



F E D E R A L
S T U D E N T A I D
We Help Put America Through School

FSA Modernization Partner

NSLDS II Reengineering
Reports Detailed Design:
FPPS Cohort Default Rate History Report
R-DRC-031

Version 1.1

November 26, 2002

Table of Contents

1	FPPS COHORT DEFAULT RATE HISTORY REPORT.....	3
1.1	OVERVIEW	3
1.2	DRC EXCEPTION REPORT PROCESS OUTLINE	3
2	REPORT PARAMETER PROCESS.....	4
2.1	PARAMETER DEFINITIONS	4
2.2	PARAMETER SCREEN.....	5
2.3	PARAMETER TABLE STORED PROCEDURE.....	5
3	INFORMATICA REPORT TABLE GENERATION.....	6
3.1	LRDR FILE RETRIEVAL.....	6
3.2	LRDR DATA EXTRACTION	7
3.3	REPORT TABLE LAYOUT.....	9
3.4	USAGE CODE DETERMINATION	10
3.4.1	<i>Usage Code Requirements.....</i>	<i>10</i>
3.4.2	<i>Logic Code to Determine Usage 1 Codes.....</i>	<i>11</i>
4	FILE GENERATION.....	16
4.1	INFORMATICA UNFORMATTED FILE GENERATION	16
4.2	JAVA FORMATTED FILE GENERATION.....	17
5	EXCEPTION REPORT DOWNLOAD PROCESS	18

Document Control

Version Number	Description	Release Date	Author
1.0	Initial Issue	11/08/2002	Justin M. Miller
1.1	Updated Format	11/13/2002	Jason Patton

1 FPPS Cohort Default Rate History Report

Description	The FPPS Cohort Default Rate History Report provides a list of the cohort default rates for guaranty agencies or lenders and the loans that made up the requested cohort default rate.
User Group	ED Users
Data Source	LRDR File
Output Media	Downloadable File
Frequency	On Request
Requirements Traceability	2.024
Output Media	Formatted flat file or Unformatted flat file (fixed character width)
Comments	This report corresponds to the DRC031 – FPPS Cohort Default Rate History Report. This report will be generated in a formatted or unformatted flat file.

1.1 Overview

This report is a NSLDS II exception report, which has the following constraints that require the user to download the report instead of viewing it on the website.

- Result set is on average greater than 100,000 rows of data (affecting performance).
- Data must be able to be returned in formatted or unformatted flat file.
- Report must be available for download for up to 120 days.

1.2 DRC Exception Report Process Outline

Generating this DRC exception report is a multi-step process. A user must first enter the report parameters through a custom coded ASP screen within the NSLDS II website. These parameters are passed via a stored procedure to a reports table in DB2. Next an Informatica process converts the contents of an existing Loan Record Detail Report file (selected based on the parameters of the user) to a report database table and inserts usage codes for each record. Depending on the user request, Informatica will either convert the report table to an unformatted flat file or a Java procedure will be called to generate a formatted flat file from the report table contents. The NSLDS II MicroStrategy website will link to the files created and all the users to download.

2 Report Parameter Process

A user will navigate to the report parameter page through the NSLDS II Website. Here the user will select to run a particular DRC exception report. The user will enter the report parameters that are stored in a table for each specific report request.

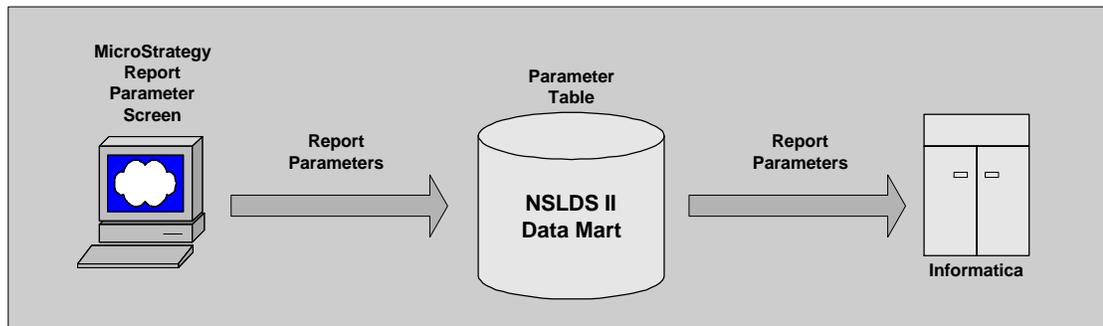


Figure 1, Report Parameter Process

2.1 Parameter Definitions

The parameter definitions for this report will be selected from a custom coded Microsoft ASP screen. Each parameter will be converted to an ASP form variable that is passed to an action page that passes each form variable to a stored procedure call statement. The stored procedure creates an ODBC connection to the DB2 server and inserts the data into a parameter table.

The following parameters are required on the FPPS Cohort Default Rate History parameter screen.

FPPS Cohort Default Rate History Parameters

Parameter	Description
Organization Type	This parameter will hold whether the organization is a school, guaranty agency, or a lender. This parameter will hold a value of 1 for a guaranty agency, and 2 for a lender.
Organization Code	This parameter holds the code of the guaranty agency or lender that the user wants to search on.
Sort By	This parameter allows the user to choose the way the report is to be sorted. This report can be sorted in the following ways: <ol style="list-style-type: none"> 1. Social Security Number 2. Claim Code, Social Security Number

3. Loan Status, Social Security Number	
Fiscal Year	This parameter holds the cohort fiscal year(s) that the user would like to search on.
Rate Type	This parameter holds whether the rate type is official or draft.
Loan Type	This parameter holds whether the user is choosing between originating or current lenders. This parameter will only hold a value when the user is searching on lenders.
Output Type	This parameter would hold whether the report would be formatted or unformatted.

2.2 Parameter Screen

Please refer to the NSLDS II Reengineering Screens Detail Design: Reports document for further details on this parameter screen.

2.3 Parameter Table Stored Procedure

A stored procedure populates a parameter table, called DCR031_PARM, with the report parameters from the ASP action page. This table is specifically created for the FPPS Cohort Default Rate History Report's parameters. A DB2 trigger will be created to call an Informatica process that will extract the data elements from the parameter table. The DB2 trigger will also pass the values for the user ID, report ID, and the report timestamp value within the parameter table. These three values are needed by Informatica to identify the correct record.

DCR031_PARM Table

Column	Type	Length	Description
ID	N	6	This column is the primary key of the table. This number uniquely identifies each row. This ID creases incrementally by 1 with each new row. This column is system generated.
USER_ID	C	8	This column records the user ID of the individual requesting the report.
RPT_ID	C	6	This column stores the report ID. Example value of this column would be 'DRC031'. Each DRC report has a unique ID.
TIMESTAMP	D	8	This column holds the time stamp when the report request is generated.
ORG_TYPE	C	1	This column holds the organization type for a school, guaranty agency, or lender. If the user selects guaranty agency then the organization type will be a 1 and for lenders it will store a 2.
ORG_CODE	C	6	This column holds the organization code for a particular school, guaranty agency, or lender.
FISCAL_YR	N	4	This column holds the fiscal year the user wants to retrieve rates for.
RATE_TYPE	C	8	This column determines whether the report data will come from an official or draft file.

Column	Type	Length	Description
LEN_FLAG	C	1	If the user retrieves information on a lender then this column determines if the loan information retrieved will be an originating or current lender. This column will be null if the user chooses information on a school or guaranty agency.
SORT_BY	C	1	This column holds the value of the loan sort criteria entered in by the user. This value will be a 1, 2, or 3. 1. Social Security Number 2. Claim Code, Social Security Number 3. Loan Status, Social Security Number
FLAG	C	1	This column helps Informatica determine which set of report parameters to take action on. It will store either Y or N. Y means that Informatica has yet to create a flat file for that particular report request. After it has taken action it updates the column to N. N means that Informatica has taken action on the report request.
OUTPUT	C	1	This column holds the indicator on how the report is formatted, either a flat file or a formatted file. A value 0 means unformatted and a value of 1 means formatted.

3 Informatica Report Table Generation

After Informatica has calculated the usage codes and populated the output table, either a formatted or unformatted flat file will be generated. The type of file generation will depend on the value of the output type filed on the report table.

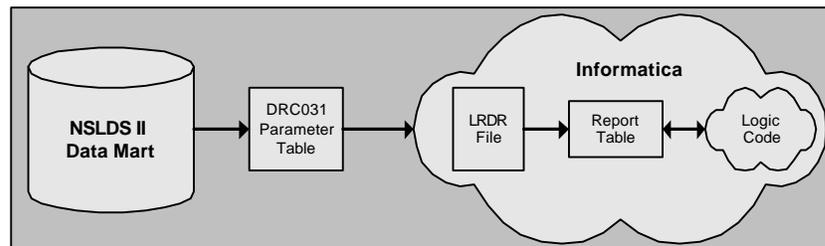


Figure 2, Report Table Generation Process

3.1 LRDR File Retrieval

Before Informatica can start pulling the relevant loan detail records from the LRDR, it must find the appropriate LRDR file. Numerous LRDR reports from previously run cohort default rate calculations are stored on the DB2 database server. Since users can select multiple fiscal years, Informatica will have to treat each fiscal year request as a separate report request, meaning each fiscal year chosen will have its own separate output file.

For each row in the parameter table Informatica reads the fiscal year, rate type, and lender flag parameters. Located on the DB2 database server will be an individual LRDR file for that specific year, both official and draft, originating and current loan holders for lenders.

Logic for this process:

Informatica reads parameter row columns Fiscal Year, Rate Type, and Lender Flag

Informatica takes Fiscal Year and Rate Type then searches for corresponding LRDR on server.

If Organization Code = 1 (GA)

Informatica searches for the Official Rate type for fiscal year requested.

If Organization Code = 2 (Lender)

Informatica searches for the LRDR based on if the Lender Flag column is an originating lender request or current lender holder request.

3.2 LRDR Data Extraction

The layout of the LRDR flat file is described below:

Date Element	Field Position	Length
Fiscal Year	1 - 4	4
School	5 - 10	6
School Branch	11 - 12	2
Current Lender	13 - 18	6
Originating Lender	19 - 24	6
GA Code	25 - 27	3
Student SSN	28 - 36	9
Student Sequence Number	37 - 40	4
Loan Number	41 - 49	9
Loan Sequence Number	50 - 53	4
Loan Supp Number	54 - 57	4
Loan Maturity Date	58 - 65	8
Out Balance	66 - 71	6
Lender Last Resort	72 - 72	1
Current Loan Status	73 - 74	2
Claim Date	75 - 82	8
Claim Reason	83 - 84	2
Loan Type	85 - 86	2
Loan Status Date	87 - 94	8
Original School	95 - 100	6
Original School Branch	101 - 102	2
Student Current Last Name	103 - 137	35
Student Current First Name	138 - 149	12
Student Current Middle Initial	150 - 150	1

Date Element	Field Position	Length
Student Current Date of Birth	151 – 158	8
Student Current SSN	159 – 167	9
Class Begin Date	168 – 175	8
Class End Date	176 – 183	8
Academic Level	184 – 184	1
Loan Amount	185 – 190	6
Guaranty Date	191 – 198	8
Consolidated Loan Number	199 – 207	9
Consolidated Loan Sequence Number	208 – 211	4
Consolidated Loan Supp Number	212 – 215	4
PC Consolidated Timestamp	216 – 235	20
PC Consolidated Sort	236 – 236	1
Original PC GA Code	237 – 239	3
Original PC Current Lender	240 – 245	6
Original Current Maturity Date	246 – 253	8
Student Stat Code	254 – 254	1
Student Stat Date	255 – 262	8
Program Type	263 – 263	1
Cumulative Amount	264 – 269	6
Filler	270 - 275	6

Informatica will extract data from this LRDR based upon the parameters that were pulled from the parameter table for the corresponding DRC report.

The logic for this extraction:

Pull LRDR dataset

* If Organization Code = Guaranty Agency

Pull LRDR dataset

Where LRDR.GA Code = (Parameter Table:Organization Code)

* If Organization Code = Lender

Since the user has the option of choosing to filter this report by originating or current lenders Informatica will need to evaluate the lender in the LRDR to designate if they are the original or current lender for that loan.

If filtering on Originating Lender

Pull LRDR dataset

Where LRDR.Originating Lender = (Parameter Table:Organization Code)

Pull LRDR dataset
Where LRDR.Current Lender = (Parameter Table:Organization Code)

Informatica pulls all of the LRDR data where the organization codes match and converts the LRDR contents into a database table.

3.3 Report Table Layout

The extracted data will be inserted into a database table that is specifically created for this report. The columns, user ID, report ID, report timestamp, output type, and sort code will be added to help uniquely identify each report. Columns for Usage 1 and Usage 2 codes will also be created.

The table below defines the DRC031 report table with column names and lengths.

Date Element	Column Name	Length
User ID	USER_ID	10
Report ID	RPT_ID	10
Report Timestamp	RPT_TIMESTAMP	8
Sort Code	SORT_CODE	1
Output Type	OUTPUT_TYPE	2
Fiscal Year	FISCAL_YR	4
School	SCH_CODE	6
School Branch	SCH_BR_CODE	2
Current Lender	CURR_LEN_CODE	6
Originating Lender	ORIG_LEN_CODE	6
GA Code	GA_CODE	3
Student SSN	STU_SSN	9
Student Sequence Number	STU_SEQ_NO	4
Loan Number	LOAN_NO	9
Loan Sequence Number	LOAN_SEQ_NO	4
Loan Supp Number	LOAN_SUPP_NO	4
Loan Maturity Date	LOAN_MAT_DT	8
Out Balance	OUT_BAL	6
Lender Last Resort	LEN_LST_RES	1
Current Loan Status	CURR_LOAN_ST	2
Claim Date	CLAIM_DT	8
Claim Reason	CLAIM_RSN	2
Loan Type	LOAN_TYPE	2
Loan Status Date	LOAN_ST_DT	8
Original School	ORIG_SCH	6
Original School Branch	ORIG_SCH_BR	2
Student Current Last Name	STU_CURR_LST_NM	35
Student Current First Name	STU_CURR_FST_NM	12
Student Current Middle Initial	STU_CURR_MID_NM	1

Date Element	Column Name	Length
Student Current Date of Birth	STU_CURR_DOB	8
Student Current SSN	STU_CURR_SSN	9
Class Begin Date	CLASS_BEG_DT	8
Class End Date	CLASS_END_DT	8
Academic Level	ACAD_LVL	1
Loan Amount	LOAN_AMT	6
Guaranty Date	EST_DT	8
Consolidated Loan Number	CON_LOAN_NO	9
Consolidated Loan Sequence Number	CON_LOAN_SEQ_NO	4
Consolidated Loan Supp Number	CON_LOAN_SUPP_NO	4
PC Consolidated Timestamp	PC_CON_TMSTMP	20
PC Consolidated Sort	PC_CON_SORT	1
Original PC GA Code	ORIG_PC_GA_CODE	3
Original PC Current Lender	ORIG_PC_CURR_LEN	6
Original Current Maturity Date	ORIG_CURR_MAT_DT	8
Student Stat Code	STU_STAT_CODE	1
Student Stat Date	STU_STAT_DATE	8
Program Type	PROG_TYPE	1
Cumulative Amount	CUM_AMT	6
Usage 1 Code	USE_1	1
Usage 2 Code	USE_2	1

3.4 Usage Code Determination

The report table will also include two columns for the usage codes, Usage 1 and Usage 2. Usage codes are used to determine how individual loans are used in determining cohort default rates. The usage code columns will is determined from a series of code logic statements that are run from a procedure. This procedure will evaluate every row in the report table and assign it a usage 1 code and usage 2 code.

3.4.1 Usage Code Requirements

This section highlights the requirements that were used to assigning usage codes.

Usage 1 Codes

Code	Description
D	Denominator only
B	Both Numerator and Denominator
N	Not Used
E	Eligible, but not counted

Usage 2 Codes

Code	Description
FD	Denominator
FB	Numerator and Denominator
DD	Denominator
DB	Numerator and Denominator
N	Not included in the cohort default rate calculation
E	Eligible, but not counted or used in the cohort default rate calculation.

Denominator

1. Exclude loans with loan status codes of:

- AL
- CA
- UA
- UB
- UC
- UD
- UI

2. Exclude Claim Reason Codes of:

- CS
- FC

3. Include Loan Type Codes of:

- SF
- SU
- SL
- D1
- D2

Numerator

1. For any FFEL loans included in the denominator:

- Include loans with a loan status of DF

2. For any Direct Loans included in the denominator:

- Include loans with a loan status of IX

3.4.2 Logic Code to Determine Usage 1 Codes

The following logic is used to determine usage 1 codes. This is a loop procedure. At the end of each iteration the usage codes will be populated into the row that was just evaluated.

Initialize Variables X, Y, W, Z, U to 0

Begin Loop through each row in table

Set X = School Code
Set Y = Student SSN

If X = W AND Y = Z (Same School, Same Student)

If Loan Status = (AL, CA, UA, UB, UC, UD, UI) OR Claim Reason = (CS, FC)
Set Usage_1_Code = N

Else

If Loan Type = (SF, SU, SL, D1, D2)

If Loan Type = CL, PL, RF, SF, SU) AND Loan Status = DF

If U = B OR U = D

Set Usage_1_Code = E

Else

Set Usage_1_Code = B

Set U = B

End

Else

If U = B OR U = D

Set Usage_1_Code = E

Else

Set Usage_1_Code = D

Set U = D

End

End

If Loan Type = (D1, D2, D4, D5, D6, D7) AND Loan Status = (IX)

If U = B OR U = D

Set Usage_1_Code = E

Else

Set Usage_1_Code = B

Set U = B

End

Else

If U = B OR U = D

Set Usage_1_Code = E

Else

Set Usage_1_Code = D

Set U = D

End

End

Else

Set Usage_1_Code = N

```
End

End

Else (Different Student)

  If Loan Status = (AL, CA, UA, UB, UC, UD, UI) OR Claim Reason = (CS, FC)
  Then
    Set Usage_1_Code = N
  Else
    If Loan Type = (SF, SU, SL, D1, D2)

      If Loan Type = (CL, PL, RF, SF, SU) AND Loan Status = DF
      Set Usage_1_Code = B
      Else
      Set Usage_1_Code = D
      End

      If Loan Type = (D1, D2, D4, D5, D6, D7) AND Loan Status = (IX)
      Set Usage_1_Code = B
      Else
      Set Usage_1_Code = D
      End

    Else
      Set Usage_1_Code = N
    End
  End
End

End
```

Insert Usage_1_Code into Usage 1 Code column in report table

Set W = School Code

Set Z = Student SSN

End Loop

3.4.2.1 Logic Code to Determine Usage 2 Codes

The following logic is used to determine usage 2 codes. This is a loop procedure. At the end of each iteration the usage codes will be populated into the row that was just evaluated.

Initialize X, Y, W, Z, U = 0

Begin Loop for each row in table

Set X = School Code

Set Y = Student SSN

If X = W AND Y = Z (Same School, Same Student)

 If Loan Status = (AL, CA, UA, UB, UC, UD, UI) OR Claim Reason = (CS, FC)

 Then

 Set Usage_2_Code = N

 Else

 If Loan Type = (SF, SU, SL, D1, D2)

 If Loan Type = (CL, PL, RF, SF, SU) AND Loan Status = DF

 If U = FB OR U = FD

 Set Usage_2_Code = E

 Else

 Set Usage_2_Code = FB

 Set U = F B

 End

 Else

 If U = FB OR U = FD

 Set Usage_2_Code = E

 Else

 Set Usage_2_Code = FD

 Set U = FD

 End

 End

 If Loan Type = (D1, D2, D4, D5, D6, D7) AND Loan Status = (IX)

 If U = DB OR U = DD

 Set Usage_2_Code = E

 Else

 Set Usage_2_Code = DB

 Set U = DB

 End

 Else

 If U = DB OR U = DD

 Set Usage_2_Code = E

 Else

 Set Usage_2_Code = DD

```
                Set U = DD
                End
            End

            Else
                Set Usage_2_Code = N
            End

        End

    Else (Different Student)

        If Loan Status = (AL, CA, UA, UB, UC, UD, UI) OR Claim Reason = (CS, FC)
            Then
                Set Usage_2_Code = N
            Else
                If Loan Type = (SF, SU, SL, D1, D2)

                    If Loan Type = (CL, PL, RF, SF, SU) AND Loan Status = DF
                        Set Usage_2_Code = FB
                    Else
                        Set Usage_2_Code = FD
                    End

                    If Loan Type = (D1, D2, D4, D5, D6, D7) AND Loan Status = (IX)
                        Set Usage_2_Code = DB
                    Else
                        Set Usage_2_Code = DD
                    End

                Else
                    Set Usage_2_Code = N
                End
            End
        End

    End

    End

    Insert Usage_2_Code into Usage 1 Code column in report table

    Set W = School Code
    Set Z = Student SSN

    Update Usage Two column in flat file = Usage_2_Code

    End Loop
```

Informatica reads Output Type

4 File Generation

After the file is created, Informatica will trigger a post-process script file that will check if a directory on the database server with the same name as the user_id parameter exist. If the directory does exist, the file will be stored in that directory. If the directory does not exist, it will create the directory and store the file in it. This process will then execute a store-procedure to remove the report row from its Parameter Table.

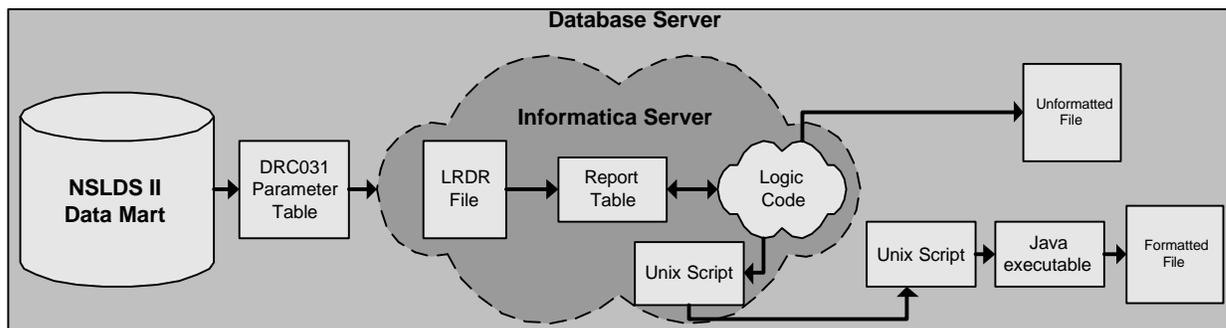


Figure 3, Flat File Generation Process

The following logic is used:

If Output Type = 0

Call Informatica to generate unformatted flat file.

If Output Type = 1

Call Java procedure to generate formatted flat file.

4.1 Informatica Unformatted File Generation

Informatica will extract data from the report table and generate the unformatted flat file. Upon completion the file will be housed in the exception reports file directory on the DB2 database server. This file will also be sorted depending on the sort code that the user entered.

Logic as follows:

If Sort Code = 1

Sort file by Student SSN

If Sort Code = 2

Sort file by Claim Code, Student SSN

If Sort Code = 3

Sort file by Loan Status, Student SSN

This file will have a unique naming convention based on the user ID, report ID, and report timestamp.

Naming format: User ID + Report ID + Report Timestamp

After the file is created Informatica will trigger a post-process script file that will check if a directory on the database server with the same name as the user_id parameter exist. If it does exist then the file will be stored in that directory. If the file does not exist then it will create the directory and store the file in it.

4.2 Java Formatted File Generation

If the output type indicates the exception report has to be formatted, then:

1. Informatica will trigger a post-process script.
2. The post-process script will call a Unix script on the database server and pass the following input parameters:
 - user id
 - report id
 - report timestamp.
3. The post-process script will execute a 'wait' command for the DB2 script to complete.
4. The DB2 script will be executed as a background process to reduce CPU utilization. The DB2 script will execute a java process on the database server and wait for it to end.
5. The DB2 Unix script will pass the input parameters to the Java report executable, ReportGenerator.
6. The ReportGenerator object will call the ReportReader to generate the SQL statement using the input parameters (select * from report table where USER_ID = user_id, RPT_ID=report_id, and TIMESTAMP=report_timestamp).
7. The ReportReader will then access the Persistence Framework (defined in the Screens Application Architecture Detailed Design document), to obtain a WorkUnit containing the result set and return the results to the ReportGenerator.
8. The ReportGenerator will use the ReportWriterFactory to create a ReportWriter based on the report ID. The ReportWriter object will be called to create a file and write the file header, page headers, row headers, page footers, and file footer while iterating through the result set.
9. The File will check if a directory on the database server with the same name as the user_id parameter exists. If the directory exists, it will store the file in that directory. If the directory doesnot exist, it will create the directoryand store the file in it.
10. Once the Java executable has exited, the DB2 script can exit and processing control will return to the post-process script on the Informatica server. This process will then execute a store-procedure to remove the report row from its Parameter Table.

This file will have a unique naming convention based on the user ID, report ID, and report timestamp.

Naming format: User ID + Report ID + Report Timestamp

5 Exception Report Download Process

Exception Reports will be stored on the DB2 database server in a file directory specifically reserved for exception reports. This directory will have a separate folder structure for each user that requests an exception report. Users will be able to retrieve and download these reports from the NSLDS II website interface through an FTP connection to the database server. Users will have access to these reports up to 120 days after they have been created.

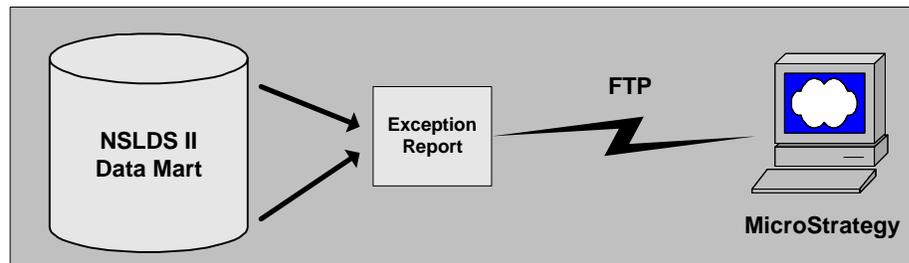


Figure 4, Report Download Process

In the MicroStrategy Web environment, there will be a specific screen where users will be able to retrieve their exception reports for download. On this screen will be an ASP code module that checks the DB2 database server file directory for any exception reports that have been generated each time the exception reports download page is accessed.

Displayed on this screen will be a dropdown box that will perform the following logic to only display those exception reports that the specific user has requested. This logic would check all files stored for the user inside their respective database file directory folder and match the user name embedded within file name to the current MicroStrategy user name:

```
If (User Name File on data set) = (MicroStrategy User Name)
    Display Exception Report Name in dropdown box
Else
    Do not display Exception Report Name in dropdown box
End
```

Once the dropdown is completely populated the user will select the requested report. The user will be prompted for their download directory on their computer where they can view the file.