DEPARTMENT OF EDUCATION-FSA
FMS INTEGRATION PARTNER
INTERFACE FUNCTIONAL DESIGN

COD TO FMS INTERFACE UPDATE

Author: Dave Stuart
Creation Date: October 30, 2002
Last Updated: January 13, 2003

TABLE OF CONTENTS

DOCUMENT CHANGE CONTROL **3**

APPROVAL **3**

SPECIFICATION **4**

GENERAL REQUIREMENTS **5**

INTERFACE SHORT DESCRIPTION: 5
DATA INTERFACE EXPECTATIONS: 5
BUSINESS EVENT THAT INITIATES INTERFACE: 5
INTERFACE CHARACTERISTICS: 5
SELECTION CRITERIA: 6
SCHEDULING REQUIREMENTS: 6
TRANSACTION PURGE CRITERIA: 6
CONTINGENCY PROCEDURES: 6
DEVELOPMENT COMPLEXITY: 6
ASSUMPTIONS: 6

DESIGN **7**

DATA FLOW DIAGRAM: 7
DATA FLOW DIAGRAM NARRATIVE: 7
DATA MAPPING: 8
DATABASE OBJECTS: 10

DEPENDENCIES **10**

DATA CLEANSING: 10
DEPENDENCIES ON OTHER RICE COMPONENTS: 10
DEPENDENCIES ON OTHER BUSINESS COMPONENTS: 10
DESIGN CONSTRAINTS: 10
PERFORMANCE CONSIDERATIONS: 10
IMPLEMENTATION CONSIDERATIONS: 10
OTHER DEPENDENCIES/SPECIAL CONCERNS: 10

ERROR HANDLING **11**

TESTING REQUIREMENTS **11**

ISSUES AND ADDITIONAL CONSIDERATIONS **11**

APPENDIX A **12**

Document Change Control

Date	Author	Version	Change Reference
10/30/02	John Kim	1.0	Document Creation Date
11/14/02	Dave Stuart	2.0	Revised Document Based on Comments
01/06/03	Dave Stuart	3.0	Revised Document Based on Comments
01/08/03	Mike Connors	4.0	Revised Document Based on Comments
01/13/03	John Kim	5.0	Revised Based on Final Review

Approval

Created By:	Dave Stuart	962-0685	**Creation Date:** 10/30/02
Team Lead Sign Off:	John Kim	962-0702	**Sign Off Date:** 01/13/03
Project Lead Sign Off:	Jen Alden	962-0682	**Sign Off Date:**
FSA Representative Sign Off:	Shirley Singleton	377-3491	**Sign Off Date:**

Specification

Interface Name:	COD to FMS Interface
Functional Design Document ID & Title:	COD_FMS_Inteface_Update_FD
Application:	Oracle G/L, Oracle A/P
Priority:	High
Estimated Production Date:	April 2003
Other Affected Applications:	N/A
Comments:	N/A

General Requirements

Interface Short Description:

The COD to FMS Interface program was originally implemented as part of FMS Phase IV efforts. The program was designed to facilitate the FMS interaction with Common Origination and Disbursement (COD) feeder system. This interface allows FMS to capture, control, and report on Direct Loan Origination and PELL related financial events.

The purpose of this document is to provide the updates to the interface program in accordance with the enhancement requirements. This document provides a detailed functional description of the updates to the interface program. As part of the TO 119 – Transaction ID effort, FMS has been tasked with the following enhancement requirements (as stated in section 1.0 of the Core FMS Requirement Matrix document):

- Ability to accept 18-character (alphanumeric) Transaction ID for all COD transactions.
- Ability to store 18-character transaction ID in an attribute field in FMS for each transaction from COD.
- FMS will not generate Transaction IDs for transactions sent by COD without Transactions IDs.
- FMS will send notifications to COD when previously failed transactions have been corrected in FMS.

In addition, the existing interface program will be tuned for performance improvements.

Data Interface Expectations:

The updated COD to FMS Interface program will capture (Requirement Number 1.1.1) and store (Requirement Number 1.1.2) the Transaction ID from COD in FMS. Subsequently, the Transaction ID will be sent to GAPS along with the transactions through the FMS to GAPS Interface program. The purpose of the Transaction ID is to enhance the reconciliation effort between COD, FMS, and GAPS. However, COD may send FMS transactions without Transaction IDs. FMS will not generate Transaction IDs (Requirement Number 1.1.3) for these transactions.

The purpose of the performance tuning is to reduce the processing time. Performance tuning includes reviewing and updating the existing code in order to make it more efficient. Appendix A illustrates the design for updating the code.

For previously failed transactions that are corrected, a status code of 'C' will be passed from FMS to COD. COD will need to be able to accept a status code of 'C'. (Requirement Number 1.1.4)

Business Event That Initiates Interface:

The COD to FMS Interface program is scheduled to run daily.

Interface Characteristics:

Interface Direction:	Inbound from COD to FMS
Interface Type:	Batch
Interface Category:	Update to the current COD to FMS Interface
Instances:	N/A

Frequency:	Daily
Volume:	The volume of transactions may vary.
From System: COD	**To System:** FMS

Selection Criteria:

Same as the existing COD to FMS Interface program.

Scheduling Requirements:

Same as the existing COD to FMS Interface program.

Transaction Purge Criteria:

Same as the existing COD to FMS Interface program.

Contingency Procedures:

Same as the existing COD to FMS Interface program.

Development Complexity:

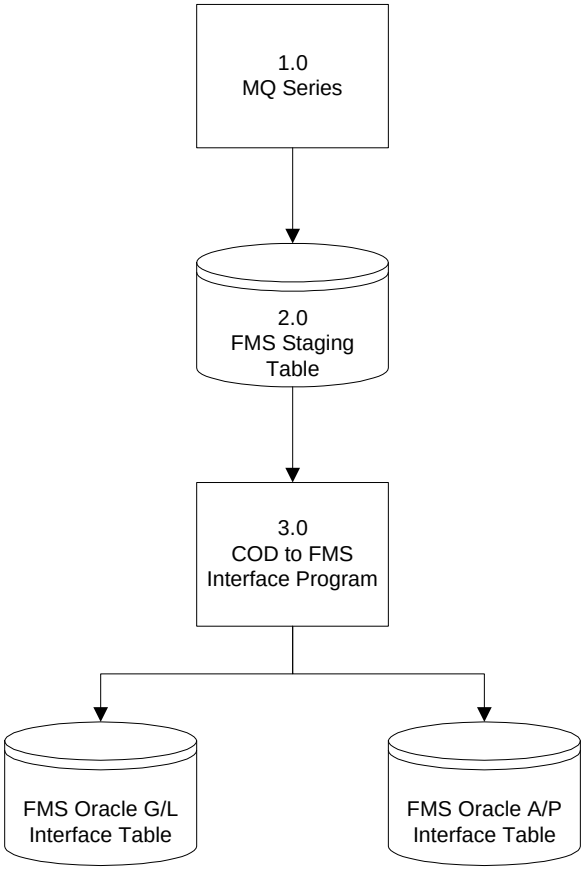
- Transaction ID – Low
- Performance Tuning - High
- Notification for Corrected Transactions – Medium

Assumptions:

- As discussed in meetings with COD representatives, MQ Series will populate the Transaction ID from COD accordingly in the FMS Staging Table.
- FMS will update the script to populate the FMS Staging Table.
- ATTRIBUTE13 fields of the GL_JE_LINES table and AP_INVOICES table will be used for storing Transaction IDs in FMS.
- The FMS Item Number is currently captured and stored in ATTRIBUTE7 of the GL_JE_LINES table. Therefore, there is no additional requirement to store this value for COD transactions.
- The manual corrections to COD initiated transactions are performed on the staging table not Oracle Base tables.
- COD will process the notifications for corrected records.

Design

Data Flow Diagram:



Data Flow Diagram Narrative:

1.0 MQ Series
MQ Series will serve as an interface between COD and FMS. MQ Series will load COD transaction data in the FMS Staging table and FMS will start processing from there.

2.0 FMS Staging Table
MQ Series loads the FMS Staging table with FMS destined transactions that originated from COD. The COD to FMS Interface Program scans this table for new transactions when initiated.

3.0 COD to FMS Interface Program
The COD to FMS Interface program will sweep the data from the FMS Staging Table and load both the G/L Interface Table and AP Interface Table. The Transaction ID will be stored in ATTRIBUTE13 of both tables. If the Transaction ID is not passed for a particular transaction, the ATTRIBUTE13 field will be left null for that record.

Data Mapping:

The table below describes the record layout for financial/non-financial transactions between COD and FMS. COD will send transactions to MQ Series in this record format and periodically a concurrent program within FMS will pull these transactions into Oracle inbound tables.

COD Transaction Record Layout:

Start	End	Length	Data Type	Name	Description	Values/Rules
1	10	10	A	Transaction ID	Unique identifier of transaction	COD will populate the first field of all COD created txns as "C". FSA FMS will populate the first field of all FSA FMS created txns as "F".
11	18	8	N	Transaction Date	Date the transaction was created on the source system	MMDDYYYY
19	26	8	N	Transaction Time	Time the transaction was created on the source system	HHMMSSNN
27	28	2	A	Transaction Type	The business event	AD, DA, DB, DD, DF, DP, ER, EX, FA, FE, FL, GR, OP, PL, PY, RA, RB, RC, RE, RF, UB, UD, XE
29	31	3	A	Program Type	The FSA program area	DLO, PEL
32	39	8	N	Grantee Institution	Common school identifier for a grantee	Common School Identifier
40	47	8	N	Payee Institution	Common school identifier for payee	Common School Identifier
48	57	10	N	Grantee DUNS #	DUNS # belonging to grantee	Grantee DUNS # or Zeroes
58	60	3	A	CFDA	Catalog of Federal Domestic Assistance; accounting code	CFDA code or Spaces Note: COD will send the CFDA code for all transactions.
61	64	4	A	CFDA Subpart	Subdivision of the program field	P, J, Q, R, K or Spaces
65	68	4	N	Award Year	School year, program year	CCYY
69	72	4	N	Award Sequence Number	Fifth part of GAPS award number	Award Sequence #
73	75	3	A	Limitation	Part of ED ACCS	Allotment Limitation code or Spaces
76	78	3	A	Activity	Part of ED ACCS	Allotment Activity code or Spaces

						Note: COD will send spaces and FSA FMS will derive the Activity code from the transaction type.
79	80	2	A	Loan/Grant Type	Defines the loan and grant type.	D1 (DL Subsidized) D2 (DL Unsubsidized) D4 (DL PLUS) PE (Pell)
81	84	4	N	Funding Fiscal Year	Initial year of the appropriation	CCYY or Zeroes
85	88	4	N	Budget Fiscal Year	Year of apportionment	CCYY or Zeroes
89	92	4	N	Cohort Year	Year in which the loan was first disbursed	CCYY or Zeroes
93	106	14	N	Amount	Dollar value	99999999999999 or Zeroes
107	107	1	A	Amount Sign	Denotes increase or decrease	+/- or Spaces
108	115	8	N	Effective Date	Used for future processing functionality only; denotes the date the transaction is to be processed in FSA FMS	MMDDYYYY or Zeroes
116	116	1	N	Participation Level	New participation level for Direct Loan	Participation Level or Zeroes
117	130	14	N	Payment Request Limit Amount	Limit amount on a payment	Zeroes
131	136	6	A	Schedule Number	SF215 schedule number for refunds and returns; SF5515 number when a bank rejects return of funds	SF215/SF5515 or Spaces
137	149	13	N	Pay Control Number	Check number for drawdowns performed in GAPS	Pay Control # or Zeroes
150	157	8	N	Confirmation Date	Treasury Confirmation or deposit date of refunds and returns	MMDDYYYY or Zeroes
158	159	2	N	Error Code	Code detailing a GAPS error	Error Code or Zeroes
160	161	2	A	Error Transaction Type	Type of GAPS error	Error Code or Spaces
161	178	18	A	Transaction ID	Unique identifier	<Program Id><Date> <Sequence #> • Program Id = ‘PL’ or ’LO’ • Date = YYYYMMDD • Sequence # = 8-digit sequence number

Database Objects:

Database Object Name	Object Type	Oracle Application	Comments
N/A			

Dependencies

Data Cleansing:

Task	Responsibility	Dependency
N/A		

Dependencies on Other RICE Components:

N/A

Dependencies on Other Business Components:

- FMS to GAPS Interface
- GAPS to FMS Interface
- FMS to COD Interface

Design Constraints:

N/A

Performance Considerations:

See Appendix A on Performance Tuning .

Implementation Considerations:

N/A

Other Dependencies/Special Concerns:

Need to make sure that modifying the existing code to improve processing time does not create problems with capturing all data from COD.

Error Handling

Potential Errors	Possible Reasons	Solution Strategy
Same as existing COD to FMS Interface.		

Testing Requirements

Test Condition	Expected Results
Refer to Unit Test Scripts in F: TO 119 - Transaction ID\ Core FMS\ Interface Design Updates\ Unit Testing\TEST SCRIPTS	

Issues and Additional Considerations

The following Issues were defined during Functional Design:

Issue	Raised By / Date Needed	Resolution/Answer	Resolved By / Date Completed

Appendix A

SFA FMS COD to GL/AP/AR Concurrent Program, Performance Improvements

This document has been created in response to the increasing execution times for the COD to FMS interface. Based upon a brief analysis of the underlying PL/SQL procedures a number of suggestions can be made regarding performance improvements.

The COD to FMS processes two types of data: invoices transferred to Oracle Payables and journal entries transferred to Oracle General Ledger. All performance improvement suggestions that are made require changes to the code for the journal entries transfer. This is due to the much larger data volumes. There are regularly thousands of journal entry records to process, while rarely more than one hundred payable invoice records. Hours of savings can be gained with improvements in processing journal entries, but only minutes from invoice processing improvements. Therefore, the most significant performance savings will be achieved by tuning the GL processing.

A brief description of the GL processing portion of this interface is provided for understanding of the improvement suggestions. The codx_trans_gl table contains records submitted from COD that are required to be loaded into the FMS Oracle General Ledger. The COD to FMS interface loops through all of the unprocessed records in this table. For each record, a number of stand-alone queries are executed to retrieve additional information, data validations occur, and data is prepared for import to Oracle GL. For each record, one or more records are inserted into the Oracle GL open interface table, gl_interface. Once the program completes looping through unprocessed records, it submits three standard Oracle requests, Journal Import, to transfer the gl_interface data to the Oracle GL base tables.

Many of the performance improvement suggestions are due to the stand-alone queries that are executed for each record. The overhead associated with executing these queries occurs for each record. As volume increases, processing time will increase proportionately. It is more efficient to retrieve as much data as possible in the main query that is used to loop through each record.

- A cursor, codx_gl_cur, is used to loop through all records in the codx_trans_gl table. The cursor definition includes an order by clause, however I do not see that the order of the records has an impact on program processing. Removal of this clause would improve the performance of the query.
- A stand-alone query is executed for each looped record to retrieve the organization_id from the hr_operating_units table. It would be more efficient to include this lookup in the main cursor, codx_gl_cur.
- A common procedure is called to retrieve the Grantee Duns Number, Payee Duns Number, and Institution. Looking at this common procedure, fms_iface_inst_pkg.get_inst_info, I noticed some processing that may not be necessary. For instance, a query is executed to return vendor related information that is not used by the COD to FMS interface. We should consider eliminating the call to the common procedure, and including only the necessary queries/logic in the COD to FMS interface.
- A stand-alone query is executed for each looped record to retrieve an FMS item identifier from a custom lookup table, codx_item_no_map. It would be more efficient to include this lookup in the main cursor, codx_gl_cur.
- A stand-alone query is executed for each looped record to retrieve the necessary accounting from the ffelga_account_mappings table. It would be more efficient to include this lookup in the main cursor, codx_gl_cur.
- A stand-alone query is executed for each looped record that attempts to select a flag to indicate whether the journal entry must be transmitted to GAPS. This query selects from the codx_trans_ap table attempting to compare a number of fields in the current records to values in this table. It is difficult to tell the business reasons for this query, but I think it's unlikely that it is accomplishing its objective. This query could potentially be removed, but at the very least needs to be modified.
- As records are transferred to the gl_interface table they are placed in three different batches. After transferring all records to the gl_interface table, the program initiates the Journal Import program once for each batch. Currently it submits the first Journal Import, waits for completion, submits the second Journal Import, waits for completion, submits

the third Journal Import and waits for completion. The Journal Import regularly has 20 – 30 minute run times so this process could take well over an hour. Rather than submitting the Journal Imports back to back, they should be submitted concurrently. This would cut the wait time by two-thirds.

These recommendations are made based upon a brief analysis of the COD to FMS code. It would be beneficial to take further steps to find additional inefficiencies in the COD to FMS interface so that the processing time may be reduced to an optimal level. These steps include:

- Enabling SQL Trace for an execution of COD to FMS in Production to better pinpoint inefficient queries
- Begin implementing performance changes and debugging in a production-like environment. This will be possible when the refreshed DEVCOM is available.
- Execute individual queries against production data to test efficiency