DEPARTMENT OF EDUCATION-
FEDERAL STUDENT AID
FINANCIAL MANAGEMENT SYSTEM

FINANCIAL MANAGEMENT SYSTEM (FMS)
FUNCTIONAL DESIGN DOCUMENT
FOR
LOAN PROCESSING & ISSUANCE FEE (LPIF)
CALCULATION PROGRAM ENHANCEMENT

October 23, 2003

Version 1.1

Document Control

Version Number	Description	Release Date	Author
1.0	Initial Issue	10/22/03	Integration Partner, FMS LPIF Interface Change Team
1.1	Incorporated Shirley Pratt's suggestions regarding clarification of error processing, clarification of logic for the calculations, renaming of the interface, and document how long the flat files would be stored (3 years) for audit purposes.	10/23/03	Integration Partner, FMS LPIF Interface Change Team

Table of Contents

Loan Processing and Issuance Fee (LPIF) Interface Enhancement ***1***

General Requirements **1**
- Business Objective 1
- Scope 2
- Business Process Diagram 3
- Component Description 4
- Component Risks 5
- Assumptions 5
- Specifications 6

Functional Design **7**
- Data file Structure 7
- Design Constraints 8
- Performance Considerations 8
- Implementation Considerations 8

Testing Requirements **8**

Appendix A - Technical Design **9**

FSA FFEL GA NSLDS LPIF Fee Load Interface - Concurrent Program Logic 9
- Calling Arguments 9
- Log Output 9
- Table and View Usage 10
- Program Logic (pseudo code) 11
- SQL Statements 11
- Validation Logic 12
- Incompatibility 13

Other Considerations 13
- Restart Strategy 13
- Crash Recovery 13

Appendix B - Issues and Additional Considerations **14**

Loan Processing and Issuance Fee (LPIF) Interface Enhancement

General Requirements

Business Objective

Currently, NSLDS transmits a file that is used as the basis for LPIF fee calculation. NSLDS transmits the LPIF file once every quarter and contains data for that quarter only. The file has one record for each GA and contains loan amounts. The file header record contains Fiscal Year (FY) information. This FY information is used to determine the rate for the entire file, which is applied to each loan amount to calculate the LPIF Fee for the quarter.

Rate calculation is supported by establishing LPIF translation currency by loading the gross amount as LPIF currency and translating it into US dollars by multiplying it with the rate for the fee.

However, in the current process, only one rate can be applied to loan amounts provided at the detail or line level information. After October 1, 2003, a Guaranty Agency (GA) will have loans that are subjected to rates of either 6.5 or 4.0 basis points. For loans with a guaranty date after October 1, 1999 but before October 1, 2003, disbursements will be paid at 6.5 basis points. For loans with a guaranty date on or after October 1, 2003, disbursements will be paid at 4.0 basis points. The FY information in the file header can no longer be used to determine basis points. In order for FMS to determine which basis point rate should be applied to the disbursement amount, a specific rate indicator will be sent with each GA detail record.

It has been agreed that from October 1, 2003 onwards, NSLDS will send basis points within the detail records in the flat file. This would eliminate the need of using standard Oracle Applications functionality of currency conversion for calculating disbursement amounts. The rate sent in the file would not be validated on the FMS side, since no guaranty date is provided in the file at the record level.

The LPIF calculation program would validate the file both at file level and at record level.

1. The following are examples of file level validations. If the file fails any of the following validations, then the whole file will be rejected. Exceptions will be reported to NSLDS via an email message:
   - Only one header record must be present in the file.
   - Only one trailer record must be present in the file.
   - Number of detail records in the file must match with the record count mentioned in the trailer record.
2. The following is an example of record level validations. If any of the records fail the following validation, then the whole file will be rejected. Exceptions will be reported to NSLDS via an email message:
   - Basis Point must be non-zero.

If the whole file has to be rejected, then an email would be generated and sent to NSLDS with an explanation of the errors encountered. NSLDS would need to review, correct, and restart the process by sending an updated/corrected LPIF file to FMS. Once the new file was received, the file would be re-validated.

Once validation has completed, information from the file would then be loaded into a temporary table and processed to arrive at the fee amount. Disbursement amount would be calculated based on the basis points provided by NSLDS in the file. The following logic will be used to calculate the disbursement amount:

- Loan Amount $10,000 and basis points 6.5 will be mentioned in the flat file as follows:
  Loan Amount = 10000
  Basis Points = 65000
- 65000 will be converted to basis points as following:
  6.5 Basis Points = 65000 / 1000
- Disbursement Amount would be 10000 * (6.5) / 100 = 650
- An invoice will be created for the amount of $650

The logic/algorithm for the calculation is located within the LPIF interface module. Once the calculation is complete, the data would then be loaded into the Oracle Payables Open interface tables and eventually imported into Oracle Account Payables as invoices.

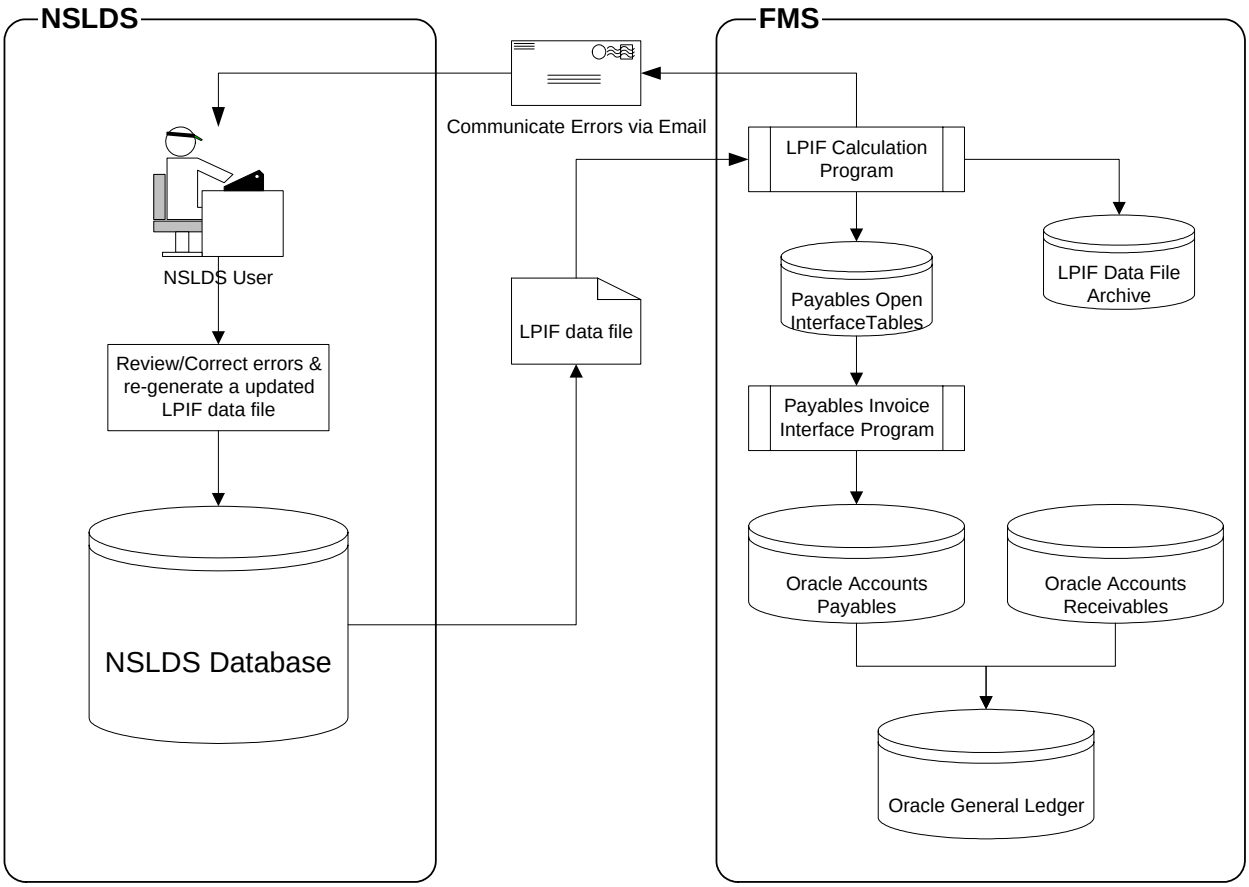
Once the FMS LPIF interface has successfully processed the NSLDS LPIF input file, the file will be retained (archived) for a minimum of 3 years for audit purposes.

Scope

The interface will be modified to achieve following:

1. To process the file received from NSLDS via the eAI bus (IBM MQseries).
2. To calculate disbursements based on multiple rates. The interface should be able to accommodate future rate changes without any further code changes.
3. The rate sent in the file from NSLDS would not be validated on the FMS side, since no guaranty date is provided in the file at the record level.
4. The interface should be able to communicate the following to NSLDS:
   A. Missing LPIF file, via email.
   B. Errors at file level or at record level, via email.

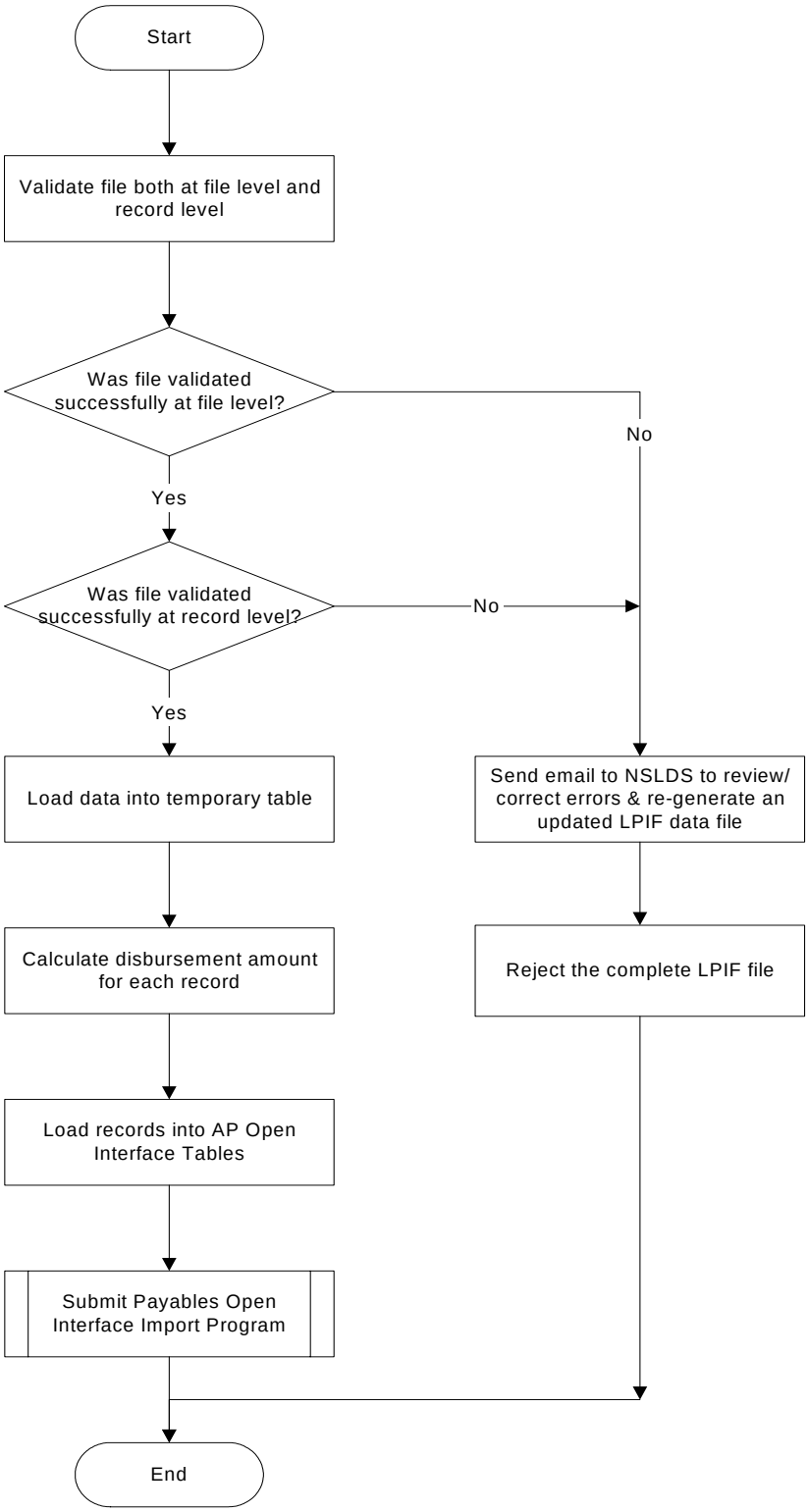
Business Process Diagram



Component Description

NSLDS Fee Load – Concurrent Program

The NSLDS Fee Load concurrent program is needed to create invoices for LPIF fees in the Accounts Payable Application. Following is the flowchart for the concurrent program.



When to Run the Program
The NSLDS Fee Load concurrent program will be set up to run automatically based on when LPIF files are provided by NSLDS.

Launch Parameters
Launch parameters for the NSLDS Fee Load concurrent program include the standard error code and error buffer parameters required for any concurrent manager program.

Business Rules Implemented
Invoices for LPIF fees will be created in Oracle AP based on the calculation rules stated above.

Log Output
The log output for the NSLDS Fee Load concurrent program consists of messages after each step of the import process. If the process is successful, proper AP invoices will be created. However, if there are any errors within the process, error messages will be provided.

Restart Procedures
Restart procedure is needed for following situations:

1. When concurrent program completes with errors:
    - When a file fails any file level validations or record level validations, which need to be reviewed and corrected by NSLDS, the whole fill would be rejected. In that situation get the corrected file from NSLDS and then resubmit the concurrent program.
2. When concurrent program gets killed because of concurrent manager issues or because of database crash:
    - If data gets trapped in temp table or stage table, then manually cleanup the tables and restart the NSLDS Fee Load – Concurrent Program.
    - If partial data is transferred to AP and rest is in AP open interface tables then resubmit the AP open interface program.

Component Risks

- The LPIF interface calculates disbursements based on the basis points and amount provided in NSLDS file. The rate sent in the file from NSLDS would not be validated on the FMS side, since no guaranty date is provided in the file at the record level

Assumptions

- NSLDS will continue to provide all data in one file to FMS per quarter.
- NSLDS will provide basis points on the line level of the data that can be used by FMS to calculate the disbursements.
- NSLDS will not make adjustments to the guaranty date and disbursement amount after the data has been processed in FMS Accounts Payable.
- No other changes to the current FMS LPIF Interface process are anticipated or within the scope of this effort.
- Testing will be coordinated and jointly planned between NSLDS and FMS.

Specifications

Program Name:	SFA FFEL GA NSLDS LPIF Fee Load Interface
Application:	Oracle Payables
Risk Factor:	LPIF interface calculates disbursements based on the file provided by NSLDS.
Value:	N/A
Benchmark:	N/A
Score:	N/A
Data Source:	Data file provided by NSLDS.
Data Requirements:	Data file provided by NSLDS.
Attributes	N/A
Prompts:	N/A
Metrics	See Requirements Traceability Matrix
Filters:	None
Output:	N/A

Functional Design

Data file Structure

NSLDS will provide flat file in following agreed upon format:

Summary File for FMS Header Record Layout Length=80

Position From	Position To	Attribute	Description	Field Format	Length
1	9	Record Sequence Number	Positional sort field for sorting the Header record to the top of the file. Set the value to all Zeroes.	Char	9
10	17	Data Provider Indicator	"NSL2000"	Char	6
18	25	Create Date	Date on which the file is created: CCYYMMDD	Date	8
26	33	Create Time	Time when the file is created: HH:MM:SS format	Char	8
34	34	Submission Type Indicator	O - Original for the Period Ending R - Replace for the Period Ending C - Corrected for the Period Ending. When not applicable set to Blank	Char	1
35	63	File Description	"ED FMS NSLDS INTERFACE HEADER"	Char	29
64	70	Filler	Future Needs	Char	7
71	78	Quarter/Year Ending Date	CCYYMMDD format; Example: 20031231 (Qtr 1); 20030331 (Qtr 2); 20030630 (Qtr 3); 20030930 (Qtr 4 and FY End)	Date	8
79	80	Filler	Future Needs	Char	2

Summary File for FMS Detail Record Layout Length=80

Position From	Position To	Attribute	Description	Field Format	Length
1	9	Record Sequence Number	Positional sort field: start with 000000001 and increment by 1 for each record.	Char	9
10	12	GA Code	Code for Guaranty Agency	Char	3
13	16	NSLDS assigned Code	NSLDS assigned unique Identifier: 8955 AMF NET GUARANTY AMOUNT 8952 LPIF DISBURSEMENT AMOUNT	Char	4
17	28	Future needs	Blank	Char	12
29	43	Amount	Amount in display format (Positive or negative)	Num	15
44	48	Fee Rate	Basis point for fee calculation. (LPIF only) IE 65000 for 6.5 basis points 40000 for 4.0 basis points	Num	5
49	80	Future needs	Blank	Char	32

Summary File for FMS Trailer Record Layout Length=80

Position From	Position To	Attribute	Description	Field Format	Length
1	9	Record Sequence Number	Positional sort field for sorting the Trailer Record to the bottom of file. Set the value to all nines.	Char	9
10	29	Filler	Blank	Char	20
30	58	File Description	"ED FMS NSLDS INTERFACE TRAILER"	Char	29
59	66	Filler	Blank	Char	8
67	75	Total Number of Detail Records	This number will reflect the number of detail records contained in the current file. Should match the sequence number of the last Detail Record.	Char	9
76	80	Future needs	Blank	Char	5

Design Constraints

The program will calculate the disbursement amount based on basis points provided at the detail record level in the flat file.

Performance Considerations

None

Implementation Considerations

None

Testing Requirements

Forthcoming in the 146.1.2 – Test Scripts deliverable, due on 11/14/03.

Appendix A - Technical Design

FSA FFEL GA NSLDS LPIF Fee Load Interface - Concurrent Program Logic

The FSA FFEL GA NSLDS LPIF Fee Load Interface will load LPIF fee amounts associated to each GA from a data file into the AP open interface tables in order to create invoices.

Once a data file is provided, the LPIF Fee Load concurrent program will be triggered via a Unix shell script that will provide the file location and the name of the file submitted.

Calling Arguments

The FSA FFEL GA NSLDS LPIF Fee Load program can be called from the Standard Report Submission form. However, the program will be triggered from a Unix shell script in order to automate file processing. Below is the list of arguments that are passed:

Argument	Prompt	Value Set	Default Value
P_FILE_LOC	File Location Directory	50 Chars	None
P_FILENAME	File Name	15 Chars	None

Log Output

```
.........1.........2.........3.........4.........5.........6.........7.........8
_____________________________________________________________________________

+-----------------------------------------------------------------------------+
SFA FFEL GA Extensions: Version : UNKNOWN - Development

Copyright (c) Oracle Corporation 1979, 1991. All rights reserved.

SFANSLDSAPLD module: FMS FFEL GA NSLDS LPIF Fee Load Interface
+-----------------------------------------------------------------------------+

Current system time is 24-JUL-2000 17:27:12

+-----------------------------------------------------------------------------+


+-----------------------------------------------------------------------------+
Start of log messages from Plsql program
+-----------------------------------------------------------------------------+
. Initializing temp table...
. Commit transaction
. Initialization Complete...
. Extracting data from file to temp table...
File Processed and would be closed
. Extraction Complete...
. Validating data...
. Validation Complete...
. Loading into NSLDS table...
. Commit transaction
. Loading Complete...
```

```
. Loading AP Interface tables...
. Submitting request for APXIIMPT
. Argument1 = NSLDS
. Argument2 = 516
. Argument3 = FFELNSLDSAMFB39
. Argument7 = Y
. Request id = 8870
. Commit transaction
. Commit transaction
. AP Interface Load Complete...
Import Complete
+---------------------------------------------------------------------------+
End of log messages from PlSql program
+---------------------------------------------------------------------------+


+---------------------------------------------------------------------------+


Executing request completion options...


Printing output file.
               Request ID : 8869
         Number of copies : 0
                  Printer : TEST



Finished executing request completion options.



+---------------------------------------------------------------------------+
Concurrent request completed successfully
Current system time is 24-JUL-2000 17:27:13

+---------------------------------------------------------------------------+
```

Table and View Usage

Table Name	Select	Insert	Update	Delete
AP_INVOICE_LINES_INTERFACE		X		
AP_INVOICES_INTERFACE		X		
FFELGA_ACCOUNT_SEGMENTS	X			
FFELGA_NSLDS_FEES	X	X	X	
FFELGA_NSLDS_TEMP	X	X		X
FND_APPLICATION	X			
FND_CONCURRENT_PROGRAMS	X			
FND_CONCURRENT_PROGRAMS_VL	X			
FND_CONCURRENT_REQUESTS	X			
FND_CURRENCIES	X			
PO_VENDOR_SITES_ALL	X			

Program Logic (pseudo code)

```
BEGIN
      IF Data File is missing THEN
         Send email to NSLDS;
         Terminate the program with status as error;
      END IF;

      Open Data File;
      Extract information from data file into temp table;
      Validate Data;
            Verify Header and Trailer count;
            Verify Record Total as specified in Header;

            IF New Submission, verify that no record currently exists;

      IF Data is Valid THEN
            Select rows             (see SQL statement 1)
            FOR Each row returned
                  Insert into FFELGA_NSLDS_FEES;
            END FOR;
      ELSE
         Reject File;
         Send email to NSLDS;
      END IF;

      Assign AP Group;

      Select rows;                  (see SQL statement 2)
      FOR Each row returned
            Assign Interface Invoice ID;
            Assign Invoice Number;
            Calculate Fee based on amount provided by NSLDS;
            Get Transaction Code (dependent on fee type);
            Get Account Segments;
            Insert into AP_INVOICE_LINES_INTERFACE;
            Insert into AP_INVOICES_INTERFACE;
      END FOR;

      Delete data from Temporary Table (FFELGA_NSLDS_TEMP);

END;
```

SQL Statements

1 - Data Selection

```
SELECT SUBSTR (FILE_DATA, 10, 3) GA_CODE,
       SUBSTR (FILE_DATA, 13, 4) NSLDS_CODE
       TO_NUMBER(LTRIM(SUBSTR(file_data,29,15),'0')),
       FEE_AMT
FROM   FFELGA_NSLDS_TEMP
WHERE  type = 'D';
```

2 - Data Selection

```
SELECT PERIOD_END_DATE,
       GA_CODE,
       FEE_TYPE,
       FEE_AMOUNT
FROM   FFELGA_NSLDS_FEES
WHERE  STATUS = v_stat_submit ('Submitted')
ORDER BY PERIOD_END_DATE,
         GA_CODE,
         FEE_TYPE
FOR UPDATE OF STATUS;
```

Validation Logic

Error Conditions:

The following errors are being handled within the FFELGA_NSLDS_TO_AP_PKG package:

PROCEDURE / FUNCTION	EXCEPTION / ERROR CONDITION	PURPOSE
INSERT_INTO_FFELGA_NSLDS_TEMP	NO DATA FOUND	This error is raised if no more data is found within the data file. A message is written to the log file.
	VALUE ERROR	This exception is raised if an attempt is made to insert inconsistent values into the temp table (data type inconsistencies). A message is written to the log file.
	OTHERS	This is a catchall exception. In case of any other errors, the error message is written to the log file.
VALIDATE_FFELGA_NSLDS_TEMP	NO DATA FOUND (header check)	This exception is raised if there is no header record found within the temp table. A message is written to the log file and validation of the data file fails.
	OTHERS	This is a catchall exception. In case of any other errors, the error message is written to the log file and validation of the data file fails.
	HEADER RECORD TOTAL VALUE <> DETAIL RECORD COUNT	This error is raised if the record total specified within the header is not equal to the number of total detail records. A message is written to the log file and validation of the data file fails.
	SUM OF DETAIL RECORD AMOUNT TOTALS <> TRAILER TOTAL AMOUNTS	This error is raised if the sum of the detail record amounts (principal, interest, other) does not equal the total amounts stated within the trailer. A message is written to the log file and validation of the data file fails.
GET_TRANSACTION_CODE	NO DATA FOUND	This error is raised if there is no transaction code found associated to the fee type (passed in as a parameter) within the Account Mapping table. A message is written to the log file and processing is terminated.
INSERT_INTO_NSLDS_TABLE	OTHERS	This is a catchall exception. In case of any other errors, the error message is written to the log file.
	SUBMISSION CODE = 'Original' AND PREVIOUS RECORD EXISTS	This error is raised if a file is received with a submission code of 'O' (Original) and there is a previous record for the same GA/fee type/period ending combination within the system. A message is written to the log file and processing is terminated.
	SUBMISSION CODE = "RESUBMIT' or 'CORRECTION' and NO PREVIOUS RECORD EXISTS	This error is raised if a file is received with a submission code of 'R' (Resubmit) or 'C' (Correction) and there is no previous record for the same GA/fee type/period ending combination within the system. A message is written to the log file.
INSERT_INTO_FFELGA_NSLDS_FEES	OTHERS	This is a catchall exception. In case of any other errors, the error message is written to the log file.
RETRIEVE_CURRENCY_DETAILS	OTHERS	This is a catchall exception. In case of any other errors, the

		error message is written to the log file.
INSERT_INTO_AP_FROM_NSLDS	NO DATA FOUND	This error is raised if the GA Code is not found within the Applications as a Vendor. A Message is written to the log file.
	OTHERS	This is a catchall exception. In case of any other errors, the error message is written to the log file.
FFELGA_NSLDS_PROCESS	VALIDATION CHECK	This error is raised if validation of the data file fails (see above errors). If validation fails, then messages are written to the log file and processing is terminated.
	DONE PROCESSING	This error is raised from other error handling an exception throughout the program in case the data file load process needs to be terminated. Messages are written to the log file and records from the temporary table are deleted

Incompatibility

The NSLDS Fee Load program has been set up to be incompatible with itself. By doing so, it is guaranteed that two instances of this concurrent program will never run simultaneously thereby avoiding possible problems with multiple instances of the program working on the same set of data.

Other Considerations

Restart Strategy
The NSLDS Fee Load program not only is a concurrent request itself, it also submits an additional standard concurrent request provided by Oracle Applications. If the error occurs within the NSLDS Fee Load program, then the user needs to make corrections and re-submit the request. However, if the error occurs in the Payables Open Interface Invoice Import program, the user needs to analyze the error(s), make corrections, and then re-submit the request for Payables Open Interface Invoice Import only.

Crash Recovery
The NSLDS Fee Load program has been marked to restart itself incase of a database crash. If a database crash were to occur while in the middle of execution, the NSLDS Fee Load program will re-start itself.

Appendix B - Issues and Additional Considerations

The following Issues were identified/resolved during Functional Design:

Issue	Raised By / Date Needed	Resolution/Answer	Resolved By / Date Completed
Current version of LPIF program on FMS is using Oracle's seed currency conversion feature. Is it necessary to continue using? Does it have in reporting impact related to LPIF data in FMS?	01-Oct-2003	Oracle's seed currency conversion feature need not be used if NSLDS is providing basis points information in flat file at detail record level.	Stephen Malleck 03-Oct-2003
The proposed change in file layout would impact both LPIF and AMF interfaces. Scope of TO 146 is limited to LPIF only.	07-Oct-2003	Verified that record layout change will not affect the AMF interface, and that only the Fee Rate Attribute of the NSLDS feed (columns 44-48) will impact the LPIF interface	Mitesh Patel 10-Oct-2003
The proposed basis points format can accommodate max 9.9999 basis points. What if basis point is 10 or above?	07-Oct-2003	This meets the business need.	Pam Eliadis 08-Oct-2003
NSLDS proposed to remove following data elements from the header record of the file layout: • Submission Type Indicator • Quarter / Year Ending Date NSLDS proposed to remove following data elements from the detail record of the file layout: • Line Item Identifier • Loan Type • Interest Amount NSLDS proposed to remove following data elements from the trailer record of the file layout: • File Description • Total Amount	03-Oct-2003	Researched the AMF and LPIF Interface modules, and determined that the FMS LPIF Interface requires the following fields in the header record of the file layout: • Submission Type Indicator • Quarter / Year Ending Date NSLDS has agreed to keep sending these data elements.	Pam Eliadis 08-Oct-2003