



DEPARTMENT OF EDUCATION  
– FEDERAL STUDENT AID  
FINANCIAL MANAGEMENT  
SYSTEM

RELEASE 3.4 DESIGN TEST  
PLAN

Creation Date: 04/11/2001  
Last Updated: 03/21/2002 6:23 PM

## Document Control

---

### Reviewers

Name	Position
Paul Stonner	Client Project Manager
Anthony Dewitt	Test Manager
Karen Whistler	Transition Manager
Todd Elliott	Contractor Project Manager

---

### Distribution

Copy No.	Name	Location

## TABLE OF CONTENTS

1.	DESIGN TESTING OVERVIEW .....	1
2.	DESIGN TEST SCOPE AND EXPECTATIONS .....	2
3.	DT TEST SCHEDULE .....	7
4.	DT ROLES AND RESPONSIBILITIES.....	8
4.1.	TEST TEAM COORDINATOR .....	9
4.2.	PROGRAM WORKSTREAM.....	9
4.3.	MANAGEMENT OFFICE .....	9
4.4.	INTEGRATION WORKSTREAM .....	10
4.5.	TECHNICAL INFRASTRUCTURE.....	10
5.	EXECUTING AND VALIDATING TEST CASES .....	10
6.	ISSUE TRACKING AND PROBLEM DIAGNOSIS AND RESOLUTION.....	11
7.	TEST TRACKING AND DOCUMENTATION .....	12
8.	TEST DATA .....	13
9.	TEST SCRIPTS.....	13
10.	TEST CYCLES.....	13
11.	DESIGN TEST RESULTS .....	13
12.	CLIENT ACCEPTANCE APPROACH.....	13
13.	EXHIBITS .....	14
13.1.	EXHIBIT 1 UNIT AND INTEGRATION SAMPLE TEST SCRIPT TEMPLATE.....	15
13.2.	EXHIBIT 2 DEFINITIONS OF TEST CASE/SCENARIO ELEMENTS .....	16
13.3.	EXHIBIT 3 TESTING INCIDENT REPORT .....	18
13.4.	EXHIBIT 4 READINESS CHECK LIST .....	19
13.5.	EXHIBIT 5 ROLES AND RESPONSIBILITIES .....	17
13.6.	EXHIBIT 6 RESULTS MATRIX.....	21
13.7.	EXHIBIT 7 RELEASE MATRIX.....	22

## 1. DESIGN TESTING OVERVIEW

Design Testing (DT) is the functional test of the system prior to production, and validates the system's design through the simulation of actual business processes. Business functions and system functions (i.e. batch processing, system security features, etc.) are executed in a controlled environment using predefined test scripts to ensure integrity and confidence in the results. A Design Test also incorporates the new business procedures to validate that the system properly enables their execution. The system's business events are examined from the user or business perspective, rather than attempting to test every possible condition from the design perspective. The developer's unit testing approach allows for robust testing that exercises logical branches and conditions.

The goal and objectives of a Design Test - in addition to validating the design - are to confirm set ups, validate configurations, conversions, training material and ensure user requirements are met. The Design Test will also confirm cross-functional handoff points and processes. The testing should also identify and resolve all go-live issues. The deployment planning approach and assumptions should also be considered and captured during Design Testing.

There are many tasks, deliverables and procedures in the execution of a Design Test. This document provides the details of the activities and work products. Samples of work products are included in the exhibit section of this document. All scripts and work products will be placed under configuration management. Any changes and updates to scripts will follow CM guidelines.

Entry and Exit criteria have been established as management control gates for the design test process. Entry criteria have been captured and listed on the following checklist.

### Design Test Workstream Readiness Checklists:

Prior to entry into a Design Test there are a number of activities and events that must take place and documents that should be completed. These events and documents have been captured in the DT workstream readiness checklist and will be reviewed at the readiness session by the management office. Each workstream will be required to complete all relevant sections and submit for review. A sample of this document is displayed in the exhibit section. All sections of the checklist must be signed off as being completed or have signature approval *in lieu* of compliance.

### Exit Criteria for Design Testing:

Exit criteria can be summarized in the following points:

- All Design Test controlled unit and integration testing scripts and cycles have been executed successfully.
- All identified errors and defects, as well as issues and incidents, have been properly documented and worked through the resolution process.

- All reviews have been conducted and the reviews yielded satisfactory results.

## 2. DESIGN TEST SCOPE AND EXPECTATIONS

The FSA FMS Design Test strategy and approach will follow the FSA FMS Phase III release strategy listed in Appendix as exhibit 7 Release Matrix. The Phase III implementation schedule includes Design Testing in three phases. Each phase represents the test event for each release for the Phase III implementation. The scope of each phase of the Design Test has been defined based on several factors. Those factors include the Phase III implementation schedule, changes with redefined scope of functionality, and external GAPS / ED CFO schedule dependencies. The following describes in detail the scope of each testing phase.

The FMS Phase III Design Test I scope includes testing the following program interfaces:

- Direct Loan Consolidation
  - The design test scripts will include testing of only **Non GAPS** transactions interfaced to the FMS General Ledger and also the export to the ED CFO Oracle instance. The transaction files will contain “unbooked” loan data.
- Direct Loan Origination
  - The scope of this cycle of DLO testing will test the connection points of the Loan Origination feeder system to FMS. The test scripts will test the file transfer “pass through” of Loan Origination files directly to GAPS. The file format will be in Pre-October GAPS format and will not be altered.
  - The design test scripts will include testing of only **Non GAPS** transactions interfaced to the FMS General Ledger and also the export to the ED CFO Oracle instance. The transaction files will contain “unbooked” loan data.
- Direct Loan Servicing
  - The scope for DLS will include the processing of booked loans, payments, adjustments, and refunds from IF10 transfer files. This financial data will be imported into the FMS General Ledger and verified.
  - G Records containing FAD Deposits and FIG Interagency transfers transactions will also be imported into the FMS GL in DT 1.
  - IF20 records (i.e. unapplied/misdirected payments) will also be processed.
- LEAPP/SLEAPP
  - The performance reports of the LEAPP/SLEAPP will be tested.
- FFEL Lender
  - The scope of FFEL lender will be to process collection and disbursement data up to the point of loading the transactions into the FMS instance and exporting that file out into the format for transmitting to FMSS.

Integration testing of loading the file into FMSS will be tested in the next test phase within the end-to-end integration test scripts.

- Debt Collections
  - The scope of the Debt Collections Services (DCS) interface is to allow accounting transactions to be created in FSA FMS general ledger from the Collections and disbursements (Accounting) files generated by Raytheon twice a week. Included in this scope is the conversion of the old PAS CAN account structure into the current FSA Account Code Structure.

In addition, the Design Test will execute regression testing for the FFEL GA program to ensure that no effects in functionality are found due to multi-org configuration. Design Test I will also include creating the export file containing summary data of test script transactions. This file will eventually flow to OCFO FMSS.

The FMS Phase III Design Test II scope includes testing of the following program interfaces:

- PELL
  - Unit Testing (the FMS team will set-up, control, and walk FSA CFO users through these tests):
    - Budgetary Journal Entries (from ED CFO FMSS)
    - School data file load from Pell RFMS
    - Obligations from RFMS (OB-only)
    - Obligate & Pay from RFMS (OB/PY)
    - Deobligations from RFMS (-OB)
    - Expenses from RFMS (EX and -EX) - if the development of this interface is completed and the testing can be coordinated with Pell
    - Error handling (at a batch level as well as at a transaction/ACCS level)
    - Funding Year expiration/"closing" process
  - Integration Testing (this testing will be derived from the 1st cycle of the 4 test/transaction cycles that Pell is planning for end-to-end testing with GAPS and FMSS and the FMS team will have to work closely with both FSA CFO and the Pell team):
    - Budgetary Journal Entries
    - School data file load from Pell RFMS
    - Obligations (OB-only)

- Obligate & Pay (OB/PY)
  - Deobligations (-OB)
  - Expenses (EX and -EX) - if the development of this interface is completed
- Direct Loan Origination
  - Design Test II will process “unbooked” loan transactions and excess cash into the FMS GL, and payment requests transactions into FMS AP and create the export file in the GAPS format. Actual transfer of data to the GAPS system will take place in DT III.
- Direct Loan Consolidation
  - Design Test II will process “unbooked” loan transactions and excess cash into the FMS GL.
  - Design Test II will incorporate testing of GAPS transactions. Obligations, Payment, De-obligations, Refunds, and Excess Cash transactions will be interfaced to FMS AP and GL and interfaced out to a GAPS formatted file. The actual transfer of data to the GAPS system will take place in DT III.
- Campus Based
  - The scope for the Campus-Based program will include processing of the following
    - Budgetary Journal Entries
    - FSEOG, FWS, and Perkins Loan award obligations (positive/negative)
    - Teacher Cancellation Payments
    - FISAP Expenditures
    - Perkins Loan Balance Sheet Data
  - Create export file in GAPS format for Teacher Cancellation payments. The actual transfer of file to GAPS system for testing will take place in next testing phase.
- GAPS
  - The scope of this testing will incorporate the creation of interface files of Pell, CB, DLO, and DLC data into GAPS format.
- LEAP/SLEAP
  - Application/Award/Reallocation (LEAP)
  - Application/Award/Reallocation (LEAP, SLEAP Tier 1)
  - Application/Award Process/ Reallocation (LEAP, SLEAP Tier 1, Tier 2)
- Regression Tests of all programs.

The FMS Phase III Design Test III scope includes testing of the following program interfaces:

- Pell
  - These tests will be the continuation of the end-to-end testing with GAPS initiated during the Design Test II Integration Testing; that is, the continuation of the 1st test cycle as well as the execution of the test cycles 2, 3, and 4; so the transactions above will be tested/retested as well as the following GAPS-initiated transactions:
    - Drawdowns (DD)
    - Treasury Confirmations (TC; of both Payments (PY) and Drawdowns (DD))
    - Returns (RE)
    - Return Confirmations (FR)
    - Refunds (RF)
    - Refund Reversals (-RF)
    - Upward and Downward Adjustments (-AD and AD)
    - Unprocessed Deobligations (-UD)
    - Processed Deobligations (PD)
    - Disbursement Offsets (DF) - requires coordination with GAPS; these are rare so they may not be tested
    - Perkins Adjustments (PK) - also requires coordination with GAPS; also don't occur very often so they may not be tested
    - GAPS transactions that do not have an accounting impact in FMS (Obligation Confirmations (OB), DUNS number changes (DB), Error transactions (ER), etc)
  
- Campus Based
  - These scripts will incorporate the complete “end to end” process flow of transactions. The GAPS-initiated transactions include:
    - Drawdowns (DD)
    - Treasury Confirmations (TC; of both Payments (PY) and Drawdowns (DD))
    - Returns (RE)

- Return Confirmations (FR)
- Refunds (RF)
- Refund Reversals (-RF)
- Upward and Downward Adjustments (-AD and AD)
- Unprocessed Deobligations (-UD)
- Processed Deobligations (PD)
- Disbursement Offsets (DF) - requires coordination with GAPS; these are rare so they may not be tested
- Perkins Adjustments (PK) - also requires coordination with GAPS; also don't occur very often so they may not be tested
- GAPS transactions that do not have an accounting impact in FMS (Obligation Confirmations (OB), DUNS number changes (DB), Error transactions (ER), etc)
- Direct Loan Origination
  - Design Test III will process “unbooked” loan transactions and excess cash into the FMS GL
  - Payment requests transactions will be loaded into FMS AP and Refund transactions into GL
  - Create the export file in the GAPS format and transferred to GAPS for processing.
  - Process of GAPS acknowledgements into FSA FMS instance
  - Creation of acknowledgement file to be returned to feeder system
- Direct Loan Consolidation
  - Design Test III will process “unbooked” loan transactions and excess cash into the FMS GL.
  - Design Test III will incorporate testing of GAPS transactions. Obligations, Payment, De-obligations, Refunds, and Excess Cash transactions will be interfaced to FMS AP and GL and interfaced out to a GAPS formatted file and transferred to GAPS for processing.
  - Process of GAPS acknowledgements into FSA FMS instance
  - Creation of acknowledgement file to be returned to feeder system.
- Direct Loan Servicing
  - Process transactions via Close process to OCFO FMSS
- Integrated FSA “Close” to ED OCFO FMSS
- Regression Tests of All programs.

Design Test III incorporates the true “end to end” testing with representation and participation of the GAPS and ED CFO OCFO FMSS organizations. This test phase will incorporate all Phase III interfaces.

All Design Test phases will include unit test scenarios by program and integration test scripts for cross-program business processes such as the “Close” process. Oracle’s Multi-Org functionality will also be tested within unit and integration test scripts.

Unit and integration test cycles for the application development work stream must be completed prior to the start date of each Design Test phase. This checkpoint has been included in the readiness checklist.

Three weeks have been allocated for the execution of the Design Test I and resolution of issues. Design Test II will be scheduled for 3 additional weeks of testing. Given the external GAPS / ED CFO schedule dependencies, the exact end date for Design Test II is still to be determined. A Design Test III will be scheduled for 2 additional weeks for integration “end to end” testing.

### 3. DT TEST SCHEDULE

Execution of Design Testing will occur from June 25, 2001 through July 13, 2001. A Test Readiness Review will be held prior to the start date of testing to verify all entry criteria have been met. The test execution phase involves executing each of the test cases and comparing the actual results with the expected results. Execution of the cases may sometimes occur more than once during the test execution phase, since discrepancies will be identified and fixed during this phase. Unit and integration test cases will be executed until no discrepancies are found or other resolutions or work-arounds are identified. Review and analysis of DT Test Results will be conducted on a daily basis. The FMS program leads will perform the acceptance testing (PAT) for the development inventory prior to code migration to the test instance. The chart below provides details on the Design Test schedule. This chart is subject to change at management discretion. All official dates will be reflected in project plan.

ID	Task Name	Start Date	End Date
	<b>Product Test Schedule</b>		
1	DT Planning	28-MAY	10-JUN
2	Logistics/Infrastructure	04-JUN	13-JUN
3	Design Documentation	28-MAY	14-JUN
4	Integration Test Planning	31-MAY	14- JUN
5	Unit Test Planning	31-MAY	14-JUN
6	FMS PAT of Development Inventory	11-JUN	16-JUN
7	Instance Creation and Preparation	18-JUN	22-JUN
8	Pre-DT Reviews	19-JUN	20-JUN
9	DT I Test Readiness Review	22-JUN	22-JUN

<b>ID</b>	<b>Task Name</b>	<b>Start Date</b>	<b>End Date</b>
10	DT Execution	25-JUN	13-JUL
11	Design Testing Kickoff Meeting	25-JUN	25-JUN
12	DT Unit Testing	25-JUN	13-JUL
13	CFO – Participation	27-JUN	13-JUL
14	DT Integration Testing	1-JUL	13-JUL
15	DT I Closure – Exit Meeting	13-JUL	13-JUL
16	Design Test II Planning	2-JUL	18-JUL
17	Design Test II Program Sample Data	23-JUL	23-JUL
18	Design Test II Scripts Final Review	2-AUG	2-AUG
17	DT II Instance Creation and Preparation Freeze Date of mapping and rule dependencies	1-AUG	5-AUG
18	Design Test II Test Readiness Review	3-AUG	3-AUG
19	DT II Execution	6-AUG	24-AUG
20	Design Testing Kickoff	6-AUG	6-AUG
21	DT II Unit Script Testing	6-AUG	24-AUG
22	CFO – Participation	13-AUG	24-AUG
23	DT II Integration Script Testing	20-AUG	24-AUG
24	DT II Closure – Exit Meeting	24-AUG	24-AUG
25	Design Test III Scripts Final Review	6-SEP	6-SEP
26	DT III Instance Creation and Preparation Freeze Date of mapping and rule dependencies	2-SEP	7-SEP
27	Design Test III Test Readiness Review	7-SEP	7-SEP
28	DT III Execution	10-SEP	21-SEP
29	Design Testing III Kickoff Meeting	10-SEP	10-SEP
30	DT III Integration “E2E” Script Testing	10-SEP	21-SEP
31	CFO – Participation	10-SEP	21-SEP
32	Program Participation	10-SEP	21-SEP
33	GAPS/ ED CFO FMSS Participation	10-SEP	21-SEP
34	DT III Closure – Exit Meeting	21-SEP	21-SEP

#### **4. DT ROLES AND RESPONSIBILITIES**

Staffing for DT involves both FSA FMS Implementation Team and FSA personnel. The primary roles for the Design Test effort include Test Coordinator(s), Program Functional Workstream, Application Development and Support (AD&S), Technical Infrastructure, Integration Workstream, Management office and others. Complete details of ownership have been documented in the Roles and Responsibility matrix. A summary of roles and responsibilities are described below:

#### **4.1. TEST TEAM COORDINATOR**

- Develop and maintain the Design Test plan and Design Test work products.
- Develop Design Test status reporting.
- Define issue resolution and recovery approach.
- Review test scripts inventory, scenario test cases and test data.
- Work with the program leads in coordinating test scripts to minimize cross-team testing conflicts.
- Develop, review and utilize Design Test readiness checklist to ensure needed application set up or configuration requirements are coordinated and performed.
- Coordinate all test activities for the team according to the test plan.
- Assist in evaluating and recording test results.
- Provide feedback and progress reports to management office regarding status of product test activities and significant issues.
- Assist in review of regression testing to validate changes to the application software and/or configuration. Regression testing will consist of testing the programs and affected sub-systems following the multi-org modification.

#### **4.2. PROGRAM WORKSTREAM**

- Development of the Design Unit and Integration Test Scripts.
- Execute the tests.
- Evaluate test results.
- Record all incidents and problems encountered during testing activities.
- Conduct regression testing to validate changes to the application software and/or configuration.

#### **4.3. MANAGEMENT OFFICE**

- Participate in the walkthrough of the Design Test Plan.

- Review all incidents and problems encountered during testing activities.
- Monitor the testing process to identify problems, mitigate potential risks and schedule slippage.
- Review test results with the test team to clarify questions, concerning system functionality and discrepancies with expected results, and ensure that the design testing activity was a valid and complete exercise.
- Confirm that the testing process is comprehensive in scope and is complete.
- Review and determine that Design Test Exit Criteria have been met.

#### **4.4. Integration Workstream**

The integration workstream is comprised of a cross section of program team members, management office, and AD&S.

#### **4.5. Technical Infrastructure**

The infrastructure team is responsible for the Instance Creation and preparation as well as System support during execution of the Design Test. The Design Test landscape should mirror the production landscape as much as possible with respect to architecture. All configuration management procedures should be followed for all code changes and migrations to the test environment. All changes or updates to the testing environment will require change requests under configuration management guidelines.

### **5. EXECUTING AND VALIDATING TEST CASES**

A test case or scenario is a discrete, executable activity that will return a predictable result. The scope of the test cases/scenarios to be included will be identified through the scenario description element on the scenario test matrix or the overview section in the test script file. Test cases are made up of a setup, test description, and expected result sections (see Exhibit 1). A test case is designed to verify the functionality of a business scenario, and describes step-by-step how the business scenario is being tested.

Executing the test cases is the responsibility of the entire test team, as coordinated by the Test Team Coordinator. Any team member performing a test will update the appropriate fields of the test script matrix when conducting Design Testing.

Validation is defined to be the comparison of the actual results to the expected results. If the actual data results match the expected results then the test script passes. FSA CFO and FSA FMS personnel will be asked to initial test scripts that passed the controlled test script.

## 6. ISSUE TRACKING AND PROBLEM DIAGNOSIS AND RESOLUTION

The test team member conducting the test will record any incidents or problems that are related to product testing. An analysis of the incident will be performed to determine if the problem was caused by:

- Problem(s) with the test data;
- Problem(s) in the test environment;
- An incorrectly run test script;
- A misunderstanding of what the expected result should be.

If necessary, the incident will be forwarded to the business requirements, software architecture or technical support team for assistance in this analysis.

Incidents or problems will be recorded using the “Design Testing Incident Report” (see Exhibits). The “Design Testing Incident Report” is designed to capture as much information as possible to relay to the business requirements, software architecture or technical support team. This information will enable recreation of the situation. Test team members completing a “Design Testing Incident Report” should:

1. Write down a description of the symptoms that occurred and the test execution steps leading up to them, noting the date and time of the discrepancy.
2. Print any screen(s) or report(s) that illustrate the error that is occurring.
3. Escalate the problem to the Test Team Coordinator immediately if this is a “show stopper” that prevents continuation of the test.
4. If the problem isn’t severe, log the information and continue with the same or other tests.

A “discrepancy” is a difference between what was *expected* to happen and what *actually* happened. Discrepancies are initially evaluated to ensure that they represent system defects. The initial evaluation of discrepancies includes:

- Review of the test data;
- Review of the test environment;
- Review of test cases;

- Review of actual system functioning.

If it is determined that the “discrepancy” represents a system defect, then the incident will be forwarded to the business requirements, software architecture or technical support team for assessment and proposed solution(s). The proposed solution(s) may be technical/development or business process based (or both). Further, some solutions will entail significant technical or development work whereas others will not (and likewise for business process-based options). Once the solution options have been identified, the Test Team Coordinator will work with project management in determining which option will be followed or if the resolution will be deferred.

Standard procedures for development, unit test, code walkthrough and migration will be followed for any solution for which technical/development work is included. The solution will be tested as part of the Design Test, and regression testing as needed will be conducted for the systems and processes impacted by the modification. After the solution has been successfully tested, the incident will be designated as “closed” by the Test Team Coordinator.

## **7. TEST TRACKING AND DOCUMENTATION**

The actual test results will be compared to the expected results to determine if the test ran correctly (or incorrectly). At the completion of each test activity the Test Team member will evaluate the status of the activity as “Pass”, “Retest”, “Fail” or “Blocked”. A status of “Pass” indicates that the expected results were achieved. A status of “Retest” indicates that the actual results do not match the expected results for reasons that can be corrected and retested. A status of “Fail” indicates that actual results do not match expected results and the errors cannot be corrected during the Design test. A status of “Blocked” indicates that the test cannot be executed because of missing system components. Explanations must be provided for all cases with status of “Blocked”. Design Testing is considered complete when each test case has a status of “Pass” or “Blocked,” and the Test Team Coordinator and the Management Office have reviewed the results.

Each scenario or combination of scenarios will be tracked via several checkpoints consisting of key processes. Each checkpoint will be measured on a Red, Yellow, or Green Basis:

- Green – The checkpoint produced the expected results
- Yellow – The checkpoint did not produce the expected results or raised major issues, however, there is an identifiable solution that can be implemented prior to “Go Live”
- Red – The checkpoint could not be completed or did not produce the expected results for unknown reasons that cannot be resolved prior to “Go Live”

Swat teams will be created and disbanded as needed by the management office to resolve critical issues.

## **8. TEST DATA**

Sample data will be gathered from the programs. Each program will be responsible for creating sample data files or using a subset provided by the feeder systems.

## **9. TEST SCRIPTS**

A test script is a collection of related test cases, put into a sequence of steps. The Test Team Coordinator, the management office, and the integration test team will review all unit test and integration test scripts. Any test scripts that have been determined to have dependencies on other scripts will have the appropriate sequencing documented in the prerequisite test column. All test scripts should be placed under configuration management upon review by the Test Team Coordinator and management office.

## **10. TEST CYCLES**

A test cycle is a collection of test scripts put into a sequence. A test cycle describes how a group of test scripts will be executed. Dependencies between scripts should be identified and recorded by FMS FSA functional program leads. The FMS PMO will be responsible for the overall management, scheduling and review of Design Test execution.

## **11. DESIGN TEST RESULTS**

The Design Unit Test Results will be included and recorded on the Summary Status Results Matrix. All results will be summarized and presented to management office for review and submission as a Design Test Exit Criteria checkpoint.

## **12. CLIENT ACCEPTANCE APPROACH**

The acceptance of Phase III will be based on the review and acceptance of the Design Testing results. These results provide confirmation that the transactional data was processed into the current FMS instance and interfaced to ED CFO correctly. The FSA CFO and FSA CFO project Lead will provide production release approval contingent upon this acceptance.

## 13. EXHIBITS

**13.1. Exhibit 1 Unit and Integration Sample Test Script Template.**

<i>Scenario #</i>		<i>Date Tested</i>		<i>Pass/Fail</i>		<i>Acceptance Signoff</i>									
<i>Scenario Name</i>							<i>Comments</i>								
<i>Scenario Description</i>															
<i>Team</i>	<i>Test Step/Action</i>	<i>Process</i>	<i>Requirement</i>	<i>Proc Doc X-Ref</i>	<i>Training Doc X-Ref</i>	<i>Navigation Path</i>	<i>Input Data Description</i>	<i>Prerequisite Test Case(s)</i>	<i>Expected Results</i>	<i>Actual Results</i>	<i>Test Case Owner</i>	<i>Sched Date</i>	<i>Actual Date</i>	<i>Pass/Fail/Not Signoff</i>	<i>Test Issues/Notes</i>

## 13.2. EXHIBIT 2 – DEFINITIONS OF TEST CASE/SCENARIO ELEMENTS

**Scenario number:** <Program acronym or 1<sup>st</sup> 2 letters of program> + number

**Scenario name:** Name of scenario. Name should reflect business process being testing.

**Scenario Comments:** Identify the test's general business process area.

**Scenario Description:** Describe briefly the objective of the test, such as to verify that something works a certain way, or to ensure that invalid data is identified, etc.

**Interfaces:** Identify any interfaces that are needed for the test.

**Reports:** Describe reports or queries to be used to help in this test or in analyzing the results of the test.

**Set-up:** Document any applications setup requirements. Also document any seed data requirements that might be needed for this test.

**Data Sheet (Separate Tab):** Document all required information of input data. Provide information such as input file names from feeder system. Include explanation of content of data. Include any infrastructure requirements for transfer/storage of data.

Sample information include:

- A sample <name or type of> interface file with ### records with no anticipated errors
- A sample <name or type of> interface file with ### records with an error in the GAPS Award ID field (specify invalid data or invalid value)
- Etc.

**Test Step / Action:** Describe the anticipated test procedures or steps, as detailed as is necessary to adequately complete the test.

**Navigation:** Include all Oracle Navigation paths and necessary parameters.

**Expected Results:** To be completed before the test is run. Describe the expected outcome of the test. This can be included on separate tab if more space is needed

**Proc-Doc Xref:** Reference to process documentation related to this step in the script if any.

**Test Case Owner:** Person performing the test.

**Date:** Date test step executed.

**Actual Result:** Describe whether the test resulted as expected or not; reference output documents, screen shots, etc., to provide documentation of results. This can be documented as a separate sheet.

**Test Issues/Notes:** List related test notes to be recorded.

**Requirement #:** Enter requirement from requirements matrix, if applicable.

**Prerequisite Test Case(s):** List other unit or integration test scripts that should be executed, if any, prior to this step.

**Comments and Other Info:** (as needed)

**13.3. EXHIBIT 3 - TESTING INCIDENT REPORT**

<b>DESIGN TESTING INCIDENT REPORT</b>	
INCIDENT REPORT #:	DATE:
Name:	Phone Number:
Subsystem/Functional Area(s) affected:	Test Case #:
Problem with (check all applicable): <input type="checkbox"/> Environment <input type="checkbox"/> Software <input type="checkbox"/> Other _____	Activity (select one): <input type="checkbox"/> Design Test <input type="checkbox"/> Other _____
Severity of Problem: <input type="checkbox"/> Severity 1 – Major problem, system doesn't respond or crashes, loss of data, feature(s) inoperative or will not execute <input type="checkbox"/> Severity 2 – Major feature halts, incorrect results after execution, other severe restrictions <input type="checkbox"/> Severity 3 – Expectations of major feature not met, problem is obstacle that can be worked around <input type="checkbox"/> Severity 4 – Feature executes correctly but minor cosmetic change required to meet test objectives	
<input type="checkbox"/> Check if Recurring Problem (explain):	
Description of Problem:	
Possible Cause/Solution:	
<b>FMS FUNCTIONAL ASSESSMENT</b>	
(This activity to be performed by member of Phase III Business Requirements staff)	
Assigned To:	Date:
Assessment:	
Recommended Solution(s):	
<b>FMS SOFTWARE ARCHITECTURE/TECHNICAL ASSESSMENT</b>	
(This activity to be performed by AD&S or Technical Infrastructure staff)	
Software Architecture/Technical Assessment Performed by:	Date:
Assessment:	
Software Affected:	
Solution(s):	
<b>CORRECTIVE ACTION</b>	
Resolved by:	Date Completed:
Corrective Action Taken:	
<b>TEST OF CORRECTIVE ACTION</b>	
Date:	Results: <input type="checkbox"/> Pass <input type="checkbox"/> Retest <input type="checkbox"/> Fail <input type="checkbox"/> Blocked
Tester Signature:	

13.4. Exhibit 4 Readiness Check list

Phase III Release CRP Workstream Readiness Checklist							
Workstream:							
							
Deliverables	Owner(s)	Reviewer(s)	Due Date	Signoff*	Signoff Date	Notes/Document References	
1) CRP Participants							
A. Participants Identified							
B. Participants Notified of CRP Schedule							
C. Superuser Mentors Identified and Notified							
2) Design Documentation							
A. Master Process List Updated							
B. Requirements Matrix Updated							
C. Process Flows Completed							
D. CE Documents Approved							
3) Issues							
A. Phas III Issues Resolved & Updated in SFA Tracker							
4) Unit Test Scripts							
A. Scripts Defined (incl. Data & Expected Results)							
B. Dependencies Documented (intra- & inter-WS)							
C. Testers Assigned							
D. Data Requirements Identified							
E. Interface & Extension Execution Requirements Identified and Communicated to Tech Team							
F. Training Procedures & Navigations Linked to Tests							
5) Integration Test Scripts	To Be Developed by Integrated Testing Team						
A. Test Scenario(s) Defined							
B. Test Flow(s) Developed							
C. Testers Assigned							
D. Data Requirements Identified							
6) Stress Test Requirements	Not Applicable for this CRP						
A. High-frequency Processes Identified							
B. Stress Test Scripts Created							
C. Locations, Users & Frequencies Identified							
7) Configuration Guides							
A. Configurations Updated							
B. Module-specific Configurations Updated							
C. DFF Configurations Updated							
8) Client Acceptance Tests							
A. Conversion CATs Completed							
B. Interface CATs Completed							
C. Extension CATs Completed							
D. Reports/Queries CATs Completed							
9) Baseline Instance							
A. Baseline Updated for Creation of CRP Instance							
10) Responsibilities & Menus							
A. Custom Responsibilities & Menus Identified							
B. CRP Participants Mapped to Responsibilities							
C. Information Communicated to Tech Team							
11) CRP Instance							
A. CRP-specific Configurations Completed							
B. Manual Data Entered							
12) Pretesting							
A. Unit Test Scripts Pretested							
B. Responsibilities & Menus Pretested							
13) Training Documents							
A. Procedures Completed							
B. Navigations Completed							
14) CRP Binders							
A. Workstream CRP Binders Completed							
B. Superuser Binders Completed							
C. Binders Reviewed							
<b>Certification</b>							
Based on a review of the items listed above, we have determined that:							
<input type="checkbox"/> The Workstream is ready to begin the R1W2 CRP							
<input type="checkbox"/> The Workstream can begin the R1W2 CRP with the following conditions:							
<input type="checkbox"/> The Workstream is not ready to begin the R1W2 CRP due to the following reasons:							
<b>Signoffs</b>							
SFA FMS Business Workstream Lead					Date		
FMS ITS Workstream Lead					Date		
KPMG Workstream Lead					Date		
* Signoff signifies that							
1) The Reviewer considers the item to be sufficiently complete to begin the CRP for the agreed upon system functionality to be tested							
2) The process designs have been reviewed and agreed to with respect to what is to be tested and how							

Page 1

### 13.5. Exhibit 5 Roles and Responsibilities

  <p style="text-align: center;">Student Financial Assistance U.S. Department of Education <i>We Help Put America Through School</i></p>							
Item #	Task	Functional Workstream	AD&S	Tech. Infrastructure	Integration Workstream	Management Office	Other
O=Owns, P=Participates, C=Coordinates, R=Reviews							
<b>CRP Planning</b>							
1	Create Master CRP Schedule & Publish	R	R	R	O	R	
2	Assign CRP Participants	O			C	R	
3	Assign TI & ADS Support Personnel		O		C	R	
4	Create CRP Participant Roster	R	R	R	O	R	
5	Plan Team Event(s)				C	R	CM Owns
6	Develop CRP Status Reporting Approach				O	R	
7	Define Issue Resolution & Recovery Approach				O	R	
8	Create CRP Kickoff Presentation				O	R	
<b>Logistics/Infrastructure</b>							
9	Determine Number of Rooms Required				O		
10	Identify and Schedule Rooms				O		
11	Finalize Team/Room Assignments	R	R	R	O		
12	Identify CRP TI Requirements (PCs, Ports, Printers, etc.)	O		P	C		
13	Setup & Test PCs, Ports, Printers, etc.			O	C		
<b>Integration Test Planning</b>							
14	Assign Integration Test Point-persons	O			C		
15	Identify Integration Test Requirements	R	R	R	O	R	
16	Develop Integration Test Scripts	P	P	P	O		
17	Determine I&E Processing Requirements	P	P	O	C		
18	Assign Integration Testing Team	O			C		
19	Publish Integration Testing Schedule	R	R	R	O	R	
<b>Unit Testing Planning</b>							
20	Develop Unit Test Scripts for New Functionality	O			R		
21	Identify Prior Release Unit Tests for Regression Testing and Modify as Necessary	O			R		
22	Create Unit Test Schedule	O			R		
23	Assign and Schedule Unit Test Executors	O			R		
<b>Design Documentation</b>							
24	Update Requirements Matrix	O			R		
25	Update Master Process List	O			R		
26	Complete CE Documents	O			R	R	
27	Complete Functional Specs	O			R		
28	Complete Technical Specs	P	O		R		
29	Complete Batch (Maestro) Schedule Requirements	O		P	R		
30	Deployment Site Infrastructure Reqs (PCs, Printers, etc.)	O		P	C		
31	Define Performance/Stress Requirements	P	P	P	O		
32	Update Development Inventory	P	O		C		
33	Update Configuration Guides	O			R		
34	Update Business Processes (Level 4)	O			R		
35	Update System Flows	P			O		
36	Update Data Flows	P			O		
37	Update User Responsibility Matrix	O			C		
38	Final Design Review Completed	P			O		
39	Complete Release (Features) Document	P			O	R	
<b>Training Documentation &amp; Planning</b>							
40	Develop Procedures & Navs (New Functionality)	O					TR Coordinates
41	Update Procedures & Navs (Existing Functionality)	O					TR Coordinates
42	Develop Release Training Plan	R					TR Owns
<b>CAT</b>							
43	Complete CAT for Development Items (incl. Transfers)	O	P		C		
44	Document and Resolve CAT Issues	O	P		C	R	
45	Approve Development Items for CRP	O	O		C		
46	Move Development Items to CRP	P	O		C		

# Measuring DT Test Results

## 13.6 Exhibit 6 Results Matrix

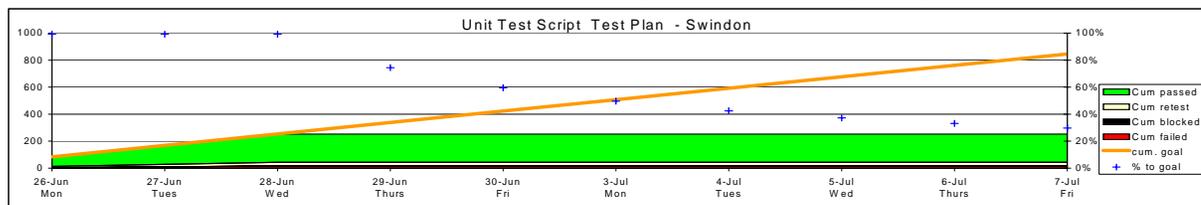
## Tracking Unit Tests

SPCD Unit Testing - Swindon

Enter items in Blue boxes

	Total scripts to test	Total Retests Required		Mon 26-Jun	Tues 27-Jun	Wed 28-Jun	Thurs 29-Jun	Fri 30-Jun	Mon 3-Jul	Tues 4-Jul	Wed 5-Jul	Thurs 6-Jul	Fri 7-Jul	Cumulative Total	
Purchasing	124	1	tested	12	12	12	0	0	0	0	0	0	0	36	
			goal	12	12	12	12	12	12	12	12	12	12	29%	
			passed	10	11	12	0	0	0	0	0	0	0	0	33
			Retest	1	0	0	0	0	0	0	0	0	0	0	1
			blocked	0	0	0	0	0	0	0	0	0	0	0	0
Inventory	234	8	tested	23	23	23	0	0	0	0	0	0	0	69	
			goal	23	23	23	23	23	23	23	23	23	23	29%	
			passed	20	18	20	0	0	0	0	0	0	0	0	
			Retest	3	3	2	0	0	0	0	0	0	0	0	
			blocked	0	0	0	0	0	0	0	0	0	0	0	
WIP	96	0	tested	10	10	10	0	0	0	0	0	0	0	30	
			goal	10	10	10	10	10	10	10	10	10	10	31%	
			passed	8	8	8	0	0	0	0	0	0	0	0	
			Retest	0	0	0	0	0	0	0	0	0	0	0	
			blocked	0	0	0	0	0	0	0	0	0	0	0	
Planning	117	4	tested	12	12	12	0	0	0	0	0	0	0	36	
			goal	12	12	12	12	12	12	12	12	12	12	31%	
			passed	10	11	8	0	0	0	0	0	0	0	0	
			Retest	2	0	2	0	0	0	0	0	0	0	0	
			blocked	0	0	0	0	0	0	0	0	0	0	0	
ENG/BOM	153	5	tested	15	15	15	0	0	0	0	0	0	0	45	
			goal	15	15	15	15	15	15	15	15	15	15	29%	
			passed	15	10	12	0	0	0	0	0	0	0	0	
			Retest	0	4	1	0	0	0	0	0	0	0	0	
			blocked	0	0	0	0	0	0	0	0	0	0	0	
Costing	122	8	tested	12	12	12	0	0	0	0	0	0	0	36	
			goal	12	12	12	12	12	12	12	12	12	12	30%	
			passed	8	10	10	0	0	0	0	0	0	0	0	
			Retest	4	2	2	0	0	0	0	0	0	0	0	
			blocked	0	0	0	0	0	0	0	0	0	0	0	
<b>Totals</b>	<b>846</b>	<b>26</b>	tested	84	84	84	0	0	0	0	0	0	252		
			goal	85	85	85	85	85	85	85	85	85	846		
			passed	71	68	70	0	0	0	85	0	0	209		
			Retest	10	9	7	0	0	0	0	0	0			
			blocked	0	0	0	0	0	0	0	0	0			
			Failed	3	7	7	0	0	0	0	0	0	17		

SAMPLE



cum. tested	84	168	252	252	252	252	252	252	252	252	252	252	846
cum. goal	84.6	169	254	338	423	508	592	677	761	846			
% to goa	99%	99%	99%	74%	60%	50%	43%	37%	33%	30%			
Cum blocked	0	0	0	0	0	0	0	0	0	0	0	0	
Cum failed	3	10	17	17	17	17	17	17	17	17	17	17	
Cum passed	71	139	209	209	209	209	209	209	209	209	209	209	
Cum retest	10	19	26	26	26	26	26	26	26	26	26	26	

● Unit Tests will receive one of 4 testing statuses:

➤ **PASS**—Actual results match the expected results and no significant issues were identified

➤ **RETEST**—Actual results do not match the expected results for reasons that can be corrected and retested

➤ **FAIL**—Actual results do not match the expected results and the errors cannot be corrected during the DT

➤ **BLOCKED**—The test cannot be executed because of missing system components (e.g., interface or extension)

**13.7 Exhibit 7 Release Matrix**

<b>RELEASE</b>	<b>DESIGN TEST (DT)</b>	<b>GO-LIVE / DEPLOYMENT</b>
FMS Release 3.3 DCS & FFEL Multi-Org Functionality	DT I 25-JUN to 20-JUL	Production 01-AUG
FMS Release 3.4 Direct Loan Servicing	DT II 06-AUG to 24-AUG	Production 01-SEP
FMS Release 3.5 GAPS Feeder Systems – PELL, Campus Based, Direct Loan	DT III (“End2End” Integration) FSA FMS, GAPS, ED OCFO FMSS, VFA 10-SEP to 21-SEP	Production 17-OCT

## 14. APPENDIX

### 14.1 CONTINGENCY TESTING

The Contingency testing has been initiated due to the OCFO FMSS implementation “go-live” delay. The current FSA FMS implementation model represents the final solution proposed for the joint “go-live” date for FSA FMS, GAPS, OCFO FMSS, and all FSA Feeder systems releases. This model is based upon the new GAPS release and the FMSS Oracle implementation. In order for FSA CFO to meet it’s mandate to be JFMIP compliant, FSA CFO has designed a contingency solution for the FSA Financial Management System that will allow OFSA to process OFSA feeder system files in the “Old” GAPS format, maintain the existing feed to GAPS, return GAPS files to the feeder systems, and support the FSA GL close to IEFARS.

This contingency model allows for ‘As Is’ processing with the exception that all feeder system files are transmitted to FSA FMS and the GAPS output files are returned to FSA FMS. Under the contingency model, FSA FMS receives the feeder system file, makes a copy and then transmits the file directly to GAPS. In addition, FSA FMS accepts the GAPS generated output file and makes a copy prior to returning the GAPS outfile back to the feeder system. The model then allows for processing into the FSA FMS instance to capture all OFSA transaction accounting.

The contingency testing methodology will consist of controlled test scripts utilizing both historical archival Program files and GAPS archival output return files. All testing will require a level of validation. Data Sampling will be the method for the validation. The FSA FMS implementation team will be performing contingency testing using archived production files. Sampling percentages will vary based on the particular program being testing. Each program’s production behavior will determine the level of validation and sampling percentage. FSA FMS received input from the FSA IV&V organization to obtain input in the sampling percentages.

The FMS Phase III Contingency Testing scope includes testing of the following program interfaces:

- PELL
  - Process 5 days of historical files from the Feeder system from the month of September 2001.
  - Contingency Testing (the FMS team will set-up, control, and walk FSA CFO users through these tests results):
    - Budgetary Journal Entries (from ED CFO FMSS)
    - School data file load from Pell RFMS
    - Obligations from RFMS (OB-only)

- Obligate & Pay from RFMS (OB/PY)
- Deobligations from RFMS (-OB)
- Expenses from RFMS (EX and -EX)
- Error handling (at a batch level as well as at a transaction/ACCS level)
- Drawdowns (DD)
- Treasury Confirmations (TC; of both Payments (PY) and Drawdowns (DD))
- Returns (RE)
- Return Confirmations (FR)
- Refunds (RF)
- Refund Reversals (-RF)
- Upward and Downward Adjustments (-AD and AD)
- Unprocessed Deobligations (-UD)
- Processed Deobligations (PD)
- Disbursement Offsets (DF) - these are rare so they may not be tested
- Perkins Adjustments (PK) - these are rare so they may not be tested
- GAPS transactions that do not have an accounting impact in FMS (Obligation Confirmations (OB), DUNS number changes (DB), Error transactions (ER), etc)
  - Process ACA file to perform stress testing
  - Verify accounting.
- Direct Loan Origination

- Process 3 days of archived production files from the month of September 2001.
  - Contingency will process “unbooked” loan transactions and excess cash into the FMS GL, and payment requests transactions into FMS AP and create the export file in the GAPS format. Actual transfer of data to the GAPS system will take place in DT III.
  - Process of GAPS acknowledgements into FSA FMS instance
  - Verify Accounting
- Direct Loan Consolidation
    - Process 3 days of archived production files from the month of September 2001.
    - Design Test II will process “unbooked” loan transactions and excess cash into the FMS GL.
    - Contingency testing will incorporate testing of GAPS transactions. Obligations, Payment, De-obligations, Refunds, and Excess Cash transactions will be interfaced to FMS AP and GL and interfaced out to a GAPS formatted file. The actual transfer of data to the GAPS system will take place in DT III. Direct Loan Consolidation
    - Process of GAPS acknowledgements into FSA FMS instance
    - Verify Accounting
- Campus Based
    - Campus Based source files for contingency testing will consist of CBS files processed for month of September 2001
    - The scope for the Campus-Based program will include processing of the following
      - Budgetary Journal Entries
      - FSEOG, FWS, and Perkins Loan award obligations (positive/negative)
      - Teacher Cancellation Payments
      - FISAP Expenditures
      - Perkins Loan Balance Sheet Data
    - The archive files incorporate the complete “contingency” process flow of transactions. The GAPS-initiated transactions include:
      - Drawdowns (DD)
      - Treasury Confirmations (TC; of both Payments (PY) and Drawdowns (DD))
      - Returns (RE)
      - Return Confirmations (FR)

- Refunds (RF)
  - Refund Reversals (-RF)
  - Upward and Downward Adjustments (-AD and AD)
  - Unprocessed Deobligations (-UD)
  - Processed Deobligations (PD)
  - Disbursement Offsets (DF) - requires coordination with GAPS; these are rare so they may not be tested
  - Perkins Adjustments (PK) - also requires coordination with GAPS; also don't occur very often so they may not be tested
  - GAPS transactions that do not have an accounting impact in FMS (Obligation Confirmations (OB), DUNS number changes (DB), Error transactions (ER), etc)
- Create export file in GAPS format for Teacher Cancellation payments. The actual transfer of file to GAPS system for testing will take place in E2E testing phase.
- GAPS IPP
    - The scope of this testing will incorporate the creation of interface files of Pell, CB, DLO, and DLC data into GAPS format for review purposes.
    - Independent IPP testing scripts will be created for each program. These scripts will provide additional verification of the functionality of the IPP source code per program. The functionality and scope of features testing are synonymous with the each programs defined definition of scope.
  - LEAP/SLEAP
    - LEAP/SLEAP testing will include processing 3 days of transactions with accounting verification.
    - All LEAP/SLEAP transactions are GAPS generated.
    - Contingency testing will utilize extraction routine.

## **14.2 AGREED UPON PROCEDURES TESTING (AUP)**

Agreed Upon Procedures testing is a new FSA testing requirement as an audit mechanism for independent verification of All FSA accounting. FSA participation in AUP is due primarily to ensure that all accounting events that get transferred to OCFO FMSS are tested and verified. AUP will follow the DT methodology with respects to test scripts development and expected results documentation. In addition, FSA FMS will work jointly with Cotton and Company to establish guidelines for documentation and testing procedures. AUP will be performed using controlled scripts in a control Oracle Financials environment.

The Scope of AUP will be limited to testing the FFEL Lender / DCS and Direct Loan Servicing programs. All scripts per agreement should exercise all OCFO business events. Verification of compliance will be performed by OCFO.