11.8 PROGRAM SOURCE CODE LISTINGS

Describe the procedures for obtaining all program source code listings contained in this system.

11.9 JOB STREAM LISTINGS

Describe the procedures for obtaining all job stream listings contained in this system. This will include all processes for normal operations and for all utility processes required to support the system.

II.10 DATA DICTIONARY/DIRECTORY/DBMS SCHEMA

The Data Dictionary/Directory is a tool for the management of data elements. It provides a central repository of information about each data element. The Data Dictionary/Directory defines the physical characteristics of a data element, the relationship among data elements, the source of the data, synonyms, and the processes or programs in the system using each data element. The document can help to assure timely, complete, and precise project documentation. The Data Dictionary/Directory is used by all the analysts, designers, programmers, and end users as a central data element information source. The Data Dictionary/Directory is designed to meet the following objectives:

* Establish a glossary of terms
* Provide standard terminology
* Define all terms associated with a system or subsystem
* Identify modules available to a system
* Provide a system-wide cross-reference capability
* Resolve problems associated with aliases and acronyms
* Provide a centralized control for system changes to data elements, programs, files, and reports
* Provide a listing of all data elements and their data feed sources.

DATA DICTIONARY/DIRECTORY/DBMS SCHEMA

The following information for each directory entry will be provided where and when appropriate.

1. Name - Alphanumeric name
2. Definition - Description of data
3. Format
4. Synonyms - User and technical
5. Type - Form, file, data base, program, data element
6. Source - Organization, medium, device, etc.
7. Recipients - Organization, medium, device, etc.
8. Where used - Reports, files, data bases, programs, on-line screens, etc.
9. Critical values - Range, scale, unit of measure, etc.
10. Editing criteria
11. Conversion factors
12. Volumes
13. Frequency - Updating and processing
14. Responsibility - For input, maintenance, output
15. Calculations - Required to calculate value
16. Controls
17. Audit requirements

III. TEST PLANS

III.1 UNIT TEST PLANS

Unit, or module testing, is the process of testing the individual subroutines, routines, or procedures in a larger program. It is aimed at the smallest component of the software product so that a "building block" approach can be used. Since unit testing occurs first in the overall testing process, thorough unit testing is critical in terms of efficient error detection. When designing test cases for a unit test, it is necessary to use the specification for the module, which describes its input and output parameters, and its processing logic. It is also necessary to have the source code. Unit testing is centered on those techniques which will uncover errors in the program logic. Since this kind of testing quickly becomes more difficult as the size of the test increases, unit testing provides the best opportunity to focus on the program's internal correctness and efficiency.

Since the development of each system is unique, the preparation of unit test plans will be customized to reflect the specific entities within the system. The plan will contain the testing procedures for each unit or module and the testing schedule proposed for this effort. The Unit Test Plan will be submitted for approval before actually conducting the testing process.

III.2 INTEGRATION TEST PLAN

The purpose of integration testing is to ensure that all related units or modules work together in accordance with the specifications. Testing in this area is aimed at data compatibility, the absence of module interface problems, and functional performance requirements. The scope of integration tests will include all programs which comprise a discrete function. Normally, this testing is conducted in an incremented manner.

The preparation of an Integration Test Plan will expand upon the Unit Test Plan in defining the testing procedures for the system. The plan will contain the testing procedures and schedules required to conduct the next higher level of testing. The format of this plan will follow the same format as used in the Unit Test Plan.

III.3 SYSTEM TEST PLAN

The purpose of the System Test Plan is to ensure to the software system developer that the new system or enhancement being implemented operates without disturbance, that its components work together, and its documentation and performance are current and up to standard, and that it meets the requirements which have been specified. The plan will prescribe the system developer's methodology for ensuring that the system is incrementally tested, and that test case selection conforms to a scheme which maximally exercises the system.

SYSTEM TEST PLAN

1. Objectives and Background

1.1 Purpose of the Test Plan
1.2 Project References

2. Test Overview and Schedule

2.1 System Description
2.2 Test Schedule
2.3 Equipment Requirements
2.4 Software Requirements
2.5 Personnel Requirements
2.6 Training Requirements
2.7 Material Requirements

3. System Test Specifications

3.1 Performance Requirements
3.2 System Function Tests
3.3 Test Methodology
3.4 System Test Constraints

4. System Test Descriptions

4.1 Test Sequence
4.2 Test Description
4.3 Test Inputs
4.4 Test Outputs
4.5 Control Information

5. System Test Procedures

5.1 Data Reduction
5.2 Test Initialization
5.3 Test Run
5.4 Completion

6. Evaluation Criteria

III.4 SYSTEM TEST ANALYSIS REPORT

The results of system testing are documentated in the System Test Analysis Report. The report is prepared by the software system developer conducting the system test. The report describes the results of each test identified in the System Test Plan. The report is reviewed and analyzed by a Quality Assurance (QA) group to ensure the completeness and accuracy of the system test. Upon approval of the report by the QA group, acceptance testing or regression testing of the system may begin.

SYSTEM TEST ANALYSIS REPORT

1. General Information

1.1 Summary
1.2 Environment
1.3 References

2. Test Results and Findings

2.1 Test Number 1

2.1.1 Dynamic Data Performance
2.1.2 Static Data Performance

2.N Test Number "N"

Include the two sub-sections shown in Section 2.1 for all specific tests conducted.

3. Software Function Findings

3.1 Function Number 1

3.1.1 Performance
3.1.2 Limits

3.N Function Number "N"

Include the two sub-sections shown in Section 3.1 for all specific tests conducted.

4. Analysis Summary

4.1 Capabilities
4.2 Deficiencies
4.3 Recommendations and Estimates to Resolve Deficiencies

5. Deficiency Correction Plan

5.1 Timing and sequencing
5.2 Re-testing
5.3 System Assurance
5.4 Re-test evaluation criteria
5.5 Re-test results

III.5 ACCEPTANCE TEST PLAN

The purpose of the Acceptance Test Plan is to establish the criteria on which the system will be accepted by the user and to define the organizational responsibilities for developing and conducting the test. The acceptance test will attempt to determine whether the software performs the functions for which it has been specified. The plan will address the organizational requirements for conducting and evaluating the acceptance tests and present a methodology for selecting test cases which will contribute to thorough testing.

The Acceptance Test Plan will contain the specific test cases that are required to verify and validate the various tests described in the plan. Multiple test cases may be required to verify a single test in the acceptance test plan, or a single test case may be used to verify multiple tests in the plan. In order to expedite acceptance test execution, test cases will be grouped together into run packages. Each package will contain forms that summarize the test steps and the results of the processing.

Before testing can be completed, it is necessary to compare the results of testing with predetermined results or "expected results". The expected results for each acceptance test case will be prepared when the test data is encoded. Therefore, it is necessary for the person developing the acceptance test data to also develop the expected results at the same time.

ACCEPTANCE TEST PLAN

1. General Information and Management Summary

1.1 Acceptance Test Overview
1.2 Environment
1.3 Acceptance Test Responsibilities
1.4 Organization of Acceptance Test Plan
1.5 References

2. Acceptance Test Approach

2.1 System Description (Inputs, Outputs, Processing)
2.2 Acceptance Test Milestones
2.3 Acceptance Test Organization

2.3.1 Schedule
2.3.2 Staff Responsibilities
2.3.3 Computer Resource Requirements
2.3.4 Materials
2.3.5 Training

3. Acceptance Test Specifications and Evaluation

3.1 Test Specifications

3.1.1 Functions to be Tested
3.1.2 Detailed Specifications for Software to be Tested
3.1.3 Manual Processes to be Tested
3.1.4 Test/Function Relationships
3.1.5 Test Progression

3.2 Acceptance Test Methods and Constraints

3.2.1 System Control Methods
3.2.2 Test Data Base Development
3.2.3 Test Data Development
3.2.4 Test Execution
3.2.5 Test Conditions
3.2.6 Extent of Testing

3.3 Evaluation

3.3.1 Criteria
3.3.2 Data Reduction

4. Acceptance Test Descriptions

4.1 Function 1 Test

4.1.1 Objectives
4.1.2 Test Input Parameters
4.1.3 Expected Results
4.1.4 Procedures for Execution and Evaluation
4.1.5 Test Controls
4.1.6 Additional Preparation Required

4.N Function "N" Test

Include all information contained in Sub-Sections 4.1.1 through 4.1.6 for each functional test conducted during the acceptance testing of the system.

III.6 ACCEPTANCE TEST ANALYSIS REPORT

The results of acceptance testing are documented in the Acceptance Test Analysis Report. The report is prepared by the end user organization conducting the acceptance test. It describes the status of the system after testing and identifies deficiencies and recommended solutions for review by OSMRE management, ISMS systems, the end user organization, and the QA group. Each test identified in the Acceptance Test Plan is described and evaluated against stated requirements in the Acceptance Test Analysis Report. The evaluation includes both efficiencies and deficiencies of system performance and functional processing. The deficiencies cited in the report must be corrected by the software system developer before the system is completely accepted by the end user organization.

ACCEPTANCE TEST ANALYSIS REPORT

1. Introduction

1.1 Summary - Describe the general functions of the functions tested and the expected results of the testing process. Define the criteria for a successful test.

1.2 Test Method - Describe the procedures and forms used in the testing process. Indicate any modifications of the approved test plan.

2. Test Results and Findings

Identify and present the results and findings of each test seperately in paragraphs 2.1 through 2."N".

2.1 Test 1

2.1.1 Dynamic Data Performance - Compare the dynamic data input and output results, including the output of internally generated data, of this test with the dynamic data input and output requirements. State the findings.

2.1.2 Static Data Performance - Compare the static data input and output results, including the output of internally generated data, of this test with the static data input and output requirements. State the findings.

2.N Test "N"

Present the results and findings of the second and succeeding tests in a manner similar to that of Section 2.1.

3. Software Function Findings

Identify and describe the findings on each function seperately in Section 3.1 through 3."N".

3.1 Function 1

3.1.1 Performance - Describe briefly the function. Describe the software capabilities that were designed to satisfy this function. State the findings as to the demonstrated capabilities from one or more tests.

3.1.2 Limits - Describe the range of data values tested, including both dynamic and static data. Identify the deficiencies, limitations, and constraints detected in the software during the testing with respect to this function.

3.N Function "N"

Present the findings on the second and succeeding functions in a manner similar to that of Section 3.1.

4. Analysis Summary

4.1 Capabilities - Describe the capabilities of the software as demonstrated by the tests. Where tests were to demonstrate fulfillment of one or more specific performance requirements, prepare findings showing the comparison of the results with these requirements. Assess the effects any differences in the test environment, as compared to the operational environment may have had on this test demonstration of capabilities.

4.2 Deficiencies - Describe the deficiencies of the software as demonstrated by the tests. Describe the impact of each deficiency on the performance of the software. Describe the cumulative or overall impact on performance of all detected deficiencies.

5. Conclusions and Recommendations

Based on the analysis of the findings, state the readiness of the system for implementation. Provide estimates of terms and effort required for the correction of noted deficiencies. Indicate the priority of each correction, in terms of urgency, and a suggested method to correct the deficiency.