tools, Drivers, or Special Conditions:

The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

For this test, multiple requesters must be used to stress the system. One requester cannot send messages fast enough with EudoraLite, even if they are set up in advance.

During the test, no other processing should be done on the system.

To further stress the system, one hundred requests were queued in the input mailbox before the IMH process was started. Then three requester's each sent one hundred additional requests at the same time. (Some of the three hundred requests were submitted twice).

Input Data:
Three hundred mail messages containing carrier snapshot requests in ED1 format.

Expected Outcome:

Large numbers of requests will not crash or confuse the system. The system will be able to handle three to four hundred snapshot requests within a period of five to seven minutes.

Test Procedure Steps:

Set up three hundred different carrier snapshot request messages in EudoraLite.
Send one hundred requests to the IMH input mailbox.
Start processing.
Each requester will immediately send the assigned test messages to the IMH input mailbox.
Wait a few minutes for IMH to finish all retrievals.
Each requester will read the responses sent by the system to his/her mailbox and verify that a snapshot was received corresponding to each request submitted.

Analvsis Procedures:

System Test Report

System ID: Carrier Snapshot
SYS ID: SYS CS-009
Version Control ID: APL V01-B02
Test Conductor: Grace McGonnigal

Test Seq Number: 01
Test Date: 10/l//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- A Sleep statement of 225 milliseconds prevented POP3 error messages. 200 milliseconds did not prevent all error messages, but did prevent most, so 225 ms seems to be a good threshold.
- Carrier snapshots were rapidly received for the one hundred requests queued up **before** processing began. Two of the requesters rapidly received carrier snapshots for all requests submitted. The third requester was not using a proper Eudora account to send requests and receive responses. Some of his requests succeeded, but most failed. However, the problem seemed to be within the Eudora setup, not within IMH. IMH did not crash, and perfectly processed all other requests.
- A saturation point was not reached with three people testing. Further testing must **be** done with a larger number of requesters.

Reviewed By (SAIC): ________________ Date: ______________

Approved By (JHU/APL): ________________ Date: ______________

System Test Plan

System ID: Carrier Update
SYS ID: SYS CU-001
Version Control ID: APL V01-B02
Affected CSCI: IMH, SDM, OMH
System Function:

1. Send, accept and process a single mailbox request for carrier snapshot add operation.
2. Use an empty subscriber list (no subscribers).

Requirement to be validated:

- IMH must read a snapshot add request from the input mailbox
- IMH must recognize it as a carrier snapshot add request
- IMH must call the appropriate SDM routine to continue processing and provide it with the correct data types and the database update record itself.
- SDM_update_data must receive a packet header, data header, and update data filled in with accurate information so that it can update the database.
- Since there are no subscribers, messages should not be sent.

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:

One database update request, with valid data, and a USDOT number not yet existing in the database.

Expected Outcome:

The database will be updated with the correct carrier snapshot record. Since there are no subscribers, no snapshot messages will be sent to mailboxes.

Test Procedure Steps:

Send the update message to the IMH input mailbox.
Step through processing in debugger to verify that no attempt is made to mail snapshots to nonexistent subscribers.
After the system processes the message, use SQLPlus to verify that the record was written to the database.

Analysis Procedures:

System Test Report

System ID: Carrier Update
SYS ID: SYS-CU-001
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/l//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The carrier snapshot was correctly written to the database.
- No mail messages were sent to nonexistent subscribers..

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Carrier Update
SYS ID: SYS-CU-002
Version Control ID: APL V01-B02
Affected CSCI: IMH, SDM, OMH
System Function:

1. Send, accept and process a single mailbox request for carrier snapshot add operation.
2. Use a subscriber list containing one and only one valid subscriber.
3. Send copy of updated carrier snapshot to the one subscriber.

Requirement to be validated:

- IMH must read a snapshot add request from the input mailbox
- IMH must recognize it as a carrier snapshot add request
- IMH must call the appropriate SDM routine to continue processing and provide it with the correct data types and the database update record itself.
- SDM_update_data must receive a packet header, data header, and update data filled in with accurate information so that it can update the database.
- SDM must call SLP to forward a snapshot update to the one subscriber.
- OMH will forward a mail message containing a copy of the newly added snapshot to the one subscriber.

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:
One database update request, with valid data, and a USDOT number not yet existing in the database.

Expected Outcome:
The carrier snapshot will be successfully added to the database. The one subscriber will receive a copy of the snapshot.

Test Procedure Steps:
Set up the subscription table in the database to contain just one subscriber.
Mail the update message to the IMH input mailbox.
When the system finishes processing the message, use SQLPlus to verify that the carrier snapshot was added to the database.
Verify that a copy of the carrier snapshot was sent to the subscriber's mailbox.

Analysis Procedures:

System Test Report

System ID: Carrier Update
SYS ID: SYS-CU-002
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/1/96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The carrier snapshot was written to the database.
- A copy of the snapshot was sent to the one subscriber's mailbox.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Carrier Update
SYS ID: SYS-CU-003
Version Control ID: APL V01-B02
Affected CSCI: IMH, SDM, OMH
System Function:

1. Send, accept and process a single mailbox request for carrier snapshot add operation.
2. Use a subscriber list containing at least three subscribers, one of which is invalid.

Requirement to be validated:

- IMH must read a snapshot add request from the input mailbox
- IMH must recognize it as a carrier snapshot add request
- IMH must call the appropriate SDM routine to continue processing and provide it with the correct data types and the database update record itself.
- SDM_update_data must receive a packet header, data header, and update data filled in with accurate information so that it can update the database.
- SDM must call SL_fulfill_subscription to forward a snapshot update to the subscribers.
- OMH will forward a mail message containing a copy of the newly added snapshot to the valid subscribers.
- Having an invalid subscriber on the list will not cause a crash or processing problems.

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:
One database update request, with valid data, and a USDOT number not yet existing in the database.

Expected Outcome:
The carrier snapshot will be successfully added to the database. The valid subscribers will receive copies of the snapshot in their mailboxes. A snapshot message will be sent to the invalid user and the mail system will deal with it as an invalid address.

Test Procedure Steps:
Send carrier snapshot update message to the IMH input mailbox.
After the system finishes processing the message, use SQLPlus to view the new snapshot record in the database.
Verify that the two valid subscribers receive copies of the snapshot in their mailboxes.

Analysis Procedures:

-

System Test Report

System ID: Carrier Update
SYS ID: SYS-CU-003
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/l//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Data structures for packet header, data header, and update data were properly filled in and passed to SDM_update_safety_data.
- The carrier snapshot was added to the database.
- The two valid subscribers received copies of the snapshot in their mailboxes.
- The invalid subscriber case caused a problem due to a bug in MBX, which is being corrected.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Carrier Update
SYS ID: SYS-CU-006
Version Control ID: APL V01-B02
Affected CSCI: IMH, SDM, OMH
System Function:

1. Send, accept and process five separate mailbox requests for carrier snapshot add operations.
2. Use a subscriber list containing at least three subscribers, one of which is invalid.

Requirement to be validated:

- IMH must read a snapshot add request from the input mailbox
- IMH must recognize it as a carrier snapshot add request
- IMH must call the appropriate SDM routine to continue processing and provide it with the correct data types and the database update record itself.
- IMH must be able to perform the above steps five successive times.
- SDM_update_data must receive a packet header, data header, and update data filled in with accurate information so that it can update the database.
- SDM must call SL_fulfill_subscription to forward a snapshot update to the subscribers.
- SDM must perform the above steps five times successively.
- OMH must forward copies of the newly added snapshots to the valid subscribers' mailboxes.
- Having an invalid subscriber on the list will not cause a crash or processing problems.

Test Tools. Drivers, or Special Conditions:

The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and send results to the requester's mailbox.

Input Data:

Five database update requests, with valid data, and USDOT numbers not yet existing in the database.

Expected Outcome:

The five carrier snapshots will be written to the database.
The two valid subscribers will receive copies of the five snapshots in their mailboxes.
The invalid subscriber will not cause processing problems. The mail system will simply handle it like any other invalid address.

Test Procedure Steps:

Send five valid carrier snapshot update requests to the IMH input mailbox.
Verify that the subscription list in the database contains three subscribers, two of which are valid and one of which is invalid.
When processing is completed, verify, using SQLPlus, that the five snapshots were added to the database.
Verify that both valid subscribers received copies of the five snapshots in their mailboxes.
Verify that the invalid subscriber did not cause processing problems.

System Test Report

System ID: Carrier Update
SYS ID: SYS-CU-006
Version Control ID: APL V01-B02
Test Conductor: Grace McGonnigal

Test Seq Number: 01
Test Date: 10/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The five carrier snapshots were correctly added to the database.
- Copies of the five snapshots were received in the two valid subscribers' mailboxes.
- The invalid subscriber case caused a problem due to a bug in MBX, which is being corrected.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Carrier Update
SYS ID: SYS-CT-J-007
Version Control ID: APL V01-B02
Affected CSCI: IMH, SDM, OMH
System Function:

1. Send, accept and process five separate mailbox requests for carrier snapshot add operations when five add requests are already in the queue when IMH processing starts.
2. Use a subscriber list containing at least three subscribers, one of which is invalid.

Requirement to be validated:

- IMH must read a snapshot add request from the input mailbox
- IMH must recognize it as a carrier snapshot add request
- IMH must call the appropriate SDM routine to continue processing and provide it with the correct data types and the database update record itself.
- IMH must be able to process five queued requests immediately upon being initialized, and then perform the above steps five successive times for additional update records.
- SDM_update_data must receive a packet header, data header, and update data filled in with accurate information so that it can update the database.
- SDM must call SL_fulfill_subscription to forward a snapshot update to the subscribers.
- SDM must perform the above steps ten times successively.
- OMH will forward mail messages containing copies of the newly added snapshots to the valid subscribers.
- Having an invalid subscriber on the list will not cause a crash or processing problems.

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verity results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:
Ten database update requests, with valid data, and USDOT numbers not yet existing in the database.

Expected Outcome:
The ten carrier snapshots will be written to the database.
The two valid subscribers will receive copies of the ten snapshots in their mailboxes.
The invalid subscriber will not cause processing problems. The mail system will simply handle it like any other invalid address.

Test Procedure Steps:

Send ten valid carrier snapshot update requests to the IMH input mailbox.

Verify that the subscription list in the database contains three subscribers, two of which are valid and one of which is invalid.

When processing is completed, verify, using SQLPlus, that the ten snapshots were added to the database.

Verify that both valid subscribers received copies of the five snapshots in their mailboxes.

Verify that the invalid subscriber did not cause processing problems.

Analvsis Procedures:

System Test Report

System ID: Carrier Update
SYS ID: SYS-CU-007
Version Control ID: APL V01-B02
Test Conductor: Grace McGonnigal

Test Seq Number: 01
Test Date: 10/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The ten carrier snapshots were correctly added to the database.
- Copies of the ten snapshots were received in the two valid subscribers' mailboxes.
- The invalid subscriber case caused a problem due to a bug in MBX, which is being corrected.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Accounting
SYS ID: SYS-AC-001
Version Control ID: APL V01-B02
Affected CSCI: ADM/ACT, IMH, SDM
System Function:

1. Send, accept, and process a single request to create an organization record in the database.
2. The message should identify an organization not yet contained in the database.
3. The message should add one or more initial users to the database as part of the organization.

Reauirement to be validated:

- IMH_read_account_request must correctly parse the create organization message received in the input mailbox, and copy each piece of information contained in the message into the appropriate structures for further processing.
- ADM_ACCT_create_erg_account must verify that the requester has the privileges to create the organization record.
- ADM_ACCT_create_erg_account must create the organization record with a unique identification label.
- ADM_ACCT_create_user_account must add any users identified in the create organization message to the database as a part of that organization.

Test Tools, Drivers, or Special Conditions:

The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:
A valid request to create one organization with two initial users was used as input.

Expected Outcome:
The organization record and two user records will be correctly written to the database.

Test Procedure Steps:
Set up the message in EudoraLite. Send it to the IMH input mailbox.

When processing is completed, use the SQLPlus utility to view the organization record and the two user records in the database.

<u>Analysis Procedures:</u>

System Test Report

System ID: Accounting
SYS ID: SYS-AC-001
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The organization record was written to the database and filled in with the proper information.
- The two user records were written to the database and were filled in properly.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Accounting
SYS ID: SYS-AC-002
Version Control ID: APL V01-B02
Affected CSCI: ADM/ACT, IMH, SDM
System Function:

1. Send, accept, and process a request to add one or more users to the database as part of an existing organization.
2. Verity that the users were correctly entered into the database.

Requirement to be validated:

- IMH_read_account_request must correctly parse the create user message received in the input mailbox, and copy each piece of information contained in the message into the appropriate structures for further processing.
- ADM_ACCT_create_user must verify that the user's organization already exists in the database.
- ADM_ACCT_create_user must verify that the requester is valid and is associated with the organization.
- ADM_ACCT_update_user_account must verify that the requester has the privileges to add the user record to the database.
- ADM_ACCT_create_user must verify that the user does not already exist in the database as part of that organization.
- ADM_ACCT_create_user must correctly add the user to the database.

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:
The input was a valid create user message containing data for two new users for a previously existing organization. The message contained a valid requester id for that organization.

Expected Outcome:
Records for the two new users will be correctly written to the database.

Test Procedure Steps:
Set up the message in EudoraLite. Send it to the IMH input mailbox.

When processing is completed, use SQLPlus to verify that the new user records were correctly entered in the database.

Analysis Procedures:

-

System Test Report

System ID: Accounting
SYS ID: SYS-AC-002
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Records for the two new users were correctly added to the database.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Accounting
SYS ID: SYS-AC-003
Version Control ID: APL V01-B02
Affected CSCI: ADM/ACT, IMH, SDM
System Function:

1. Send, accept, and process a request to update an organization already existing in the database
2. The message will contain a valid requester id and the requester will have the necessary privileges.

Requirement to be validated:

- IMH_read_account_request must correctly parse the update organization message received in the input mailbox, and copy each piece of information contained in the message into the appropriate structures for further processing.
- ADM_ACCT_update_org_account must verify that the organization already exists in the database.
- ADM_ACCT_update_org_account must verify that the requester is valid and is associated with the organization.
- ADM_ACCT_update_user_account must verify that the requester has the privileges to update the organization record.
- ADM_ACCT_update_org_account must correctly make the requested modifications to the organization record.

Test Tools, Drivers, or Special Conditions:

The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:
Input was a message containing an existing organization, a valid requester id, and a request to modify the telephone number and part of the mailing address.

Expected Outcome:
The telephone number and mailing address will be updated for that organization.

Test Procedure Steps:

Set up the message in EudoraLite. Send it to the IMH input mailbox.

Analysis Procedures:

-

System Test Report

System ID: Accounting
SYS ID: SYS-AC-003
Version Control ID: APL V01-B02
Test Seq Number: 01
Test Conductor: Grace McGonnigal
Test Date: 10/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The organization record was modified as requested in the database.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Accounting
SYS ID: SYS-AC-004
Version Control ID: APL V01-B02
Affected CSCI: ADM/ACT, IMH, SDM
System Function:

1. Send, accept, and process a message containing a request to update user information.
2. The input message should contain a valid organization, requester, and user.

Reauirement to be validated:

- IMH_read_account_request must correctly parse the update organization message received in the input mailbox, and copy each piece of information contained in the message into the appropriate structures for further processing.
- ADM_ACCT_update_user_account must verify that the user's organization already exists in the database.
- ADM_ACCT_update_user_account must verify that the requester is valid and is associated with the organization.
- ADM_ACCT_update_user_account must verify that the requester has the privileges to update the user record.
- ADM_ACCT_update_user_account must verify that the user already exists in the database.
- ADM_ACCT_update_user_account must make the requested modifications to the user record.

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and send results to the requester's mailbox.

Input Data:
The input was a message requesting that the telephone number, extension, and city be changed for one user, and that the country be changed for another user.

Expected Outcome:
All requested changes will be made to the two users records in the database.

Test Procedure Steps:
Use SQLPlus to view the records for the two users. Verify that the information in their fields is different from the information in the update user request.
Set up the message in EudoraLite. Send it to the IMH input mailbox.

When processing is completed, use the SQLPlus utility to verify that the records for the two users have changed appropriately.

Analysis Procedures:

System Test Report

System ID: Accounting
SYS ID: SYS-AC-004
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The requested changes were made to the two user records.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Accounting
SYS ID: SYS-AC-005
Version Control ID: APL V01-B02
Affected CSCI: ADM/ACT, IMH, SDM
System Function:

1. Send, accept, and process a request to delete a user.
2. The request message will contain a user id existing in the database, and a valid requester.
3. The request message will contain the correct organization id for the user.

Requirement to be validated:

- IMH_read_account_request must correctly parse the delete user message received in the input mailbox, and copy each piece of information contained in the message into the appropriate structures for further processing.
- ADM_ACCT_delete_user_account must verify that the requester has the privileges to delete the user record.
- ADM_ACCT_delete_user_account must verify that the requester is not deleting his own record.
- ADM_ACCT_delete_user_account must remove the users identified from the database.

Test Tools, Drivers, or Special Conditions:

The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:

Input consisted of a delete user request that specified one valid user in a valid organization. The requester was also valid, and had the privileges to delete users.

Expected Outcome:

The user will be deleted.

Test Procedure Steps:

Using the SQLPlus utility, verify that the user exists in the database.
Set up the message in EudoraLite. Send it to the IMH input mailbox.
Using the SQLPlus utility, verify that the user no longer exists in the database.

Analysis Procedures:

System Test Report

System ID: Accounting
SYS ID: SYS-AC-005
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The correct user was deleted from the database.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Accounting
SYS ID: SYS-AC-006
Version Control ID: APL V01-B02
Affected CSCI: ADM/ACT, IMH, SDM
System Function:

1. Send, accept, and process a request to delete an organization.
2. The message will contain an organization identifier existing in the database, and will have a valid requester with the privileges to delete an organization.

Requirement to be validated:

- IMH_read_account_request must correctly parse the create organization message received in the input mailbox, and copy each piece of information contained in the message into the appropriate structures for further processing.
- ADM_ACCT_delete_org_account must verify that the requester has the privileges to delete the organization record.
- ADM_ACCT_delete_org_account must verify the organization exists in the database.
- ADM_ACCT_delete_org_account must delete the organization.

Test Tools, Drivers, or Special Conditions:

The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:

Input was a message containing a valid organization identifier and a valid requester.

Expected Outcome:

The organization record will be deleted from the database.

Test Procedure Steps:

Using SQLPlus, verify that the organization exists in the database.
Set up the message in EudoraLite. Send it to the IMH input mailbox.
Using SQLPlus, verify that the organization no longer exists in the database.

Analysis Procedures:

-

System Test Report

System ID: Accounting
SYS ID: SYS-AC-006
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The requested organization was deleted from the database.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Subscription List
SYS ID: SYS-SL-001
Version Control ID: APL V01-B02
Affected CSCI: SLP, IMH, SDM
System Function:

1. Send, accept, and process a single request to create a subscription record in the database.
2. The subscription request message should be for a subscriber not yet included in the database.
3. At least two USDOT numbers should be entered in the parameter list.

Requirement to be validated:

- IMH_read_subscription_request must correctly parse the create subscription message received in the input mailbox, and copy each piece of information contained in the message into the appropriate structures for further processing.
- SLP_create must check whether the subscription already exists in the database.
- SLP_create must create the subscription using the sender's identifier as the user id.
- SLP_create must correctly add the USDOT numbers to the subscriber parameter table in the database.

Test Tools, Drivers, or Special Conditions:

The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:
A valid request to add a subscriber was used as input. Two USDOT numbers were placed in the parameter list.

Expected Outcome:
The subscriber will be added to the database.

Test Procedure Steps:
Set up the message in EudoraLite. Send it to the IMH input mailbox.
When processing is completed, use the SQLPlus utility to view the subscription record.

Analysis Procedures:

System Test Report

System ID: Subscription List
SYS ID: SYS-SL-001
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The subscriber was correctly entered in the database subscription table.
- The subscriber's selected USDOT numbers were correctly entered into the subscriber parameter list.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Subscription List
SYS ID: SYS-SL-002
Version Control ID: APL V01-B02
Affected CSCI: SLP, IMH, SDM, OMH
System Function:

1. Send, accept, and process a single request to add a carrier snapshot to the database.
2. The subscription table in the database should contain at least one valid subscriber who will receive a copy of the snapshot.

Requirement to be validated:

- IMH_read_subscription_request must correctly parse the create subscription message received in the input mailbox, and copy each piece of information contained in the message into the appropriate structures for further processing.
- SLP_monitor must read the subscription table.
- SLP_monitor must call SDM to extract the new snapshot from the database and send copies to all subscribers.
- SLP_create must check whether the subscription already exists in the database.
- SLP_create must create the subscription using the sender's identifier as the user id.

Test Tools, Drivers, or Special Conditions:

The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:
A valid request to add a carrier snapshot was used as input.

Expected Outcome:
The carrier snapshot will be added to the database.
All valid subscribers in the subscription table of the database will receive copies of the carrier snapshot.

Test Procedure Steps:
Set up the message in EudoraLite. Send it to the IMH input mailbox.
When processing is completed, use the SQLPlus utility to view the new carrier snapshot record in the database.
Read each subscribers mailbox to verify that a copy of the carrier snapshot was received.

Analysis Procedures:

-

System Test Report

System ID: Subscription List
SYS ID: SYS-SL-002
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 10/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The carrier snapshot was added to the database.
- Each subscriber received a copy of the carrier snapshot.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

System Test Plan

System ID: Subscription List
SYS ID: SYS-SL-003
Version Control ID: APL V01-B02
Affected CSCI: SLP, IMH, SDM, OMH
System Function:

1. Send, accept, and process a single request retrieve a carrier snapshot and add the USDOT number to the requester's parameter list.
2. The message should contain a valid USDOT number contained in the database.
3. The requester should already be listed as a subscriber in the subscription table, and should have one or more USDOT numbers already set up in the subscriber parameters table. The requested USDOT number should not yet be in the subscriber parameters table, however.
4. The user-seq value used in the message must match that in the subscription record.

Requirement to be validated:

- IMH_read_subscription_request must correctly parse the create subscription message received in the input mailbox, and copy each piece of information contained in the message into the appropriate structures for further processing.
- IMH_translate_message must recognize the input message as a request to retrieve the snapshot and append the USDOT number to the subscriber's parameter list.
- SLP_create must check whether the subscription already exists in the database.
- SLP_create must call SLP_append to append the requested USDOT number to the subscriber's parameter list.

Test Tools, Drivers, or Special Conditions:

The EudoraLite mail package was used to send the request. The SDB library of SQL statements embedded in C code was used to retrieve the snapshot from the Oracle v.7.2 database. The SQLPlus utility was used to view the database and verify results. The MBX library, based on the Distinct mail protocol software, was used to extract the message from the mailbox for processing and to forward results to the requester's mailbox.

Input Data:

A valid request to retrieve a carrier snapshot and add the USDOT number to the subscriber's parameter list was used. The format was as follows:

```
R
A=111
PTestl
```

where PTestl is the user-seq value.

Expected Outcome:

A copy of the carrier snapshot will be sent to the requester's mailbox.
The USDOT number will be added to the requester's parameter list.

Test Procedure Steps:

Set up the message in EudoraLite. Send it to the IMH input mailbox.
When processing is completed, use the SQLPlus utility to view the subscriber parameters table and verify that the requested USDOT number has been added.
Read the requester's mailbox to verify that a copy of the carrier snapshot was received.

Analysis Procedures:

System Test Report

System ID: Subscription List
SYS ID: SYS-SL-003
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 11/5//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The USDOT number was added to the requester's subscriber parameter list in the database.
- The requester received a copy of the carrier snapshot.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Appendix C

SAFER System Integration Test Plans and Test Reports

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: IMH-CS-001
Version Control ID: APL V01-B02
Affected CSCI: IMH
Integration Function:

1. Detect, accept and process a single mailbox request for a carrier snapshot using a USDOT## as input in pseudo-ED1 format;
2. Send results to SDM.

Requirement to be validated:

- IMH must read a snapshot request from the input mailbox
- IMH must recognize it as a carrier snapshot request
- IMH must parse the EDI format request to obtain the DOT number
- IMH must fill in the request structure with the DOT number
- IMH must call the appropriate SDM routine to continue processing

Test Tools. Drivers, or Special Conditions:

The EudoraLite mail package was used to send the request. The MBX_Read routine was used to extract a message from the mailbox.

Input Data:

An input file in EDI format was sent to IMH via the Eudora mail package:

```
ST*285*000010010
BGN*28*54321*951117*0800*GM
NM1*41*l*ECKEL*JOHN*MORRIS***57*CSI
NXl*MC
N9*2I* 1
LM*FH
LQ*T07*Q 106
LM*FH
LQ*T10*Ql
LM*FH
LQ*T10*Q200
REF*MCI*0000004
```

Expected Outcome:

IMH will call SDM_retrieve_safety_data. with the packet header request type correctly filled with SAF_TRANS_DATA_REQUEST. The request structure will contain the correct DOT number in the query criteria.
The carrier snapshot for DOT number 4 should be received in text form in the requester's mailbox.

Test Procedure Steps:
Set up the message in Eudora. Send it to the IMH input mailbox.

Check packet header, request header, and request passed to SDM_retrieve_safety_data. to verify that data types and DOT number are set correctly.

Analvsis Procedures:

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: IMH-CS-001
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1/96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Data structures for packet header, request header, and request were properly filled in and passed to SDM_retrieve_safety_data for the request.

Reviewed By (SAIC): __________________ Date: ___________

Approved By (JHU/APL): __________________ Date: ___________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: SDM-CS-001
Version Control ID: APL V01-B02
Affected CSCI: SDM
Integration Function:

1. Detect, accept and process a single carrier snapshot request using a USDOT# as input
2. Send results to OMH.

Requirement to be validated:

- SDM receives the packet header, request header, and request properly filled in with data types and the desired DOT number.
- SDM calls OMH with the packet_header, request header, and send request properly filled in.

Test Tools, Drivers, or Special Conditions:
None.

Input Data:
Packet header, request header, and request are received as input.

Expected Outcome:

OMH is called with data structures properly filled in, and is equipped with the information needed to query the database and send results back to the requester.

Test Procedure Steps:

After a valid carrier snapshot request is sent via Eudora to IMH, verify that SDM_retrieve_safety_data receives the request with the correct return address, data type and DOT number.

Verify that OMH_send_safety_data is called with data structures filled in with the correct return address, data type and DOT number.

Analysis Procedures:

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: SDM-CS-001
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- SDM_retrieve_safety_data received all information necessary to complete the retrieval.
- OMH_send_safety_data was given all information necessary to complete the retrieval.

Reviewed By (SAIC): __________________ Date: ___________

Approved By (JHU/APL): __________________ Date: ___________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: OMH-CS-001
Version Control ID: APL VOLB02
Affected CSCI: OMH
Integration Function:

1. Detect, accept and process a single carrier snapshot request using a USDOT# as input
2. Send results to a single requester's mailbox.

Requirement to be validated:

- OMH must receive the proper carrier snapshot request from SDM_retrieve_safety_data.
- OMH must query the database and retrieve a carrier snapshot, provided that the DOT number is a valid one.
- OMH must send the snapshot to the requester's mailbox in text form.

Test Tools, Drivers, or Special Conditions:
The MBX_Send routine was used to send the snapshot back to the requester's mailbox. SDB_get_view was called to retrieve the snapshot from the database.

Input Data:

The packet header, request header, and send request are received by OMH as inputs.

Expected Outcome:

OMH will successfully query the database, retrieve the requested snapshot, and send it to the requester's mailbox.

Test Procedure Steps:

Verify that OMH_send_safety_data receives the packet header, request header, and send request properly filled in.

Verify that OMH_output_query_database successfully calls SDB to retrieve the snapshot.

Verify that OMH_output_message__generator successfully calls MBX_Send to forward the snapshot to the requester.

Read the snapshot for the desired DOT number from the requester's mailbox.

Analysis Procedures:

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: OMH-CS-001
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The desired snapshot, for DOT number 4, was received in the requester's mailbox, and was read using the EudoraLite mail package.

Reviewed By (SAIC): __________________ Date: ___________

Approved By (JHU/APL): __________________ Date: ___________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: IMH-CS-002
Version Control ID: APL V01-B02
Affected CSCI: IMH
Integration Function:

1. Detect, accept and process five mailbox requests for carrier snapshot using USDOT numbers as input in pseudo-EDI format;
2. Send results to SDM.

Requirement to be validated:

- IMH must read a snapshot request from the input mailbox
- IMH must recognize it as a carrier snapshot request
- IMH must parse the EDI format request to obtain the DOT number
- IMH must fill in the request structure with the DOT number
- IMH must call the appropriate SDM routine to continue processing
- IMH must loop through the above sequence of steps four more times provided that five snapshot requests are successively queued in the input mailbox with no other intervening messages.

Test Tools, Drivers, or Special Conditions:
The EudoraLite marl package was used to send the requests to the input mailbox. The MBX_Read routine was used to extract messages from the mailbox.

Input Data:
Five mail messages, with valid carrier snapshot requests.

Expected Outcome:

IMH will call SDM_retrieve_safety_data five successive times with the packet header request type correctly filled with SAF_TRANS_DATA_REQUEST. The request structure will contain the correct DOT number in the query criteria.
The carrier snapshot for the five DOT numbers should be received in text form in the requester's mailbox.

Test Procedure Steps:

Start IMH process.
Set up the messages in Eudora. Send them to the IMH input mailbox.
Check packet header, request header, and request passed to SDM_retrieve_safety_data to verify that data types and DOT number are set correctly for each of the five messages.

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: IMH-CS-002
Version Control ID: APL V01=B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Data structures for packet header, request header, and request were properly filled in and passed to SDM_retrieve_safety_data for each of the five messages.

Reviewed By (SAIC): ___________________ Date: ___________

Approved By (JHU/APL): ___________________ Date: ___________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: SDM-CS-002
Version Control ID: APL V01-B02
Affected CSCI: SDM
Integration Function:

- Detect, accept and process five carrier snapshot requests using USDOT numbers as input;
- Send results to OMH.

Requirement to be validated:

For each of five successive carrier snapshot requests:

- SDM receives the packet header, request header, and request properly filled in with data types and the desired DOT number.
- SDM calls OMH with the packet_header, request header, and send request properly filled in.
- SDM performs the above steps five times for five successive valid carrier snapshot requests forwarded to it by IMH.

Test Tools, Drivers, or Special Conditions:
None

Input Data:
Packet header, request header, and request are received as input.

Expected Outcome:
OMH is called with data structures properly filled in, and is equipped with the information needed to query the database and send results back to the requester.

Test Procedure Steps:

For each of the five carrier snapshot requests forwarded from IMH to SDM:

Verify that SDM_retrieve_safety_data receives the request with the correct return address, data type and DOT number.
Verify that OMH_send_safety_data is called with data structures filled in with the correct return address, data type and DOT number.

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: SDM-CS-002
Version Control ID: APL VOLB02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

For each of the five requests:

- SDM_retrieve_safety_data received all information necessary to complete the retrieval.
- OMH_send_safety_data was given all information necessary to complete the retrieval.

Reviewed By (SAIC): __________________ Date: ___________

Approved By (JHU/APL): __________________ Date: ___________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: OMH-CS-002
Version Control ID: APL V01-B02
Affected CSCI: OMH
Integration Function:

- Detect, accept and process five carrier snapshot requests using USDOT numbers as input
- Send results to a single requester's mailbox.

Requirement to be validated:

For each of five successive requests:

- OMH must receive the proper carrier snapshot request from
- SDM_retrieve_safety_data.
- OMH must query the database and retrieve a carrier snapshot, provided that the DOT number is a valid one.
- OMH must send the snapshot to the requester's mailbox in text form.
- OMH must loop through the above steps four additional times provided that a total of five carrier snapshots are queued in the input mailbox and sent by IMH to OMH.

Test Tools, Drivers, or Special Conditions:
The MBX_Send routine was used to send the snapshot back to the requester's mailbox. SDB_get_view was called to retrieve the snapshot from the database.

Input Data:
Five successive sets of packet header, request header, and send request are received by OMH as inputs.

Expected Outcome:
OMH will successfully query the database, retrieve the requested snapshot, and send it to the requester's mailbox for each of the five requests.

Test Procedure Steps:

For each of the five requests:

Verify that OMH_send_safety_data receives the packet header, request header, and send request properly filled in.
Verify that OMH_output_query_database successfully calls SDB to retrieve the snapshot.
Verify that OMH_output_message_generator successfully calls MBX_Send to forward the snapshot to the requester.

Read the snapshot for the desired DOT number from the requester's mailbox.

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: OMH-CS-002
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The five desired snapshots were received in the requester's mailbox, and were read using the EudoraLite mail package.

Reviewed By (SAIC): __________________ Date: ___________

Approved By (JHU/APL): _________________ Date: __________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: IMH-CS-003
Version Control ID: APL V01=B02
Affected CSCI: IMH
Integration Function:

1. Detect, accept and process five mailbox requests for a carrier snapshot using USDOT numbers as input in pseudo-EDI format when five requests are already queued in the input mailbox;
2. Send results to SDM.

Requirement to be validated:

- IMH must read a snapshot request from the input mailbox
- IMH must recognize it as a carrier snapshot request
- IMH must parse the EDI format request to obtain the DOT number
- IMH must fill in the request structure with the DOT number
- IMH must call the appropriate SDM routine to continue processing
- IMH must loop through the above sequence of steps a total of ten times when the first five requests are already queued in the input mailbox before IMH is started, and then five additional requests are sent immediately after IMH processing begins.

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the requests to the input mailbox.
The MBX_Read routine was used to extract a message from the mailbox.

Input Data:
Ten mail messages, with valid carrier snapshot requests.

Expected Outcome:

IMH will call SDM_retrieve_safety_data ten successive times with the packet header request type correctly filled with SAF_TRANS_DATA_REQUEST. The request structure will contain the correct DOT number in the query criteria.
The carrier snapshot for the ten DOT numbers should be received in text form in the requester's mailbox.

Test Procedure Steps:
Set up the ten messages in Eudora. Without starting IMH, send the first five to the IMH input mailbox.
Start the IMH process.
Immediately, use Eudora to send the next five messages to the IMH input mailbox.

For each message, check packet header, request header, and request passed to SDM_retrieve_safety_data to verify that data types and DOT number are set correctly.

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: IMH-CS-003
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Data structures for packet header, request header, and request were properly filled in and passed to SDM_retrieve_safety_data for all ten messages.

Reviewed By (SAIC): __________________ Date: __________

Approved By (JHU/APL): __________________ Date: __________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: SDM-CS-003
Version Control ID: APL V01=B02
Affected CSCI: SDM
Integration Function:

1. Detect, accept and process five carrier snapshot requests using USDOT#'s as input when five messages are already queued in the mailbox before the IMH process is started;
2. Send results to OMH.

Requirement to be validated:

If IMH processing is started with five carrier snapshot requests already queued in the mailbox, and then five additional requests are immediately sent, SDM will do the following for each:

- SDM receives the packet header, request header, and request properly filled in with data types and the desired DOT number.
- SDM calls OMH with the packet_header, request header, and send request properly filled in.
- SDM performs the above steps five times for five successive valid carrier snapshot requests forwarded to it by IMH.

Test Tools, Drivers, or Special Conditions:
None

Input Data:
Packet header, request header, and request are received as input for each of the ten requests.

Expected Outcome:
OMH is called with data structures properly filled in, and is equipped with the information needed to query the database and send results back to the requester for the ten DOT numbers.

Test Procedure Steps:

For each of the ten carrier snapshot requests forwarded from IMH to SDM:

Verify that SDM,_retrieve_safety_data receives the request with the correct return address, data type and DOT number.

Verify that OMH_send_safety_data is called with data structures filled in with the correct return address, data type and DOT number.

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: SDM-CS-003
Version Control ID: APL VOLB02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

For each of the ten requests:

- SDM_retrieve_safety_data received all information necessary to complete the retrieval.
- OMH_send_safety_data was given all information necessary to complete the retrieval.
- Ten correct snapshots were received by the requester.

Reviewed By (SAIC): __________________ Date: __________

Approved By (JHU/APL): __________________ Date: __________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: OMH-CS-003
Version Control ID: APL V01-B02
Affected CSCI: OMH
Integration Function:

1. Detect, accept and process five carrier snapshot reequests using USDOT numbers a input when five requests are already queued in the mailbox before IMH process is started.
2. Send results to a single requester's mailbox.

Requirement to be validated:

For each of the ten requests:

- OMH must receive the proper carrier snapshot request from SDM_retrieve_safety_data.
- OMH must query the database and retrieve a carrier snapshot, provided that the DOT number is a valid one.
- OMH must send the snapshot to the requester's mailbox in text form.
- OMH must loop through the above steps four additional times provided that a total of five carrier snapshots are queued in the input mailbox and sent by IMH to OMH.

Test Tools, Drivers, or Special Conditions:
The MBX_Send routine was used to send the snapshot back to the requester's mailbox. SDB_get_view was called to retrieve the snapshot from the database.

Input Data:
Ten successive sets of packet header, request header, and send request are received by OMH as inputs.

Expected Outcome:
OMH will successfully query the database, retrieve the requested snapshot, and send it to the requester's mailbox for each of the ten requests.

Test Procedure Steps:
For each of the ten requests:
Verify that OMH_send_safety_data receives the packet header, request header, and send request properly filled in.
Verify that OMH_output_query_database successfully calls SDB to retrieve the snapshot.
Verify that OMH_output_message_generator successfully calls MBX_Send to forward the snapshot to the requester.
Read the snapshot for the desired DOT number from the requester's mailbox.

Analysis Procedures:

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: OMH-CS-003
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/l//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The ten desired snapshots were received in the requester's mailbox, and were read using the EudoraLite mail package.

Reviewed By (SAIC): ___________________ Date: ________________

Approved By (JHU/APL): ___________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: IMH-CS-004
Version Control ID: APL V01-B02
Affected CSCI: IMH
Integration Function:

1. Detect, accept and process a carrier snapshot request using a single digit USDOT #.
2. Detect, accept and process a carrier snapshot request using a seven digit USDOT #.
3. Detect, accept and process a carrier snapshot request using a USDOT # not contained in the database.
4. Detect, accept and process a carrier snapshot request using an erroneous USDOT # containing alphabetic characters.
5. Send results to SDM.

Requirement to be validated:

- IMH must read the snapshot request from the input mailbox
- IMH must recognize it as a carrier snapshot request
- IMH must parse the ED1 format request to obtain the DOT number
- IMH must fill in the request structure with the DOT number
- IMH must call the appropriate SDM routine to continue processing

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the requests to the input mailbox. The MBX_Read routine was used to extract a message from the mailbox.

Input Data:
Mail messages, with carrier snapshot requests using a single digit USDOT#, a seven digit USDOT#, a USDOT# not contained in the database, and an erroneous USDOT# containing alphabetic characters.

Expected Outcome:
IMH will call SDM_retrieve_safety_data. ten successive times with the packet header request type correctly filled with SAF_TWS_DATA_REQUEST. The request structure will contain the correct DOT number in the query criteria.
The carrier snapshot for the ten DOT numbers should be received in text form in the requester's mailbox.

Test Procedure Steps:
Set up the messages in Eudora. Send them to the IMH input mailbox.

For each message, check packet header, request header, and request passed to SDM_retrieve_safety_data to verify that data types and DOT numbers are set correctly.

(For the DOT number containing alphabetic characters, it obviously cannot be set correctly but will probably have been converted to a zero).

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: IMH-CS-004
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The single digit USDOT# test case succeeded.
- The seven digit USDOT# test case succeeded.
- The USDOT# that did not exist in the database resulted in a failure message.
- The USDOT# containing alphabetic characters resulted in a failure message.
- For all test cases, the request forwarded to SDM had the structures filled in correctly, with the exception of the USDOT # containing alphabetic characters. For that one, the ascii to long integer conversion failed, returned a zero, placed zero in the structure as the carrier id, and resulted in retrieval failure, as would be expected. IMH did not crash.

Reviewed By (SAIC): ________________ Date: ______________

Approved By (JHU/APL): ________________ Date: ______________

Integration Test Plan

1ntegration ID: Carrier Snapshot
ITP ID: SDM-CS-004
Version Control ID: APL V01-B02
Affected CSCI: SDM
Integration Function:

1. Detect, accept and process a carrier snapshot request using a single digit USDOT #.
2. Detect, accept and process a carrier snapshot request using a seven digit USDOT #.
3. Detect, accept and process a carrier snapshot request using a USDOT # not contained in the database.
4. Detect, accept and process a carrier snapshot request using an erroneous USDOT # containing alphabetic characters.
5. Send results to OMH.

Requirement to be validated:

- SDM must receive the packet header, request header, and request with the fields filled in properly for each test request. It must forward the correct packet header, request header, and request to OMH for further processing.

Test Tools, Drivers, or Special Conditions:
None.

Input Data:
Packet headers, request headers, and requests filled in correctly, with carrier snapshot requests single digit USDOT#, a seven digit USDOT#, a USDOT# not contained in the database, and an erroneous USDOT# containing alphabetic characters.

Expected Outcome:

For each request, SDM will call OMH_send_safety_data with the data type and USDOT# set correctly. (In case of the erroneous USDOT# with alphabetic carriers, conversion to zero will be considered a desirable result).

Test Procedure Steps:

For each message, check packet header, request header, and request received by SDM_retrieve_safety_data. have data types and DOT numbers set correctly. (For the DOT number containing alphabetic characters, it obviously cannot be set correctly but will probably have been converted to a zero). Similarly, verify that packet header, request header, and request have the correct settings when passed to OMH_send_safety_data for each message.

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: SDM-W-004
Version Control ID: APL V01-B02
Test Conductor: Grace McGonnigal

Test Seq Number: 01
Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The single digit USDOT# test case succeeded.
- The seven digit USDOT# test case succeeded.
- The USDOT# that did not exist in the database resulted in a failure message.
- The USDOT# containing alphabetic characters resulted in a failure message.
- For all test cases, the request received by SDM had the structures filled in correctly, with the exception of the USDOT # containing alphabetic characters. For that one, the ascii to long integer conversion failed, returned a zero, placed zero in the structure as the carrier id.
- For all test cases, the packet header, request header, and send request forwarded to OMH had the data types and USDOT numbers set correctly, or to zero in the case of the alphabetic DOT number.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHUAPL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: OMH-CS-004
Version Control ID: APL V01-B02
Affected CSCI: OMH
Integration Function:

1. Detect, accept and process a carrier snapshot request using a single digit USDOT #.
2. Detect, accept and process a carrier snapshot request using a seven digit USDOT #.
3. Detect, accept and process a carrier snapshot request using a USDOT # not contained in the database.
4. Detect, accept and process a carrier snapshot request using an erroneous USDOT # containing alphabetic characters.
5. Send results to the requester's mailbox.

Requirement to be validated:

- OMH must receive the proper carrier snapshot request from SDM_retrieve_safety_data.
- OMH must query the database and retrieve a carrier snapshot, provided that the DOT number is a valid one.
- OMH must send the snapshot to the requester's mailbox in text form.

Test Tools, Drivers, or Special Conditions:
None.

Input Data:
Packet headers, request headers, and send requests filled in correctly, with carrier snapshot requests single digit USDOT#, a seven digitUSDOT#, a USDOT# not contained in the database, and an erroneous USDOT# containing alphabetic characters.

Expected Outcome:
For each test message, OMH will successfully query the database, retrieve the requested snapshot, and send it to the requester's mailbox.

Test Procedure Steps:

For each test message:

Verify that OMH_send_safety_data receives the packet header, request header, and send request properly filled in.

Verify that OMH_output_query_database successfully calls SDB to retrieve the snapshot.

Verify that OMH_output_message_generater successfully calls MBX_Send to forward the snapshot to the requester.

Read the snapshot for the desired DOT number from the requester's mailbox.

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: OMH-CS-004
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The single digit USDOT# test case succeeded.
- The seven digit USDOT# test case succeeded.
- The USDOT# that did not exist in the database resulted in a failure message.
- The USDOT# containing alphabetic characters resulted in a failure message.
- For all test cases, the request received by SDM had the structures filled in correctly, with the exception of the USDOT # containing alphabetic characters. For that one, the ascii to long integer conversion failed, returned a zero, placed zero in the structure as the carrier id.
- For all test cases, the packet header, request header, and send request received by OMH had the data types and USDOT numbers set correctly, or to zero in the case of the alphabetic DOT number.
- For the single digit and seven digit USDOT numbers, the snapshot retrieval succeeded and the snapshots were sent to the requester's mailbox and read from it. For the non-existing USDOT# and the one containing alphabetic characters (which was converted to zero, a non-existing USDOT#), the database queries failed, and a failure message was sent to the requester's mailbox.

Reviewed By (SAIC): __________________ Date: _______________

Approved By (JHU/APL): __________________ Date: _______________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: IMH-CS-005
Version Control ID: APL V01-B02
Affected CSCI: IMH
Integration Function:

1. Process a carrier snapshot request that is pure nonsense (that does not begin with ‘R’, ‘C’,or ‘S’).
2. Process a carrier snapshot request containing ‘R’ followed by nonsense.
3. Process a carrier snapshot request containing ‘C’ followed by nonsense.
4. Process a carrier snapshot request containing ‘S’ followed by nonsense.

Requirement to be validated:

- IMH must process erroneous incoming requests by rejecting them without crashing, and by informing the requester of an error if appropriate.

Test Tools, Drivers, or Special Conditions:

The EudoraLite mail package was used to send the requests to the input mailbox. The MBX_Read routine was used to extract a message from the mailbox.

Input Data:

Mail messages, with carrier snapshot requests containing pure nonsense, ‘R’ followed by nonsense, ‘S’ followed by nonsense, and ‘C’ followed by nonsense.

Expected Outcome:

IMH will not forward the pure nonsense request to SDM, since it is not a request at all.
IMH will forward ‘C’ followed by nonsense to SDM as a carrier snapshot update request although it does not contain a valid record for updating the database.
IMH will not forward ‘R’ or ‘S’ followed by nonsense requests to SDM.

Test Procedure Steps:

Set up the messages in Eudora. Send them to the IMH input mailbox.

View output screen for possible error messages.

Check request structures forwarded to SDM_retrieve_safety_data. for the ‘R’ and ‘S’ plus nonsense requests.

Check structures forwarded to SDM_update_safety_data. for the ‘C’ plus nonsense request.

Analvsis Procedures:

-

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: IMH-CS-005
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- For the pure nonsense request, IMH printed a message to the console that it was not a valid message type and did not process it further.
- For the ‘C’ plus nonsense message, IMH accepted it as a snapshot update request and forwarded it to SDM.
- For the 'R' plus nonsense message, IMH accepted it as a snapshot retrieval request, but did not extract a valid USDOT# from the message. It forwarded a request for USDOT# zero to SDM.
- The ‘S’ plus nonsense message caused IMH to crash while trying to parse the EDT segments to obtain a USDOT#. (This problem was corrected later by improving error handling in the ED1 parse routines. After correction, a request for USDOT# was forwarded to SDM, but IMH no longer crashed.)

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: SDM-CS-005
Version Control ID: APL V01-B02
Affected CSCI: SDM
Integration Function:

1. Process a carrier snapshot request that is pure nonsense (that does not begin with 'R', 'C', or 'S').
2. Process a carrier snapshot request containing 'R'followed by nonsense.
3. Process a carrier snapshot request containing 'C'followed by nonsense.
4. Process a carrier snapshot request containing 'S'followed by nonsense.
5. Forward results, if any, to OMH for further processing.

Requirement to be validated:

- SDM must process erroneous incoming requests without crashing.

Test Tools, Drivers, or Special Conditions:
None.

Input Data:
Input for 'R'or 'S'plus nonsense was the carrier snapshot request structure. Input for 'C' plus nonsense was the carrier snapshot update structure.

Expected Outcome:

The pure nonsense message will never reach SDM.
'C'plus nonsense will result in a failure by SDB when trying to update the database with a totally invalid record.
'R'and 'S'plus nonsense with result in SDM trying to retrieve USDOT# = 0, which does not exist. An error message will be forwarded to OMH.

Test Procedure Steps:

Set up the messages in Eudora. Send them to the IMH input mailbox.

View output screen for possible error messages.

Check request structures forwarded to SDM_retrieve_safety_data. for the 'R'and 'S' plus nonsense requests.

Check structures forwarded to SDM_update_safety_data. for the 'C'plus nonsense request.

Analysis Procedures:

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: SDM-CS-005
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- For the pure nonsense request, SDM was not called by IMH.
- For the 'C'plus nonsense message, SDB attempted to update the database and failed.
- For the 'R'and 'S'plus nonsense message, SDM received structures containing zero for the USDOT#. SDB then failed to retrieve a snapshot. OMH was called to output an error message.

Reviewed By (SAIC): ________________ Date: ______________

Approved By (JHU/APL): ________________ Date: ______________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: OMH-CS-005
Version Control ID: APL V01-B02
Affected CSCI: OMH
Integration Function:

1. Process a carrier snapshot request that is pure nonsense (that does not begin with 'R', 'C',or 'S').
2. Process a carrier snapshot request containing 'R' followed by nonsense.
3. Process a carrier snapshot request containing 'C' followed by nonsense.
4. Process a carrier snapshot request containing 'S' followed by nonsense.
5. Forward results, if any, to requester's mailbox.

Requirement to be validated:

- OMH must process erroneous incoming requests without crashing.

Test Tools, Drivers, or Special Conditions:

EudoraLite mail package and MBX_Send routine were used to return results to the requester.

Input Data:

Input for 'R' or 'S' plus nonsense was the carrier snapshot send request structure. OMH received no input for pure nonsense or 'C' plus nonsense.

Expected Outcome:

The pure nonsense message will never reach OMH.
'C' plus nonsense will result in a failure by SDB when trying to update the database with a totally invalid record and will not require any processing by OMH.
'R' and 'S' plus nonsense with result in OMH forwarding a snapshot retrieval failure message to the requester's mailbox.

Test Procedure Steps:

Check request structures forwarded to OMH by SDM_retrieve_safety_data. for the 'R' and 'S' plus nonsense requests.

Check output screen and requester's mailbox for error messages.

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: OMH-W-005
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- For the pure nonsense request, OMH was not reached.
- For the 'C' plus nonsense message, SDB attempted to update the database and failed. No action was required of OMH
- For the 'R' and 'S' plus nonsense message, SDM received structures containing zero for the USDOT#. SDB then failed to retrieve a snapshot. OMH output snapshot retrieval failure messages to the requester's mailbox.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: IMH-CS-006
Version Control ID: APL V01-B02
Affected CSCI: IMH
Integration Function:

1. Send, accept, and process a single mailbox request for a carrier snapshot using a USDOT# as input in Aspen format (D=USDOT#).
2. Send results to SDM.

Requirement to be validated:

- IMH must read a snapshot request from the input mailbox
- IMH must recognize it as a carrier snapshot request in Aspen format
- IMH must parse the Aspen (D=#) format request to obtain the DOT number
- IMH must fill in the request structure with the DOT number
- IMH must call the appropriate SDM routine to continue processing

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the requests to the input mailbox. The MBX_Read routine was used to extract a message from the mailbox.

Input Data:
Mail message containing the following two lines:
RCS
D=4

Expected Outcome:

IMH will call SDM_retrieve_safety_data. with the packet header request type correctly filled with SAF_TRANS_DATA_REQUEST. The request structure will contain the correct DOT number in the query criteria.
The carrier snapshot for DOT number 4 should be received in text form in the requester's mailbox.

Test Procedure Steps:

Set up the message in Eudora. Send it to the IMH input mailbox.

Check packet header, request header, and request passed to SDM_retrieve_safety_data. to verify that data types and DOT number are set correctly.

Analysis Procedures:

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: IMH-CS-006
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Data structures for packet header, request header, and request were properly filled in and passed to SDM_retrieve_safety_data.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: SDM-CS-006
Version Control ID: APL V01-B02
Affected CSCI: SDM
Integration Function:

1. Detect, accept and process a single carrier snapshot request using USDOT numbers as input in Aspen format (D=USDOT#);
2. Send results to OMH.

Reauirement to be validated:

- SDM receives the packet header, request header, and request properly filled in with data types and the desired DOT number.
- SDM calls OMH with the packet header, request header, and send request properly filled in.

Test Tools, Drivers, or Special Conditions:
None

Input Data:
Packet header, request header, and request are received as input,

Expected Outcome:
OMH is called with data structures properly filled in, and is equipped with the information needed to query the database and send results back to the requester.

Test Procedure Steps:

Verify that SDM_retrieve_safety_data receives the request with the correct return address, data type and DOT number.

Verify that OMH_send_safety_data is called with data structures filled in with the correct return address, data type and DOT number.

Analysis Procedures:

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: SDM-CS-006
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- SDM_retrieve_safety_data received all information necessary to complete the retrieval.
- OMH_send_safety_data was given all information necessary to complete the retrieval.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: OMH-CS-006
Version Control ID: APL V01-B02
Affected CSCI: OMH
Integration Function:

1. Detect, accept and process a single carrier snapshot request using USDOT numbers as input in Aspen format (D=USDOT#).
2. Send results to a single requester's mailbox.

Reauirement to be validated:

- OMH must receive the proper carrier snapshot request from SDM_retrieve_safety_data.
- OMH must query the database and retrieve a carrier snapshot, provided that the DOT number is a valid one.
- OMH must send the snapshot to the requester's mailbox in text form.

Test Tools, Drivers, or Special Conditions:
The MBX_Send routine was used to send the snapshot back to the requester's mailbox. SDB_get_view was called to retrieve the snapshot from the database.

Input Data:
Packet header, request header, and send request are received by OMH as inputs.

Expected Outcome:
OMH will successfully query the database, retrieve the requested snapshot, and send it to the requester's mailbox for the request.

Test Procedure Steps:

Verify that OMH_send_safety_data receives the packet header, request header, and send request properly filled in.

Verify that OMH_output_query_database successfully calls SDB to retrieve the snapshot.

Verify that OMH_output_message_generator successfully calls MBX_Send to forward the snapshot to the requester.

Read the snapshot for the desired DOT number from the requester's mailbox.

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: OMH-CS-006
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The desired snapshot was received in the requester's mailbox, and was read using the EudoraLite mail package.

Reviewed By (SAIC): ____________ Date: ____________

Approved By (JHU/APL): ____________ Date: ____________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: IMH-CS-009
Version Control ID: APL V01-B02
Affected CSCI: IMH
Integration Function:

1. Send, accept, and process mailbox requests for a carrier snapshot using USDOT numbers as inputs at various rates to determine system responsiveness.
2. Determine saturation point- first point at which mailbox is filling faster than it is emptying.
3. Before doing this, determine the minimum Sleep statement that will prevent POP3 error messages.
4. Send results to SDM.

Requirement to be validated:

- IMH must read a snapshot request from the input mailbox
- IMH must fill in the request structure with the DOT number
- IMH must call the appropriate SDM routine to continue processing
- IMH must perform the above steps repeatedly under any reasonable level of system load.
- Spurious POP3 error messages should not be printed to the output screen.

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the requests to the input mailbox. The MBX_Read routine was used to extract a message from the mailbox.

For this test, multiple requesters must be used to stress the system. One requester cannot send messages fast enough with EudoraLite, even if they are set up in advance.

During the test, no other processing should be done on the system.

To further stress the system, one hundred requests were queued in the input mailbox before the IMH process was started. Then three requester's each sent one hundred additional requests at the same time. (Some of the three hundred requests were submitted twice).

Input Data:
Three hundred mail messages containing carrier snapshot requests in ED1 format.

Expected Outcome:

Large numbers of requests will not crash or confuse IMH. IMH will be able to handle three to four hundred snapshot requests within a period of five to seven minutes.

Test Procedure Steps:

Each requester wiII read the responses sent by OMH to his/her mailbox and verify that a snapshot was received corresponding to each request submitted. Set up three hundred different carrier snapshot request messages in Eudora.
Send one hundred to the IMH input mailbox.
Start IMH processing.
Each requester will immediately send the assigned test messages to the IMH input mailbox.

Wait a few minutes for IMH to finish all retrievals.

Analvsis Procedures:

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: IMH-CS-009
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- A Sleep statement of 225 milliseconds prevented POP3 error messages. 200 milliseconds did not prevent all error messages, but did prevent most, so 225 ms seems to be a good threshold.
- Carrier snapshots were rapidly received for the one hundred requests queued up before IMH processing began. Two of the requesters rapidly received carrier snapshots for all requests submitted. The third requester was not using a proper Eudora account to send requests and receive responses. Some of his requests succeeded, but most failed. However, the problem seemed to be within the Eudora setup, not within IMH. IMH did not crash, and perfectly processed all other requests.
- A saturation point was not reached with three people testing. Further testing must be done with a larger number of requesters.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: SDM-CS-009
Version Control ID: APL V01-B02
Affected CSCI: SDM
Integration Function:

1. Send, accept, and process mailbox requests for a carrier snapshot using USDOT numbers as inputs at various rates to determine system responsiveness.
2. Determine saturation point- first point at which mailbox is filling faster than it is emptying.
3. Before doing this, determine the minimum Sleep statement that will prevent POP3 error messages.
4. Send results to OMH.

Requirement to be validated:

- SDM receives the packet header, request header, and request properly filled in with data types and the desired DOT number.
- SDM calls OMH with the packet header, request header, and send request properly filled in.

Test Tools, Drivers, or Special Conditions:
None

Input Data:
Packet header, request header, and request are received as input for each request.

Expected Outcome:
SDM receives requests properly filled in with the USDOT numbers.
OMH is called with data structures properly filled in, and is equipped with the information needed to query the database and send results back to the requester.
Large number of requests will be processed sequentially and result in no confusion or overload.

Test Procedure Stens:

Set up three hundred different carrier snapshot request messages in Eudora.
Send one hundred to the IMH input mailbox.
Start IMH processing.

Each requester will immediately send the assigned test messages to the IMH input mailbox.

Wait a few minutes for IMH to finish all retrievals.

Verify that SDM_retrieve_safety_data. receives the request with the correct return address, data type and DOT number.

Verify that OMH_send_safety_data is called with data structures filled in with the correct return address, data type and DOT number.

Analysis Procedures:

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: SDM-CS-009
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/l//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Carrier snapshots were rapidly received for the one hundred requests queued up before processing began. Two of the requesters rapidly received carrier snapshots for all requests submitted. The third requester was not using a proper Eudora account to send requests and receive responses. Some of his requests succeeded, but most failed. However, the problem seemed to be within the Eudora setup, not within SDM. SDM did not crash, and perfectly processed all other requests.
- SDM_retrieve_safety_data received all information necessary to complete the retrievals.
- OMH_send_safety_data was given all information necessary to complete the retrievals.
-

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Snapshot
ITP ID: OMH-CS-009
Version Control ID: APL V01-B02
Affected CSCI: OMH
Integration Function:

1. Send, accept, and process mailbox requests for a carrier snapshot using USDOT numbers as inputs at various rates to determine system responsiveness.
2. Determine saturation point- first point at which mailbox is filling faster than it is emptying.
3. Before doing this, determine the minimum Sleep statement that will prevent POP3 error messages.
4. Send results to requesters' mailboxes.

Requirement to be validated:

- OMH must receive the proper carrier snapshot request from SDM_retrieve_safety_data.
- OMH must query the database and retrieve a carrier snapshot, provided that the DOT number is a valid one.
- OMH must send the snapshot to the requester's mailbox in text form.
- Rapidly processing a large number of requests in sequence will not cause a crash or processing overload for OMH.

Test Tools, Drivers, or Special Conditions:
The MBX_Send routine was used to send the snapshot back to the requester's mailbox. SDB_get_view was called to retrieve the snapshot from the database.

Input Data:
Packet header, request header, and send request are received by OMH as inputs for each request.

Expected Outcome:
OMH will successfully query the database, retrieve the requested snapshot, and send it to the requester's mailbox for the request even though hundreds of snapshot requests are received in a few minutes.

Test Procedure Steps:

Set up three hundred different carrier snapshot request messages in Eudora.

Send one hundred to the IMH input mailbox.

Start IMH processing.

Each requester will immediately send the assigned test messages to the IMH input mailbox.

Wait a few minutes for IMH to finish all retrievals.

Verify that OMH_send_safety_data receives the packet header, request header, and send request properly filled in.

Verify that OMH_output_query_database successfully calls SDB to retrieve the snapshot.

Verify that OMH_output_message_generator successfully calls MBX_Send to forward the snapshot to the requester.

Read the snapshot for the desired DOT number from the requester's mailbox.

Analvsis Procedures:

-

Integration Test Report

Integration ID: Carrier Snapshot
ITP ID: OMH-CS-009
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/l//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Carrier snapshots were rapidly received for the one hundred requests queued up before processing began. Two of the requesters rapidly received carrier snapshots for all requests submitted. The third requester was not using a proper Eudora account to send requests and receive responses. Some of his requests succeeded, but most failed. However, the problem seemed to be within the Eudora setup, not within OMH. OMH did not crash, and perfectly processed all other requests.

Reviewed By (SAIC): __________________ Date: _______________

Approved By (JHU/APL): __________________ Date: _______________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: IMH-CU-001
Version Control ID: APL V01-B02
Affected CSCI: IMH
Integration Function:

1. Send, accept and process a single mailbox request for carrier snapshot add operation.
2. Use an empty subscriber list (no subscribers).
3. Send results to SDM.

Reauirement to be validated:

- IMH must read a snapshot add request from the input mailbox
- IMH must recognize it as a carrier snapshot add request
- IMH must call the appropriate SDM routine to continue processing and provide it with the correct data types and the database update record itself.

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the requests to the input mailbox. The MBX-Read routine was used to extract messages from the mailbox.

Input Data:

One database update request, with valid data, and a USDOT# not yet existing in the database.

Expected Outcome:

IMH will call SDM_update_safety_data. with the correct packet header, data header, and update data.

Test Procedure Stew:

Check packet header, data header, and update data passed to SDM_update_safety_data to verify that data structures are correctly filled in for the update request.

Analysis Procedures:

Integration Test Report

Integration ID: Carrier Update
ITP ID: IMH-CU-001
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Data structures for packet header, data header, and update data were properly filled in and passed to SDM_update_safety_data.

Reviewed By (SAIC): ________________ Date: ______________

Approved By (JHU/APL): ________________ Date: ______________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: SDM-CU-001
Version Control ID: APL V01-B02
Affected CSCI: SDM
Integration Function:

1. Send, accept and process a single mailbox request for carrier snapshot add operation.
2. Use an empty subscriber list (no subscribers).
3. Send results to OMH if appropriate.

Reauirement to be validated:

- SDM_update_data must receive a packet header, data header, and update data filled in with accurate information so that it can update the database.

Test Tools, Drivers, or Special Conditions:

None.

Input Data:

One database update request, with valid data, and a USDOT# not yet existing in the database.

Subscription list file (existing, but empty).

Expected Outcome:

IMH will call SDM_update_safety_data with the correct packet header, data header, and update data.

Test Procedure Steps:

Check packet header, data header, and update data passed to SDM_update_safety_data to verify that data structures are correctly filled in for the update request.

Verify that no attempt is made by SL_fulfill_subscription to send updates to users, since the subscription list is empty in this test.

Analysis Procedures:

Integration Test Report

Integration ID: Carrier Update
ITP ID: SDM-CU-001
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Data structures for packet header, data header, and update data were properly filled in and passed to SDM_update_safety_data.
- SL_fulfill_subscription correctly found that the subscription list was empty, and did not try to send updates to any mailboxes.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: OMH-CU-001
Version Control ID: APL V01-B02
Affected CSCI: OMH
Integration Function:

1. Send, accept and process a single mailbox request for carrier snapshot add operation.
2. Use an empty subscriber list (no subscribers).
3. Do not attempt to send updates to any subscribers.

Reauirement to be validated:

- Since there are no subscribers who need to receive updates, OMH should not be reached in this test.

Test Tools, Drivers, or Special Conditions:
None.

Input Data:

No input should be received by OMH.

Expected Outcome:

OMH will never be called and will do nothing in this test case.

Test Procedure Steps:

Verify that within SL_fulfill_subscription, since the subscription list is empty, SDM_retrieve_safety_data. is never called.

Check output screen for possible error messages.

Check mailboxes for erroneous attempts to send updates to users.

Analysis Procedures:

Integration Test Report

Integration ID: Carrier Update
ITP ID: OMH-CU-001
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- OMH was never called, and did nothing.

Reviewed By (SAIC): ________________ Date: ____________

Approved By (JHU/APL): ________________ Date: ____________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: IMH-CU-002
Version Control ID: APL V01-B02
Affected CSCI: IMH
Integration Function:

1. Send, accept and process a single mailbox request for carrier snapshot add operation.
2. Use a subscriber list containing one and only one valid subscriber.
3. Send results to SDM.

Requirement to be validated:

- IMH must read a snapshot add request from the input mailbox
- IMH must recognize it as a carrier snapshot add request
- IMH must call the appropriate SDM routine to continue processing and provide it with the correct data types and the database update record itself.

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the requests to the input mailbox. The MBX_Read routine was used to extract messages from the mailbox.

Input Data:

One database update request, with valid data, and a USDOT# not yet existing in the database.

Expected Outcome:

IMH will call SDM_update_safety_data. with the correct packet header, data header, and update data.

Test Procedure Steps:

Check packet header, data header, and update data passed to SDM_update_safety_data. to verify that data structures are correctly filled in for the update request.

Analysis Procedures:

Integration Test Report

Integration ID: Carrier Update
ITP ID: IMH-CU-002
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Data structures for packet header, data header, and update data were properly filled in and passed to SDM_update_safety_data.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: SDM-CU-002
Version Control ID: APL V01-B02
Affected CSCI: SDM
Integration Function:

1. Send, accept and process a single mailbox request for carrier snapshot add operation.
2. Use a subscriber list containing one and only one valid subscriber.
3. Send results to OMH to forward to the subscriber.

Requirement to be validated:

- SDM_update_data must receive a packet header, data header, and update data filled in with accurate information so that it can update the database.
- SDM must call SL_fulfill_subscription to forward a snapshot update to the one subscriber.

Test Tools, Drivers, or Special Conditions:
Use SQLPlus **to view** the newly added record and verify that it was written to the database properly.

Input Data:

One database update request, with valid data, and a USDOT# not yet existing in the database.

Subscription list file containing the mail address for one subscriber.

Expected Outcome:

SDM_update_safety_data will receive the correct packet header, data header, and update data.

SL_fulfill_subscription will call SDM_retrieve_safety_data to obtain the updated snapshot and request OMH to forward it to the subscriber.

Test Procedure Steps:

Check packet header, data header, and update data passed to SDM_update_safety_data. to verify that data structures are correctly filled in for the update request.

Verify that SL_fulfill_subscription and SDM_retrieve_safety_data. perform all processing necessary to enable OMH to forward a copy of the updated snapshot to the one subscriber.

Analysis Procedures:

Integration Test Report

Integration ID: Carrier Update
ITP ID: SDM-W-002
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Data structures for packet header, data header, and update data were properly filled in and passed to SDM_update_safety_data SDM_update_safety_data. successfully added the record to the database.
- SL_fulfill_subscription correctly read the subscription list, and called SDM_retrieve_safety_data to retrieve the updated snapshot and forward information to OMH.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: OMH-W-002
Version Control ID: APL V01-B02
Affected CSCI: OMH
Integration Function:

1. Send, accept and process a single mailbox request for carrier snapshot add operation.
2. Use a subscriber list containing one and only one subscriber.
3. Send a copy of the updated snapshot to the one subscriber.

Requirement to be validated:

- OMH will forward a mail message containing a copy of the newly added snapshot to the one subscriber.

Test Tools, Drivers, or Special Conditions:
MBX_Send routine is used to forward the message to the subscriber's mailbox. EudoraLite mail package is used to read the message and verify that an updated carrier snapshot was received by the subscriber.

Input Data:

A valid packet header, request header, and send request will be the input to OMH.

Expected Outcome:

OMH will query the database, obtain a copy of the newly added record, and send it to the subscriber.

Test Procedure Steps:

Verify that OMH_query_database successfully retrieves the new record.

Verify that OMH_output_message_generator forwards the new snapshot to the subscriber.

Read message in subscriber's mailbox. Verify that the correct snapshot was sent.

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Update
ITP ID: OMH-CU-002
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The new carrier snapshot was sent to the subscriber's mailbox.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: IMH-CU-003
Version Control ID: APL V01-B02
Affected CSCI: IMH
Integration Function:

1. Send, accept and process a single mailbox request for carrier snapshot add operation.
2. Use a subscriber list containing at least three subscribers, one of which is invalid.
3. Send results to SDM.

Requirement to be validated:

- IMH must read a snapshot add request from the input mailbox
- IMH must recognize it as a carrier snapshot add request
- IMH must call the appropriate SDM routine to continue processing and provide it with the correct data types and the database update record itself.

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the requests to the input mailbox. The MBX_Read routine was used to extract messages from the mailbox.

Input Data:

One database update request, with valid data, and a USDOT# not yet existing in the database.

Expected Outcome:

IMH will call SDM_update_safety_data. with the correct packet header, data header, and update data.

Test Procedure Steps:

Check packet header, data header, and update data passed to SDM_update_safety_data. to verify that data structures are correctly filled in for the update request.

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Update
ITP ID: IMH-W-003
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

. Data structures for packet header, data header, and update data were properly filled in and passed to SDM_update_safety_data.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: SDM-W-003
Version Control ID: APL V01-B02
Affected CSCI: SDM
Integration Function:

1. Send, accept and process a single mailbox request for carrier snapshot add operation.
2. Use a subscriber list containing at least three subscribers, one of which is invalid.
3. Send results to OMH to forward to the subscribers.

Reuuirement to be validated:

- SDM_update_data must receive a packet header, data header, and update data filled in with accurate information so that it can update the database.
- SDM must call SL_fulfill_subscription to forward a snapshot update to the subscribers.

Test Tools, Drivers, or Special Conditions:
Use SQLPlus to view the newly added record and verify that it was written to the database properly.

Input Data:

One database update request, with valid data, and a USDOT# not yet existing in the database.

Subscription list file containing the mail address for three or more subscribers, one of which is invalid.

Expected Outcome:

SDM_update_safety_data. will receive the correct packet header, data header, and update data.

SL_fulfill_subscription will call SDM_retrieve_safety_data. to obtain the updated snapshot and request OMH to forward it to the subscribers.

Test Procedure Steas:

Check packet header, data header, and update data passed to SDM_update_safety_data. to verify that data structures are correctly filled in for the update request.

Verify that SL_fulfill_subscription and SDM_retrieve_safety_data perform all processing necessary to enable OMH to forward a copy of the updated snapshot to the subscribers.

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Update
ITP ID: SDM-CU-003
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Data structures for packet header, data header, and update data were properly filled in and passed to SDM_update_safety_data. SDM_update_safety_data successfully added the record to the database.
- SL_fulfill_subscription correctly read the subscription list, and called SDM_retrieve_safety_data. to retrieve the updated snapshot and forward information to OMH.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: OMH-W-003
Version Control ID: APL V01-B02
Affected CSCI: OMH
Integration Function:

1. Send, accept and process a single mailbox request for carrier snapshot add operation.
2. Use a subscriber list containing at least three subscribers, one of which is invalid.
3. Send a copy of the updated snapshot to the subscribers.

Requirement to be validated:

- OMH will forward a mail message containing a copy of the newly added snapshot to the valid subscribers.
- Having an invalid subscriber on the list will not cause a crash or processing problems.

Test Tools, Drivers, or Special Conditions:

MBX_Send routine is used to forward the message to the subscriber's mailbox. EudoraLite mail package is used to read the message and verify that an updated carrier snapshot was received by the subscriber.

Input Data:

A valid packet header, request header, and send request will be the input to OMH.

Expected Outcome:

OMH will query the database, obtain a copy of the newly added record, and send it to the subscribers.

Test Procedure Steps:

Verify that OMH_query_database successfully retrieves the new record.

Verify that OMH_output_message_generator forwards the new snapshot to the subscribers.

Read message in each subscriber's mailbox. Verify that the correct snapshot was sent.

Check output screen for error messages concerning invalid subscriber.

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Update
ITP ID: OMH-CU-003
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The new carrier snapshot was sent to each valid subscriber's mailbox.
- An error message concerning "Unknown user" was printed on the output screen. The invalid subscriber caused no processing problems.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: IMH-W-006
Version Control ID: APL V01-B02
Affected CSCI: IMH
Integration Function:

1. Send, accept and process five separate mailbox requests for carrier snapshot add operations.
2. Use a subscriber list containing at least three subscribers, one of which is invalid.
3. Send results to SDM.

Requirement to be validated:

- IMH must read a snapshot add request from the input mailbox
- IMH must recognize it as a carrier snapshot add request
- IMH must call the appropriate SDM routine to continue processing and provide it with the correct data types and the database update record itself.
- IMH must be able to perform the above steps five successive times.

Test Tools, Drivers, or Special Conditions:
The EudoraLite mail package was used to send the requests to the input mailbox. The MBX_Read routine was used to extract messages from the mailbox.

Input Data:

Five database update requests, with valid data, and USDOT#s not yet existing in the database.

Expected Outcome:

IMH will call SDM_update_safety_data. with the correct packet header, data header, and update data for each update request.

Test Procedure Steps:

Check packet header, data header, and update data passed to SDM_update_safety_data to verify that data structures are correctly filled in for each update request.

Analvsis Procedures:

-

Integration Test Report

Integration ID: Carrier Update
ITP ID: IMH-W-006
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: **7/1//96**

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Data structures for packet header, data header, and update data were properly filled in and passed to SDM_update_safety_data for each of the five requests.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: SDM-CU-006
Version Control ID: APL V01-B02
Affected CSCI: SDM
Integration Function:

1. Send, accept and process five separate mailbox requests for carrier snapshot add operations.
2. Use a subscriber list containing at least three subscribers, one of which is invalid.
3. Send results to OMH to forward to the subscribers.

Reauirement to be validated:

- SDM_update_data must receive a packet header, data header, and update data tilled in with accurate information so that it can update the database.
- SDM must call SL_fulfill_subscription to forward a snapshot update to the subscribers.
- SDM must perform the above steps five times successively.

Test Tools, Drivers, or Special Conditions:
Use SQLPlus to view the newly added records and verify that they were written to the database properly.

Input Data:

Five database update requests, with valid data, and USDOT#'s not yet existing in the database.

Subscription list file containing the mail address for three or more subscribers, one of which is invalid.

Expected Outcome:

SDM_update_safety_data. will receive the correct packet header, data header, and update data for each request.

SL_fulfill_subscription will call SDM_retrieve_safety_data to obtain the updated snapshot and request OMH to forward it to the subscribers for each request.

Test Procedure Steps:

Check packet header, data header, and update data passed to SDM_update_safety_data to verify that data structures are correctly filled in for the update requests.

Verify that SL_fulfill_subscription and SDM_retrieve_safety_data perform all processing necessary to enable OMH to forward a copy of each of the five updated snapshots to the subscribers.

Analysis Procedures:

Integration Test Report

Integration ID: Carrier Update
ITP ID: SDM-CU-006
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Data structures for packet header, data header, and update data were properly filled in and passed to SDM_update_safety_data.SDM_update_safety_data successfully added the five records to the database.
- SL_fulfill_subscription correctly read the subscription list, and called SDM_retrieve_safety_data to retrieve the updated snapshots and forward information to OMH.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: OMH-CU-006
Version Control ID: APL V01-B02
Affected CSCI: OMH
Integration Function:

1. Send, accept and process five separate mailbox requests for carrier snapshot add operations.
2. Use a subscriber list containing at least three subscribers, one of which is invalid.
3. Send a copy of the updated snapshots to the subscribers.

Requirement to be validated:

- OMH will forward mail messages containing copies of the newly added snapshots to the valid subscribers.
- Having an invalid subscriber on the list will not cause a crash or processing problems.

Test Tools, Drivers, or Special Conditions:
MBX_Send routine is used to forward the messages to each subscriber's mailbox. EudoraLite mail package is used to read the messages and verify that updated carrier snapshots was received by the subscribers.

Input Data:

A valid packet header, request header, and send request will be the input to OMH for each of five requests.

Expected Outcome:

OMH will query the database, obtain copies of the newly added records, and send it to the subscribers.

Test Procedure Steps:

Verify that OMH_query_database successfully retrieves the new records.

Verify that OMH_output_message_generator forwards the new snapshots to the subscribers.

Read messages in each subscriber's mailbox. Verify that the correct snapshots were sent.

Check output screen for error messages concerning invalid subscriber.

Analvsis Procedures:

-

Integration Test Report

Integration ID: Carrier Update
ITP ID: OMH-CU-006
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The five new carrier snapshots were sent to each valid subscriber's mailbox.
- An error message concerning "Unknown user" was printed on the output screen. The invalid subscriber caused no processing problems.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: IMH-CU-007
Version Control ID: APL V01-B02
Affected CSCI: IMH
Integration Function:

1. Send, accept and process five separate mailbox requests for carrier snapshot add operations when five add requests are already in the queue when IMH processing starts.
2. Use a subscriber list containing at least three subscribers, one of which is invalid.
3. Send results to SDM.

Reauirement to be validated:

- IMH must read a snapshot add request from the input mailbox
- IMH must recognize it as a carrier snapshot add request
- IMH must call the appropriate SDM routine to continue processing and provide it with the correct data types and the database update record itself.
- IMH must be able to process five queued requests immediately upon being initialized, and then perform the above steps five successive times for additional update records.

Test Tools, Drivers, or Special Conditions:

The EudoraLite mail package was used to send the requests to the input mailbox. The MBX_Read routine was used to extract messages from the mailbox.

Input Data:

Ten database update requests, with valid data, and USDOT#'s not yet existing in the database.

Expected Outcome:

IMH will call SDM_update_safety_data with the correct packet header, data header, and update data for each update request.

Test Procedure Steps:

Check packet header, data header, and update data passed to SDM_update_safety_data to verify that data structures are correctly filled in for each update request.

Analysis Procedures:

-

Integration Test Report

Integration ID: Carrier Update
ITP ID: IMH-CU-007
Version Control ID: APL V01-B02
Test Conductor: Grace McGonnigal

Test Seq Number: 01
Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Data structures for packet header, data header, and update data were properly filled in and passed to SDM_update_safety_data for each of the ten requests.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: SDM-CU-007
Version Control ID: APL V01-B02
Affected CSCI: SDM
Integration Function:

1. Send, accept and process five separate mailbox requests for carrier snapshot add operations when five requests are already in the queue when IMH process is started..
2. Use a subscriber list containing at least three subscribers, one of which is invalid.
3. Send results to OMH to forward to the subscribers.

Requirement to be validated:

- SDM_update_data must receive a packet header, data header, and update data filled in with accurate information so that it can update the database.
- SDM must call SL_fulfill_subscription to forward a snapshot update to the subscribers.
- SDM must perform the above steps ten times successively.

Test Tools, Drivers, or Special Conditions:
Use SQLPlus to view the newly added records and verify that they were written to the database properly.

Input Data:

Ten database update requests, with valid data, and USDOT#'s not yet existing in the database.

Subscription list file containing the mail address for three or more subscribers, one of which is invalid.

Expected Outcome:

SDM_update_safety_data. will receive the correct packet header, data header, and update data for each request.

SL_fulfill_subscription will call SDM_retrieve_safety_data to obtain the updated snapshot and request OMH to forward it to the subscribers for each request.

Test Procedure Steps:

Check packet header, data header, and update data passed to SDM_update_safety_data to verify that data structures are correctly filled in for the update requests.

Verify that SL_fulfill_subscription and SDM_retrieve_safety_data perform all processing necessary to enable OMH to forward a copy of each of the ten updated snapshots to the subscribers.

Analysis Procedures:

Integration Test Plan

Integration ID: Carrier Update
ITP ID: SDM-CU-007
Version Control ID: APL V01-B02 Test Seq Number: 01
Test Conductor: Grace McGonnigal Test Date: 7/l//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Data structures for packet header, data header, and update data were properly filled in and passed to SDM_update_safety_data. SDM_update_safety_data. successfully added the ten records to the database.
- SL_fulfill_subscription correctly read the subscription list, and called SDM_retrieve_safety_data to retrieve the updated snapshots and forward information to OMH.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Integration Test Plan

Integration ID: Carrier Update
ITP ID: OMH-CU-007
Version Control ID: APL V01-B02
Affected CSCI: OMH
Integration Function:

1. Send, accept and process five separate mailbox requests for carrier snapshot add operations when five add request messages are already in the queue when IMH processing is started.
2. Use a subscriber list containing at least three subscribers, one of which is invalid.
3. Send a copy of the updated snapshots to the subscribers.

Requirement to be validated:

- OMH will forward mail messages containing copies of the newly added snapshots to the valid subscribers.
- Having an invalid subscriber on the list will not cause a crash or processing problems.

Test Tools, Drivers, or Special Conditions:

MBX_Send routine is used to forward the messages to each subscriber's mailbox. EudoraLite mail package is used to read the messages and verify that updated carrier snapshots was received by the subscribers.

Input Data:

A valid packet header, request header, and send request will be the input to OMH for each of ten requests.

Expected Outcome:

OMH will query the database, obtain copies of the newly added records, and send it to the subscribers.

Test Procedure Steps:

Verify that OMH_query_database successfully retrieves the new records.

Verify that OMH_output_message_generator forwards the new snapshots to the subscribers.

Read messages in each subscriber's mailbox. Verify that the correct snapshots were sent.

Check output screen for error messages concerning invalid subscriber.

Analysis Procedures:

Integration Test Report

Integration ID: Carrier Update
ITP ID: OMH-W-007
Version Control ID: APL V01-B02
Test Conductor: Grace McGonnigal

Test Seq Number: 01
Test Date: 7/1//96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- The ten new carrier snapshots were sent to each valid subscriber's mailbox.
- An error message concerning "Unknown user" was printed on the output screen. The invalid subscriber caused no processing problems.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Appendix D

Administrative Manager Unit Test Plans and Test Reports

Unit Test Plan

Unit ID: ADM_ACCT_create_org_account
UTP ID: ACCT_CRE_ORGl
Version Control ID: SAIC V01-B01
Unit Function:
It creates an organizational account, and multiple user accounts if the request is valid. This function may insert an entry into ORG_ACCOUNT table, an entry into BANK table, and entries into USER table.

Reauirement to be validated:
. 3.2.1.1
- 3.2.2

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\VOl-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_Org_Create_dvr1.c

Input Data:
Key input data:
Requester-user-id = "jsmith"
Org name = "Science Applications International Corporation"
Org Shortname = "SAIC"
Bank name = "First Virginia"
Bank branch = "McLean"
Two Users:
User1 = "jsmith" with super user privileges
User2 = "jdoe"

Expected Outcome:
An entry keyed "SAIC_x" is added to Org_Account table.
An entry keyed "First Virginia" + "McLean" is added to Bank table.
Two entries keyed "jsmith" and "jdoe" are added to User-Account table.

Test Procedure Stew:
-

Analvsis Procedures:
-

Unit Test Report

Unit ID: ADM_ACCT_create_org_account
UTP ID: ACCT_CRE_ORG 1
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 12/16/96

Purpose of Test:
Test on successful creation of organizational accounts and user accounts.

Actual Results:

- An organizational account keyed "SAIC-1" is inserted into Org_Account table.
- A bank keyed "First Virginia" + "McLean" is inserted into Bank table.
- Two users keyed "SAIC-1" + "jsmith" and "SAIC-1" + "jdoe" are inserted into User-Account table.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Unit Test Plan

Unit ID: ADM_ACCT_create_org_account
UTP ID: ACCT_CRE_ORG2
Version Control ID: SAIC V01-B01
Unit Function:
It creates an organizational account, and multiple user accounts if the request is valid. This function may insert an entry into ORG_ACCOUNT table, an entry into BANK table, and entries into USER table.

Requirement to be validated:

. 3.2.1.1
- 3.2.2

Test Tools, Drivers, or Special Conditions:
Special Conditions:
An organization with name "Science Applications International Corporation" and no suborg name exists in the database.
Test Drivers:
\\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_Org_Create_dvr2.c

Input Data:
Key input data:
1. Requester-user-id = "jsmith"
Org name = "Science Applications International Corporation"
Org Shortname = "SAIC"
Two Users:
User1 = "jsmith" with super user privileges
User2 = "jdoe"

Expected Outcome:
Appropriate error messages displayed (on the screen). No account will be created. No further processing on the request.

Test Procedure Steps:

-

Analysis Procedures:

-

Unit Test Report

Unit ID: ADM_ACCT_create_org_account
UTP ID: ACCT_CRE_ORG2
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 12/16/96

Purpose of Test:
Test on unsuccessful creation of organizational accounts and user accounts.

Actual Results:

- Error message "Org exists, please provide a different suborg name to create a new account." displayed on the screen.
- No entries are added to any of the database tables.
- Program terminated.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Unit Test Plan

Unit ID: ADM_ACCT_create_org_account
UTP ID: ACCT_CRE_ORG3
Version Control ID: SAIC V01-B01
Unit Function:
It creates an organizational account, and multiple user accounts if the request is valid. This function may insert an entry into ORG_ACCOUNT table, an entry into BANK table, and entries into USER table.

Requirement to be validated:

. 3.2.1.1
- 3.2.2

Test Tools, Drivers, or Special Conditions:
Special Conditions:
An organization with name "Science Applications International Corporation" and no suborg name exists in the database.
Test Drivers:
\\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_Org_Create_dvr3.c

Input Data:
Key input data:
1. Requester-user-id = "ssmith"
Org name = "Science Applications International Corporation"
Suborg name = "SAFER Program"
Org Shortname = "SAIC"
Two Users:
User1 = "jsmith" with super user privileges
User2 = "jdoe"

Expected Outcome:
Appropriate error messages displayed. No account will be created. No further processing on the request.

Test Procedure Steps:

Analysis Procedures:

Unit Test Report

Unit ID: ADM_ACCT_create_org_account
UTP ID: ACCT_CRE_ORG3
Version Control ID: SAIC V01-B01 Test Seu Number: 1
Test Conductor: Ping Liang Test Date: 12/16/96

Purpose of Test:
Test on unsuccessful creation of organizational accounts and user accounts.

Actual Results:

- Error message "Requester id is not in the user list to be created." displayed on the screen.
- No changes to any of the database tables.
- Program terminated.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Unit Test Plan

Unit ID: ADM_ACCT_create_org_account
UTP ID: ACCT_CRE_ORG4
Version Control ID: SAIC V01-B01
Unit Function:
It creates an organizational account, and multiple user accounts if the request is valid. This function may insert an entry into ORG_ACCOUNT table, an entry into BANK table, and entries into USER table.

Requirement to be validated:

- 3.2.1.1
- 3.2.2

Test Tools, Drivers, or Special Conditions:
Special Conditions:
An organization with name "Science Applications International Corporation" and no suborg name exists in the database.
Test Drivers:
\\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_Org_Create_dvr4.c

Input Data:
Key input data:
1. Requester-user-id = 'jsmith'
Org name = "Science Applications International Corporation"
Suborg name = "SAFER Program"
Org Shortname = "SAIC"
Two Users:
User1 = "jsmith" without org-create privilege
User2 = "jdoe"

Expected Outcome:
Appropriate error messages displayed. No account will be created. No further processing on the request.

Test Procedure Steps:

Analysis Procedures:

-

Unit Test Report

Unit ID: ADM_ACCT_create_org_account
UTP ID: ACCT_CRE_ORG4
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 12/16/96

Purpose of Test:
Test on unsuccessful creation of organizational accounts and user accounts.

Actual Results:

- Error message "The requester does not have the privilege to create an org account." displayed on the screen.
- No change to any of the database tables.
- Program terminated.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Unit Test Plan

Unit ID: ADM_ACCT_create_org_account
UTP ID: ACCT_CRE_ORG5
Version Control ID: SAIC V01-B01
Unit Function:
It creates an organizational account, and multiple user accounts if the request is valid. This function may insert an entry into ORG_ACCOUNT table, an entry into BANK table, and entries into USER table.

Requirement to be validated:
. 3.2.1.1
- 3.2.2

Test Tools, Drivers, or Special Conditions:

Special Conditions:
An organization with account-id "SAIC_1", name "Science Applications International Corporation" and suborg name "UNKNOWN" exists in the database.
Test Drivers:
\\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM|ACCT\Unit_Test\Drivers\ADM_ACCT_Org_Create_dvr5.c

Input Data:
Key input data:
1. Requester-user-id = "jsmith"
Org name = "Science Applications International Corporation"
Suborg name = "SAFER Program"
Org Shortname = "SAlC"
Bank name = "First Virginia"
Bank branch = "McLean"

Three Users:
User1 = "jsmith" with super user privileges
User2 = "jsmith"
User3 = "jdoe"

Expected Outcome:
An entry keyed "SAIC-2" added to Org_Account table.
No new entry added to Bank table since the bank already exists.
The first user is added to User-Account table.
Error message on the second user creation since the user exists.
The third user is added to User-Account table.

Test Procedure Steps:

Analysis Procedures:

Unit Test Report

Unit ID: ADM_ACCT_create_org_account
UTP ID: ACCT_CRE_ORG5
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 12/16/96

Purpose of Test:
It tests on successful generation of an organizational account id and creations of an organizational account and user accounts. It also tests on continuing creation of next user when error happens on a user.

Actual Results:

- Organizational account keyed "SAIC-2" created.
- User account "jsmith" created.
- Error creating user "jsmith". Error message "User id jsmith has been taken, failed to create an account for smith." displayed.
- User account "jdoe" created.
- No new bank created. The bank part of the org references an existing entry in BANK table.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Unit Test Plan

Unit ID: ADM_ACCT_create_user_account
UTP ID: ACCT_CRE_USERl
Version Control ID: SAIC V01-B01
Unit Function:
It creates one or more user accounts if the request is valid.
This function may insert entries into USER-ACCOUNT table.

Requirement to be validated:

- 3.2.2

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM | ACCT\Unit_Test\Drivers\ADM_ACCT_User_Create_dvr1.c

Special Conditions:
Org account SAIC-1 exists with one user "jsmith" who has create-user privilege.

Input Data:
Key input data:
Requester-user-id = "jsmith"
Account-id = "SAIC-1"
Two Users:
User1 = "jjones"
User2 = "jdoe"

Expected Outcome:
Two entries keyed "jjones" and "jdoe" of SAIC-1 are added to User-Account table.

Test Procedure Steps:

Analysis Procedures:

Unit Test Report

Unit ID: ADM_ACCT_create_user_account
UTP ID: ACCT_CRE_USER1
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 12/16/96

Purpose of Test:
Test on successful creation of user accounts.

Actual Results:

- Two users keyed "SAIC-1" + "jjones" and "SAIC_1" + "jdoe" are inserted into User-Account table.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (.lHU/APL): ____________________ Date: ________________

Unit Test Plan

Unit ID: ADM_ACCT_create_user_account
UTP ID: ACCT_CRE_USER2
Version Control ID: SAIC V01-B01
Unit Function:
It creates one or more user accounts if the request is valid.
This function may insert entries into USER-ACCOUNT table.

Requirement to be validated:

- 3.2.2

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_User_Create_dvr2.c

Special Conditions:
Org account SAIC-1 exists with three users “jjones”, “jdoe”, and “jsmith”.
User “jsmith” has create-user privilege.

Input Data:
Key input data:
Requester-user-id = “jsmith”
Account-id = “SAIC-1”
User1 to create "jdoe"
User2 to create “mjohnson”
User3 with last name Jones to created (user id is not provided)

Expected Outcome:
Error message displayed while creating user “jdoe”
Successful creation of user “mjohnson”
Error message displayed while creating user with last name Jones

Test Procedure Steps:

-

Analysis Procedures:

Unit Test Report

Unit ID: ADM_ACCT_create_user_account
UTP ID: ACCT_CRE_USER2
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 12/16/96

Purpose of Test:
Test on unsuccessful creation of a user account since the user to create already exists.
Test on continuing creation of the next user even if error detected on the current one.

Actual Results:

- Error creating user jdoe. Error message "User id jdoe has been taken, failed to create an account for Doe." displayed on the screen.
- User mjohnson created.
- Error creating an account for Jones. Error message "User-id is not provided, can't create user Jones." displayed on the screen.

Reviewed By (SAIC): __________________ Date: _______________

Approved By (JHU/APL): __________________ Date: _______________

Unit Test Plan

Unit ID: ADM_ACCT_create_user_account
UTP ID: ACCT_CRE_USER3
Version Control ID: SAIC V01-B01
Unit Function:
It creates one or more user accounts if the request is valid.
This function may insert entries into USER-ACCOUNT table.

Requirement to be validated:
. 3.2.2

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_User_Create_dvr3.c

Special Conditions:
Org account SAIC-1 exists with three users "jjones", "jdoe", and "jsmith".
User "jsmith" has create-user privilege.
User "jdoe" does not have create-user privilege.

Input Data:
Key input data:
Requester-user-id = jdoe"
Account-id = "SAIC-1"
User to create "mjoe"

Expected Outcome:
Error message displayed while creating user "mdoe"

Test Procedure Steps:

Analysis Procedures:

-

Unit Test Report

Unit ID: ADM_ACCT_create_user_account
UTP ID: ACCT_CRE_USER2
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 12/16/96

Purpose of Test:
Test on unsuccessful creation of a user account due to insufficient privileges.

Actual Results:

- Error creating user mode. Error message "The requester does not have the privilege to create a user account." displayed on the screen..

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Unit Test Plan

Unit ID: ADM_ACCT_create_user_account
UTP ID: ACCT_CRE_USER3
Version Control ID: SAIC V01-B01
Unit Function:
It creates one or more user accounts if the request is valid.
This function may insert entries into USER-ACCOUNT table.

Requirement to be validated:

- 3.2.2

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM|ACCT\Unit_Test\Drivers\ADM_ACCT_User_Create_dvr3.c

Special Conditions:
Org account SAIC-1 exists with three users “jjones”, "jdoe”, and “jsmith”.
User “jsmith” has create-user privilege.
User “jdoe” does not have create-user privilege.

Input Data:
Key input data:
Requester-user-id = "jdoe”
Account-id = “SAIC-1”
User to create “mjoe”

Expected Outcome:
Error message displayed while creating user “mdoe”

Test Procedure Steps:

-

Analysis Procedures:

Unit Test Report

Unit ID: ADM_ACCT_create_user_account
UTP ID: ACCT_CRE_USER3
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 12/16/96

Purpose of Test:
Test on unsuccessful creation of a user account due to insufficient privileges.

Actual Results:

- Error creating user mode. Error message "The requester does not have the privilege to create a user account." displayed on the screen..

Reviewed By (SAIC): ___________________ Date: _______________

Approved By (JHU/APL): ___________________ **Date:** _______________

Unit Test Plan

Unit ID: ADM_ACCT_delete_org_account
UTP ID: ACCT_DEL_ORG1
Version Control ID: SAIC V01-B01
Unit Function:
This function deletes an organizational account and all its user accounts.
It may delete an entry in ORG_ACCOUNT table, an entry in BANK table and entries in USER-ACCOUNT table.

Requirement to be validated:

- 3.2.1
- 3.2.2

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_Org_Delete_dvr1.c
Special Conditions:

- Org “SAIC_1” exists with two users “jsmith” and "jdoe”
- User “jdoe” does NOT have delete-org privilege

Input Data:
Requesting-user-id = “jdoe”
Org to delete is “SAIC_1”

Expected Outcome:
Error message displayed on the screen.

Test Procedure Steps:

-

Analvsis Procedures:

-

--.

Unit Test Report

Unit ID: ADM_ACCT_delete_org_account
UTP ID: ACCT_DEL_ORG1
Version Control ID: SAIC V01-B01
Test Conductor: Ping Liang

Test Seq Number: 1
Test Date: 8/7/96

Purpose of Test:
Test on unsuccessful deletion of an org account due to insufficient privileges.

Actual Results:

- Error message "The requester does not have the privilege to delete an org account." displayed on the screen.
- Org account SAIC-1 still exists.

Reviewed By (SAIC): ____________________ Date:________________

Approved By (JHU/APL): ____________________ Date: ________________

Unit Test Plan

Unit ID: ADM_ACCT_delete_org_account
UTP ID: ACCT_DEL_ORG2
Version Control ID: SAIC V01-B01
Unit Function:
This function deletes an organizational account and all its user accounts.
It may delete an entry in ORG_ACCOUNT table, an entry in BANK table and entries in USER-ACCOUNT table.

Requirement to be validated:

- 3.2.1
- 3.2.2

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_Org_Delete_dvr2.c
Special Conditions:

- Org “SAIC-1” exists with two users “jsmith” and “jdoe”
- Org “SAIC-2” exists with one user “jjones”.

Input Data:
Requesting-user-id = “jsmith”
Org to delete is “SAIC-2”

Expected Outcome:
Error message displayed on the screen.

Test Procedure Steps:

-

Analysis Procedures:

-

Unit Test Report

Unit ID: ADM_ACCT_delete_org_account
UTP ID: ACCT_DEL_ORG2
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 8/7/96

Purpose of Test:

Test on unsuccessful deletion of an org account since the requester is not associated with the org.

Actual Results:

- Error message "Requester is not associated with the org." displayed on the screen.
- Org account SAIC-2 still exists.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ **Date:** ________________

Unit Test Plan

Unit ID: ADM_ACCT_delete_org_account
UTP ID: ACCT_DEL_ORG3
Version Control ID: SAIC V01-B01
Unit Function:
This function deletes an organizational account and all its user accounts.
It may delete an entry in ORG_ACCOUNT table, an entry in BANK table and entries in USER-ACCOUNT table.

Requirement to be validated:

- 3.2.1
- 3.2.2

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_Org_Delete_dvr3.c
Special Conditions:

- Org “SAIC_1” exists with two users “jsmith” and "jdoe"
- User “jsmith” has delete-org privilege
- Org “SAIC_2” exists with 3 users “jsmith”, “jdoe” and “jjones”

Input Data:
Requesting-user-id = “jsmith”
Org to delete is “SAIC-1”

Expected Outcome:
Org account SAIC-1 is deleted.
Users “jsmith” and “jdoe” associated with “SAIC_1” are deleted.
Users “jsmith” and “jdoe” associated with “SAIC_2" still exist.

Test Procedure Steps:

Analysis Procedures:

Unit Test Report

Unit ID: ADM_ACCT_delete_org_account
UTP ID: ACCT_DEL_ORG3
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 8/7/96

Purpose of Test:
Test on successful deletion of an org account and its user accounts.

Actual Results:

- Org account SAIC_1 does not exists anymore.
- User accounts "jsmith", "jdoe" with account-id "SAIC_1" do not exist anymore.
- User accounts "jsmith", "jdoe" with account-id "SAIC-2" still exist.

Reviewed By (SAIC): ________________ **Date:** ______________

Approved By (JHU/APL): ________________ **Date:** ______________

Unit Test Plan

Unit ID: ADM_ACCT_delete_user_account
UTP ID: ACCT_DEL_USER1
Version Control ID: SAIC V01-B01
Unit Function:
This function removes an user account associated with an org from the database. It deletes an entry in USER-ACCOUNT table.

Requirement to be validated:

- 3.2.2

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_User_Delete_dvr1.c
Special Conditions:

- Org "SAIC-2" exists with three users "jsmith", "jjones" and jdoe"
- User "jdoe" does NOT have delete-user privilege

Input Data:
Requesting-user-id = "jdoe"
User to delete is "jjones"

Expected Outcome:
Error message displayed on the screen.

Test Procedure Steps:

Analysis Procedures:

Unit Test Report

Unit ID: ADM_ACCT_delete_user_account
UTP ID: ACCT_DEL_USER1
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 8/7/96

Purpose of Test:
Test on unsuccessful deletion of an user account due to insufficient privileges.

Actual Results:

- Error message "The requester does not have the privilege to delete a user account." displayed on the screen.
- User account "jjones" associated with "SAIC-2" still exists.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Unit Test Plan

Unit ID: ADM_ACCT_delete_user_account
UTP ID: ACCT_DEL_USER2
Version Control ID: SAIC V01-B01
Unit Function:
This function removes an user account associated with an org from the database. It deletes an entry in USER-ACCOUNT table.

Requirement to be validated:

- 3.2.2

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAPER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_User_Delete_dvr2.c
Special Conditions:

- Org “SAIC-2” exists with three users “jsmith”, “jjones” and jdoe”
- Org “SAIC-1” exists with one user “pjones”

Input Data:
Requesting-user-id = “pjones”
User to delete = “jjones”

Expected Outcome:
Error message displayed on the screen.

Test Procedure Steps:

Analysis Procedures:

Unit Test Report

Unit ID: ADM_ACCT_delete_user_account
UTP ID: ACCT_DEL_USER2
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 8/7/96

Purpose of Test:
Test on unsuccessful deletion of an user account since the requester is not associated with the org the user is in.

Actual Results:

- Error message "Requester is not associated with the org." displayed on the screen.
- User account jjones" with org account SAIC-2 still exists.

Reviewed By (SAIC): ____________ Date: ____________

Approved By (JHU/APL): ____________ **Date:** ____________

Unit Test Plan

Unit ID: ADM_ACCT_delete_user_account
UTP ID: ACCT_DEL_USER3
Version Control ID: SAIC V01-B01
Unit Function:
This function removes an user account associated with an org from the database. It deletes an entry in USER-ACCOUNT table.

Requirement to be validated:
. 3.2.2

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_User_Delete_dvr3.c
Special Conditions:

- Org "SAIC_2" exists with three users "jsmith", "jjones" and jdoe"
- User "jsmith" has delete-user privilege

Input Data:
Requesting-user-id = "jsmith"
Account-id = "SAIC-2"
Users to delete are "jjones", "mdoe", "jdoe" and "jsmith"

Expected Outcome:
Successful deletion of user "jjones"
Error message diaplayed on the screen while deleting user "mdoe"
Successful deletion of user "jdoe"
Error message displayed on the screen while deleting user "jsmith'

Test Procedure Steps:

-

Analvsis Procedures:

-

Unit Test Report

Unit ID: ADM_ACCT_delete_user_account
UTP ID: ACCT_DEL_USER3
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 8/7/96

Purpose of Test:
Test on successful deletion of two user accounts and unsuccessful deletion of the last user of the org.
Test on detecting errors when deleting a user that is not associated with the org.
Test on continuing deleting next user when error happens.

Actual Results:

- User “jjones” with SAIC_2 deleted
- Error deleting user “mdoe”. Error message “SDB error in deleting user.”
- User "jdoe” with SAIC_2 deleted
- User “jsmith” with SAIC_2 was not deleted. Error message “The user is the last user. Cannot delete.” displayed on the screen.

Reviewed By (SAIC): ________________ **Date:** ______________

Approved By (JHU/APL): ________________ **Date:** ______________

Unit Test Plan

Unit ID: ADM_ACCT_delete_user_account
UTP ID: ACCT_DEL_USER3A
Version Control ID: SAIC V01-B01
Unit Function:
This function removes an user account associated with an org from the database.
It deletes an entry in USER-ACCOUNT table.

Requirement to be validated:

- 3.2.2

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_User_Delete_dvr3.c
Special Conditions:

- Org “SAIC_2” exists with three users “jsmith”, “jjones” and jdoe”
- User “jsmith” has delete-user privilege

Input Data:
Requesting-user-id = “jsmith”
Account-id = “SAIC_2”
Users to delete are “jjones”, “mdoe”, “jdoe” and “jsmith”

Expected Outcome:
Successful deletion of user “jjones”
Error message diaplayed on the screen while deleting user “mdoe”
Successful deletion of user “jdoe”
Error message displayed on the screen while deleting user jsmith”

Test Procedure Steps:

Analysis Procedures:

Unit Test Report

Unit ID: ADM_ACCT_delete_user_account
UTP ID: ACCT_DEL_USER3A
Version Control ID: SAIC V01-B01 Test Seq Number:**2**
Test Conductor: Ping Liang Test Date: 8/12/96

Purpose of Test:
Rerun previous test because of code changes. Currently, the requester can not delete himself which automatically meets the requirement of not to delete the last user.

Actual Results:

- User "jjones" with SAIC-2 deleted
- Error deleting user "mdoe". Error message "SDB error in deleting user."
- User "jdoe" with SAIC-2 deleted
- User "jsmith" with SAIC-2 was not deleted. Error message "Can not delete self." displayed on the screen.

Reviewed By (SAIC): ________________ **Date:** ______________

Approved By (JHU/APL): ________________ Date: ______________

Unit Test Plan

Unit ID: ADM_ACCT_update_org_account
UTP ID: ACCT_UPD_ORG1
Version Control ID: SAIC V01-B01
Unit Function:
This function updates fields of an organizational accounts as well as fields of the user accounts associated with the organizational account.
It may update an entry in ORG_ACCOUNT table, an entry in BANK table and entries in USER_ACCOUNT table.

Requirement to be validated:

- 3.2.1
- 3.2.2.1

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_Org_Update_dvr1.c
Special Conditions:

- Org “SAIC-1” exists with two users “jsmith” and “jdoe”
- User “jsmith” exists with update-org privilege

Input Data:
Requesting-user-id = “jsmith”
change org phone# to 7035554444
change bank contact-name to “Mary Jones”
change user “jdoe” account status to inactive

Expected Outcome:
Org SAIC-l's phone# changed to 7035554444
Org SAIC-l's bank contact person name changed to Mary Jones
User jdoe's account status changed to inactive

Test Procedure Steps:

Analysis Procedures:

•

Unit Test Report

Unit ID: ADM_ACCT_update_org_account
UTP ID: ACCT_UPD_ORG1
Version Control ID: SAIC V01-B01
Test Conductor: Ping Liang

Test Seq Number: 1
Test Date: 8/5/96

Purpose of Test:
Test on successful updating of a field in Org_Account table, a field in Bank-Account table and a field in User-Account table with one org-update request. Also tests on update of a field of character string type as well as update of a field of character type.

Actual Results:

- Org SAIC-l's phone has been changed to 7035554444
- Contact-name of the bank with which SAIC-1 is associated has been changed to Mary Jones
- User jdoe's acct_status has been changed to ‘1’
- Successful update messages displayed on the screen

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Unit Test Plan

Unit ID: ADM_ACCT_update_org_account

UTP ID: ACCT_UPD_ORG2_1

Version Control ID: SAIC V01-B01

Unit Function:

This function updates fields of an organizational accounts as well as fields of the user accounts associated with the organizational account.
It may update an entry in ORG_ACCOUNT table, an entry in BANK table and entries in USER_ACCOUNT table.

Requirement to be validated:

- 3.2.1
- 3.2.2.1

Test Tools, Drivers, or Special Conditions:

Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM|ACCT\Unit_Test\Drivers\ADM_ACCT_Org_Update_dvr2.c
Special Conditions:

- Org "SAIC_1" exists with two users "jsmith" and jdoe"
- User "jsmith" exists with update-org privilege
- SAIC_1 banks with "First Virginia", "McLean" branch and no other orgs are associated that branch.

Input Data:

Requesting-user-id = "jsmith"
Change org SAIC_1's bank-branch to "FALLS CHURCH"
Change or g SAIC_1's bank-account-id to "1234567890"

Expected Outcome:

Org SAIC_1 is associated with a new bank branch (a new entry in bank table)
Bank "First Virginia" + "McLean" is removed from Bank table
Bank-account-id of SAIC_1 in Org_Account table is changed to "1234567890"

Test Procedure Steps:

-

Analysis Procedures:

Unit Test Report

Unit ID: ADM_ACCT_update_org_account
UTP ID: ACCT_UPD_ORG2_1
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 8/6/96

Purpose of Test:
Test on changes of an org's bank to see if a new bank is created, and if the unreferenced old bank is deleted.

Actual Results:

- a new bank entry whose bank-branch = "FALLS CHURCH" gets created
- bank-account-id of SAIC-1 is updated to "1234567890"
- old bank entry whose bank-branch = "McLean" is NOT removed. BUG!

Reviewed By (SAIC): ________________ Date: ______________

Approved By (JHU/APL): ________________ Date: ______________

Unit ID: ADM_ACCT_update_org_account
UTP ID: ACCT_UPD_ORG2_2
Version Control ID: SAIC V01-B01
Unit Function:
This function updates fields of an organizational accounts as well as fields of the **user** accounts associated with the organizational account.
It may update an entry in ORG_ACCOUNT table, an entry in BANK table and entries in USER-ACCOUNT table.

Requirement to be validated:

- 3.2.1
- 3.2.2.1

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_Org_Update_dvr2.c
Special Conditions:

- Org "SAIC_1" exists with two users "jsmith" and jdoe"
- User "jsmith" exists with update-org privilege
- SAIC_1 banks with "First Virginia", "McLean" branch and no other orgs are associated that branch.

Input Data:
Requesting-user-id = "jsmith"
Change org SAIC-l's bank-branch to "FALLS CHURCH"
Change or g SAIC-l's bank-account-id to "1234567890"

Expected Outcome:
Org SAIC-1 is associated with a new bank branch (a new entry in bank table)
Bank "First Virginia" + "McLean" is removed from Bank table
Bank-account-id of SAIC-1 in Org_Account table is changed to "1234567890"

Test Procedure Steps:

Analysis Procedures:

-

Unit Test Report

Unit ID: ADM_ACCT_update_org_account
UTP ID: ACCT_UPD_ORG2_2
Version Control ID: SAIC V01-B01
Test Conductor: Ping Liang

Test Seq Number: 2
Test Date: 8/6/96

Purpose of Test:
Test on changes of an org's bank to see if a new bank is created, and if the unreferenced old bank is deleted.
Bug was found at previous test and was fixed. Rerun the test.

Actual Results:

- a new bank entry whose bank-branch = "FALLS CHURCH" gets created
- bank-account-id of SAIC_1 is updated to "1234567896'

old bank entry whose bank-branch = "McLean" is removed.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): __________________ Date: ______________

Unit Test Plan

Unit ID: ADM_ACCT_update_org_account
UTP ID: ACCT_UPD_ORG3
Version Control ID: SAIC V01-B01
Unit Function:
This function updates fields of an organizational accounts as well as fields of the user accounts associated with the organizational account.
It may update an entry in ORG_ACCOUNT table, an entry in BANK table and entries in USER_ACCOUNT table.

Requirement to be validated:
. 3.2.1

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_Org_Update_dvr3.c
Special Conditions:

- Org "SAIC_1" exists with two users "jsmith" and "jdoe"
- **SAIC-l's** phone = "7035554444"
- User "jdoe" does NOT have update-org privilege.

Input Data:
Requesting-user-id = "jdoe"
change org phone# to 7034445555

Expected Outcome:
No change is made. Error message displayed on the screen.

Test Procedure Steps:

Analysis Procedures:

-

Unit Test Report

Unit ID: ADM_ACCT_update_org_account
UTP ID: ACCT_UPD_ORG3
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 8/6/96

Purpose of Test:
Test on unsuccessful update on org field due to insufficient privileges.

Actual Results:

- Error message "The requester does not have the privilege to update an org account." displayed on the screen.
- SAIC-l's phone remains "7035554444".

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): __________________ Date: _______________

Unit Test Plan

Unit ID: ADM_ACCT_update_org_account
UTP ID: ACCT_UPD_ORG3
Version Control ID: SAIC V01-B01
Unit Function:
This function updates fields of an organizational accounts as well as fields of the user accounts associated with the organizational account.
It may update an entry in ORG_ACCOUNT table, an entry in BANE table and entries in USER_ACCOUNT table.

Requirement to be validated:

- 3.2.1

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_Org_Update_dvr3.c
Special Conditions:

- Org “SAIC-1” exists with two users "jsmith” and "jdoe”
- SAIC_1’s phone = “7035554444”
- User “jdoe” does NOT have update-org privilege.

Input Data:
Requesting-user-id = “jdoe”
change org phone# to 7034445555

Expected Outcome:
No change is made. Error message displayed on the screen.

Test Procedure Steps:

Analysis Procedures:

-

Unit Test Report

Unit ID: ADM_ACCT_update_org_account
UTP ID: ACCT_UPD_ORG3
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 8/6/96

Purpose of Test:
Test on unsuccessful update on org field due to insufficient privileges.

Actual Results:

- Error message "The requester does not have the privilege to update an org account." displayed on the screen.

Reviewed By (SAIC): ________________ Date: ______________

Approved By (JHU/APL): ________________ Date: ______________

Unit Test Plan

Unit ID: ADM_ACCT_update_user_account
UTP ID: ACCT_UPD_USER1
Version Control ID: SAIC V01-B01
Unit Function:
This function updates fields of user accounts. A user can update only a subset field of its own account unless the user has update-user privilege.
It may update entries in USER_ACCOUNT table.

Requirement to be validated:

- 3.2.2.1
- 3.2.3.1

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM | ACCT\Unit_Test\Drivers\ADM_ACCT_User_Update_dvr1.c
Special Conditions:

- Org "SAIC-2" exists with two users "jsmith" and "jdoe"
- User "jsmith" exists with update-user privilege
- User "jdoe" does not have update-user privilege

Input Data:
Requesting-user-id = "jdoe"
1. Change his phone# to 7035554444
2. Give himself create-org privileges
3. Change user "jsmith" phone number to 7035554444

Expected Outcome:
jdoe's phone# changed to 7035554444
Error changing create-org privileges, error message displayed
Error changing jsmith's phone#, error message displayed

Test Procedure Steps:

Analysis Procedures:

-

Unit Test Report

Unit ID: ADM_ACCT_update_user_account
UTP ID: ACCT_UPD_USER1
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 8/7/96

Purpose of Test:
Test on successful updating of a updateable field of the requester himself.
Test on unsuccessful updating of a field that requires special privileges.
Test on unsuccessful updating other user's info due to insufficient privileges.

Actual Results:

- User jdoe's phone has been changed to 7035554444
- Error message "User jdoe does not have the privilege to update the field." displayed on the screen when jdoe trying to change his create-org-priv.
- Error message "Do not have privilege to update user jsmith." displayed on screen when user jdoe trying to update jsmith's phone number.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): __________________ Date: _______________

Unit Test Plan

Unit ID: ADM_ACCT_update_user_account
UTP ID: ACCT_UPD_USER2
Version Control ID: SAIC V01-B01
Unit Function:
This function updates fields of user accounts. A user can update only a subset field of its own account unless the user has update-user privilege.
It may update entries in USER_ACCOUNT table.

Requirement to be validated:
. 3.2.2.1
. 3.2.3.1

Test Tools, Drivers, or Special Conditions:
Test Driver: \\PSAFER2\SAFER\SAIC SAFER\PingSAFER\V01-B03-01\ADM ACCT\Unit_Test\Drivers\ADM_ACCT_User_Update_dvr2.c
Special Conditions:

- Org “SAIC-2” exists with two users “jsmith” and "jdoe”
- User “jsmith” exists with update-user privilege

Input Data:
Requesting-user-id = “jsmith”
1. Change his own phone# to 7035554444
2. Change user “sdoe” phone number to 7035554444
3. Give user jdoe create-org privileges

Expected Outcome:
jsmith’s phone# changed to 7035554444
Error changing user sdoe’s phone number, error message displayed
jdoe has create-org privilege

Test Procedure Steps:
-

Analysis Procedures:

Unit Test Report

Unit ID: ADM_ACCT_update_user_account
UTP ID: ACCT_UPD_USER2
Version Control ID: SAIC V01-B01 Test Seq Number: 1
Test Conductor: Ping Liang Test Date: 8/7/96

Purpose of Test:
Test on successful updating of a updateable field of the requester himself.
Test on unsuccessful updating of non-exist user.
Test on successful updating other user's info with update-user privileges.

Actual Results:

- User jsmith's phone has been changed to 7035554444
- Error message "User sdoe does not exist, update failed." displayed on the screen when jsmith trying to update user sdoe
- User jdoe now has create-org privilege

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): __________________ Date: _______________

Appendix E

Subscription List Processor Unit Test Plans and Test Reports

Unit Test Plan

Unit ID: sdb_get_available_sub_id
UTP ID: SDB_GET_AVAlLABLE_SUB_ID01
Version Control ID: SAIC V01-B01
Unit Function:
This function reads through the subscription table in the database and returns the highest existing sub-id + 1

Requirement to be validated:

- 3.1.4.1

Test Tools, Drivers, or Special Conditions:

Input Data:
none

Expected Outcome:
The next available subscription-id

Test Procedure Steps:

- Query the database for the highest sub-id (select subscription-id from subscription)
- Call function and check answer against a manual sql query

Analysis Procedures:

-

Unit Test Report

Unit ID: sdb_get_available_sub_id
UTP ID: SDB_GET_AVAILABLE_SUB_ID01
Version Control ID: SAIC V01-B01 Test Seq Number: 01
Test Conductor: Curt Stapleton Test Date: 8/6/96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Manual query showed 6 to be the highest sub-id
- function returned 7 (ok)

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): __________________ Date: _______________

Unit Test Plan

Unit ID: sdb_get_subscription
UTP ID: SDB_GET_SUBSCRIPTION
Version Control ID: SAIC V01-B01
Unit Function:
This function selects a subscription record from the database given an intger value. It then puts that subscription record into a C data structure and returns that structure.

Requirement to be validated:

Test Tools, Drivers, or Special Conditions:

Input Data:
An integer value

Expected Outcome:
A C data structure for a subscription with id =input found in the database

Test Procedure Steps:

- Call function with existing subscription id
- Call function with negative subscription id
- Call function with 0 subscription id
- Call function with non-existent positive subscription id

Analysis Procedures:

-

Unit Test Report

Unit ID: sdb_get_subscription
UTP ID: SDB_GET_SUBSCRIPTION
Version Control ID: SAIC V01-B01 Test Seq Number: 01
Test Conductor: Curt Stapleton Test Date: 8/5/96

Purpose of Test:
The verify that the function executes as expected given valid and invalid input.

Actual Results:

- Returned correct subscription
- Returned subscription with id = 0 (correct action)
- Returned subscription with id = 0 (correct action)
- Returned subscription with id = 0 (correct action)

Reviewed By (SAIC): __________________ Date: _______________

Approved By (JHU/APL): _________________ Date: ______________

Unit Test Plan

Unit ID: sdb_get_sub_id_list
UTP ID: SDB_GET_SUB_ID_LIST01
Version Control ID: SAIC V01-B01
Unit Function:
This function reads all subscription ids found in the subscription table and returns a **linked** list of those ids.

Requirement to be validated:
. 3.1.4.2

Test Tools, Drivers, or Special Conditions:

Input Data:
none

Expected Outcome:
Correct list of subscription ids

Test Procedure Steps:

- Use sqlplus to select list of subscription ids (select subscription-id from subscription)
- Compare sql list to the list created and returned by sdb_get_sub_id_list

Analysis Procedures:

Unit Test Report

Unit ID: sdb_get_sub_id_list
UTP ID: SDB_GET_SUB_ID_LIST01
Version Control ID: SAIC V01-B01 Test Seq Number: 01
Test Conductor: Curt Stapleton Test Date: **8/5/96**

Purpose of Test:
To verify that the list of ids returned matches the list produced by a sql select command

Actual Results:

- function returns the correct list

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): __________________ Date: ______________

Unit Test Plan

Unit ID: sdb_insert_new_subscription

UTP ID: SDB_INSERT_NEW_SUBSCRIPTION01

Version Control ID: SAIC V01-B01

Unit Function:

This function adds a new subscription to the subscription database table and returns SLP_OK upon success.

Requirement to be validated:

- 3.1.4.1

Test Tools, Drivers, or Special Conditions:

Input Data:

a new subscription (C data structure)

Expected Outcome:

an enumerated value indicating success or failure

Test Procedure Steps:

- Call function with a subscription

Analysis Procedures:

--

Unit Test Report

Unit ID: sdb_insert_new_subscription

UTP ID: SDB_INSERT_NEW_SUBSCRIPTION01
Version Control ID: SAIC V01-B01 Test Seq Number: 01
Test Conductor: Curt Stapleton Test Date: 8/6/96

Purpose of Test:
To verify that the function executes as planned

Actual Results:
The function inserted the subscription as planned.

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): __________________ Date: _______________

Unit Test Plan

Unit ID: slp_create
UTP ID: SLP_CREATE01
Version Control ID: SAIC V01-B01
Unit Function:
This function is responsible for receiving subscription requests, validating those requests, adding the subscription to the database and sending baseline information if necessary.

Requirement to be validated:

- 3.1.4.1

Test Tools, Drivers, or Special Conditions:

Input Data:
packet header
request header
update header

Expected Outcome:
a BOOLEAN value reflecting the status of all operations performed, and functions called.

Test Procedure Steps:

- Test event-status ON branch
- Test event-staus OFF branch
- Call with valid view name in request (branch #3)
- Call with invalid view name in request
- Test with returned subscription-id = 0
- Test with returned subscription-id != 0
- Test with baseline-data-flag ON
- Test with baseline-data-flag OFF

Analysis Procedures:

-

Unit Test Report

Unit ID: slp_create
UTP ID: SLP_CREATE01
Version Control ID: SAIC V01-B01 Test Seq Number: 01
Test Conductor: Curt Stapleton Test Date: 8/5/96

Purpose of Test:
To test all branches and possible conditions and verify that function executes as planned.

Actual Results:

- Event status flag OFF branch is ok
- Event status flag ON branch is ok.
- Invalid view name branch is ok
- Valid view name branch is ok
- subscription-id =0 branch is ok
- subscription-id != 0 branch is ok.
- baseline-data-flaog ON is ok
- baseline-data-flag OFF branch is ok

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): __________________ Date: _______________

Unit Test Plan

Unit ID: sdb_slp_get_view
UTP ID: SDB_SLP_GET_VIEW01
Version Control ID: SAIC V01-B01
Unit Function:
This function retrieves a view record from the database and returns a data structure containing that view information.

Requirement to be validated:
. **3.1.4.1**

Test Tools, Drivers, or Special Conditions:

Input Data:
a view name

Expected Outcome:
A structure containing view data from the database

Test Procedure Steps:

- Call with existing view name (existing = in the db)
- Call with a non-existent view name

Analysis Procedures:

-

Unit Test Report

Unit ID: sdb_slp_get_view
UTP ID: SDB_SLP_GET_VIEW01
Version Control ID: SAIC V01-B01 Test Seq Number: 01
Test Conductor: Curt Stapleton Test Date: 8/6/96

Purpose of Test:
To verify that the function executes as planned.

Actual Results:

- Valid view name works ok
- Invalid view name returns null view (ok)

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Unit Test Plan

Unit ID: slp_listmov
UTP ID: SLP_LISTMOV01
Version Control ID: SAIC V01-B01
Unit Function:
This function receives an integer and calls a DB function to retrieve the subscription identified by the integer input. The subscription is returned.

Requirement to be validated:

-

Test Tools, Drivers, or Special Conditions:

Input Data:
integer representing subscription id

Expected Outcome:
The subscription with id = input integer

Test Procedure Steps:

- Call function with existing subscription id
- Call function with negative subscription id
- Call function with 0 subscription id
- Call function with non-existent positive subscription id

Analysis Procedures:

Unit Test Report

Unit ID: slp_listmov
UTP ID: SLP_LISTMOV01
Version Control ID: SAIC V01-B01 Test Seq Number: 01
Test Conductor: Curt Stapleton Test Date: 8/5/96

Purpose of Test:
The verify that the function executes as expected given valid and invalid input.

Actual Results:

- Returned correct subscription
- Returned subscription with id = 0 (correct action)
- Returned subscription with id = 0 (correct action)
- Returned subscription with id = 0 (correct action)

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ____________________ Date: ________________

Unit Test Plan

Unit ID: slp_monitor
UTP ID: SLP_MONITOR01
Version Control ID: SAIC V01-B01
Unit Function:

This function is responsible for comparing a new and an old snapshot and distributing it to interested subscribers.

Requirement to be validated:
. 3.1.4.2

Test Tools, Drivers, or Special Conditions:

Input Data:

new snapshot

Expected Outcome:

return BOOLEAN value indicating result of function calls.

Test Procedure Steps:

- Call slp_monitor with a valid new snapshot
- Call slp_monitor with an invalid snapshot

Analysis Procedures:

-
-
-

Unit Test Report

Unit ID: slp_monitor
UTP ID: SLP_MONITOR01
Version Control ID: SAIC V01-B01
Test Seq Number: 01
Test Conductor: Curt Stapleton
Test Date: 08/05/96

Purpose of Test:

Verify that the function executes as planned.

Actual Results:

- Executed and returned a BOOLEAN value
- Executed and returned a BOOLEAN value

Reviewed By (SAIC): __________________ Date: _______________

Approved By (JHU/APL): _________________ Date: ______________

Unit Test Plan

Unit ID: slp_start
UTP ID: SLP_START01
Version Control ID: SAIC V01-B01
Unit Function:
This function validates the event names that are listed in a subscription request

Requirement to be validated:

- 3.1.4.1

Test Tools, Drivers, or Special Conditions:

Input Data:
subscription request pointer
view pointer
event element list pointer

Expected Outcome:
a enumerated value indicating whether or not the event names listed in the subscription request are valid or not.

Test Procedure Steps:

- Call function with valid subscription request pointer
- Call function with invalid subscription request pointer
- Call function with valid view pointer
- Call function with invalid view pointer
- Call function with valid element list pointer
- Call function with invalid element list pointer
- Call function with subscription request with valid event names
- Call function with subscription request with invalid event names

Analysis Procedures:

Unit Test Report

Unit ID: slp_start
UTP ID: SLP_START01
Version Control ID: SAIC V01-B01 Test Seq Number: 02
Test Conductor: Curt Stapleton Test Date: 8/6/96

Purpose of Test:
To verify that the function executes as planned given various inputs.

Actual Results:

- Works with valid subscription request pointer
- Returns SLP_ERROR with NULL subscription request pointer
- Works with valid view name
- Returns SLP_ERROR with NULL view name
- Works with valid element list pointer
- Returns SLP_ERROR NULL element list pointer
- Returns SLP_OK given valid event names
- Returns SLP_ERROR given invalid event names

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): ___________________ Date: _______________

Unit Test Plan

Unit ID: slp_start
UTP ID: SLP_START01_1
Version Control ID: SAIC V01-B01
Unit Function:
This function validates the event names that are listed in a subscription request

Requirement to be validated:

- 3.1.4.1

Test Tools, Drivers, or Special Conditions:

Input Data:
subscription request pointer
view pointer
event element list pointer

Expected Outcome:
a enumerated value indicating whether or not the event names listed in the subscription request are valid or not.

Test Procedure Steps:

- Call function with valid subscription request pointer
- Call function with invalid subscription request pointer
- Call function with valid view pointer
- Call function with invalid view pointer
- Call function with valid element list pointer
- Call function with invalid element list pointer
- Call function with subscription request with valid event names
- Call function with subscription request with invalid event names

Analysis Procedures:

Unit Test Report

Unit ID: slp_start
UTP ID: SLP_START01_1
Version Control ID: SAIC V01-B01 Test Seq Number: 01
Test Conductor: Curt Stapleton Test Date: 8/6/96

Purpose of Test:
To verify that the function executes as planned given various inputs.

Actual Results:

- Works with valid subscription request pointer
- Crashes with NULL subscription request pointer
- Works with valid view name
- Crashes with NULL view name
- Works with valid element list pointer
- Crashes with NULL element list pointer
- Returns SLP_OK given valid event names
- Returns SLP_ERROR given invalid event names

Reviewed By (SAIC): ____________________ Date:________________

Approved By (JHU/APL): __________________ Date: _______________

Unit Test Plan

Unit ID: slp_tempwrite
UTP ID: SLP_TEMPWRITE01
Version Control ID: SAIC V01-B01
Unit Function:
Currently this function is responsible for receiving a pointer to a subscription and a pointer to a snapshot, and sending the snapshot to user defined in the subscription via the safety data manager.

Requirement to be validated:
. 3.1.4.2

Test Tools, Drivers, or Special Conditions:

Input Data:
subscription pointer
snapshot pointer

Expected Outcome:
function should call sdm with correct packet header, request header, and data request, and then return a BOOLEAN value.

Test Procedure Steps:

- Call function with valid subscription pointer and snapshot pointer
- Call function with invalid subscription pointer and valid snapshot pointer
- Call function with invalid snapshot pointer and valid subscription.
- Call function with invalid subscription pointer and invalid subscription

Analysis Procedures:

Unit Test Report

Unit ID: slp_tempwrite
UTP ID: SLP_TEMPWRITE01
Version Control ID: SAIC V01-B01 Test Seq Number: 02
Test Conductor: Curt Stapleton Test Date: 8/6/96

Purpose of Test:
To verify function behaves as planned, given each possible input.

Actual Results:

- Function works fine with valid inputs
- Function returns SLP_ERROR when given one or two null pointers

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): __________________ Date: _______________

Unit Test Plan

Unit ID: slp_tempwrite
UTP ID: SLP_TEMPWRITE01_1
Version Control ID: SAIC V01-B01
Unit Function:
Currently this function is responsible for receiving a pointer to a subscription and a pointer to a snapshot, and sending the snapshot to user defined in the subscription via the safety data manager.

Requirement to be validated:

- 3.1.4.2

Test Tools, Drivers, or Special Conditions:

Input Data:
subscription pointer
snapshot pointer

Expected Outcome:
function should call sdm with correct packet header, request header, and data request, and then return a BOOLEAN value.

Test Procedure Steps:

- 👍♋●● function with valid subscription pointer and snapshot pointer
- 👍♋●● function with invalid subscription pointer and valid snapshot pointer
- 👍♋●● function with invalid snapshot pointer and valid subscription.
- Call function with invalid subscription pointer and invalid subscription

Analysis Procedures:

Unit Test Report

Unit ID: slp_tempwrite
UTP ID: SLP_TEMPWRITE01_1
Version Control ID: SAIC V01-B01 Test Seq Number: 01
Test Conductor: Curt Stapleton Test Date: 8/5/96

Purpose of Test:
To verify function behaves as planned, given each possible input.

Actual Results:

- Function works fine with valid inputs
- Functions crashes when given one or two null pointers

Reviewed By (SAIC): ____________________ Date: ________________

Approved By (JHU/APL): __________________ Date: _______________