

Chapter 10

NSLDS Database Tables

- 10.1 Database Overview
- 10.2 Entity/Attribute Table
- 10.3 Table Relationships and Keys
- 10.4 Query Index List

Contents

Chapter 10: NSLDS Database Tables

10.1 Database Overview	10-1
10.1.1 NSLDS—Two Databases in One.....	10-1
10.1.2 How NSLDS Data Is Organized.....	10-5
10.1.3 Table Relationships and Keys.....	10-7
10.1.3.1 Parent Tables and Primary Keys.....	10-7
10.1.3.2 Dependent Tables and Foreign Keys	10-7
10.1.4 Table Indexes	10-7
10.1.5 Data Model	10-9
10.1.6 System Hierarchy.....	10-11
10.2 Entity/Attribute Table.....	10-20
10.3 Table Relationships and Keys.....	10-110
10.4 Query Index List	10-195

Figures

Figure 10–1, OSAP Example.....	10–1
Figure 10–2, Database Tables.....	10–4
Figure 10–3, Date of Birth Cell	10–6
Figure 10–4, School Branch Identifier Cells	10–6
Figure 10–5, NSLDS Tables Relative to Unique and Non-Unique Indexes	10–8
Figure 10–6, Data Model Subject Areas.....	10–10
Figure 10–7, Business System Diagram, High Level	10–11
Figure 10–8, NSLDS Batch Program Architecture	10–12
Figure 10–9, Business System Diagram, System Support.....	10–14
Figure 10–10, Business System Diagram, Repository Build 1: Repository Archival and ED Systems	10–15
Figure 10–11, Business System Diagram, Repository Build 2: Data Providers.....	10–16
Figure 10–12, Business System Diagram, Reporting Capabilities 1: Loan Administration ...	10–17
Figure 10–13, Business System Diagram, Reporting Capabilities 2: Operations Support.....	10–18
Figure 10–14, Business System Diagram, Reporting Capabilities 3: Reports	10–19

10.1 Database Overview

This chapter lists the NSLDS database tables, their relationships to one another, and the indices for each table. The previous Web chapters explain how to use the NSLDSFAP Web site. Information displayed on the Web site answers most of the routine needs of the Title IV user community. For users requiring specific, tailored information that cannot be accumulated using standard online procedures, NSLDS also provides access to the database by means of custom, ad hoc queries. You can learn how to query the database in Chapter 11, QMF, and use this chapter for reference only.

10.1.1 NSLDS—Two Databases in One

NSLDS maintains an Active database and an Online Abstract database, each of which contains many tables.

Active Database

Use the table and attribute names provided in this chapter to access tables in the Active Database. The Active Database contains loan-level data for student accounts. The active database is a central repository that maintains loan-level and Pell grant information.

Online Abstract Database

The Online Abstract Database is a small subset of the NSLDS database containing loan, grant, and student data for a statistically valid, random sampling of 1.5 million borrowers selected from the Active Database. These tables can be accessed by typing **OSAP.** in front of their names in a query (Figure 10–1).

```
SELECT      GA_CODE
            SUM(CUM_PRIN_COLL_AMT)
FROM        OSAP.IRS_OFF
GROUP BY   GA_CODE
```

Figure 10–1, OSAP Example

Figure 10–2 lists the table names and briefly describes the contents of the NSLDS tables users can access by means of ad hoc queries and whether it's in the active database only or in the online abstract database as well. An asterisk in the High Volume column indicates that a table contains more than 5 million records and that queries run against it should be carefully written to avoid tying up system resources for excessive amounts of time.

Table Description	Table Name	High Volume	Database	
Accounting Calendar	ACCT_CALNDR		A	
Activity Function	ACT_FUNC		A	
Address Support	ADD_SUPT		A	

Table Description	Table Name	High Volume	Database	
Aggregate Description	AGG_DESCN		A	
Aggregate Descriptor	AGG_DESC		A	
Aggregate Descriptor Detail	AGG_DESC_DET		A	
Aggregate Descriptor Item	AGG_DESC_ITEM		A	
Aggregate Organization	AGG_ORG		A	
Aggregate Purpose	AGG_PURP		A	
Aid Overpayment	AID_OVRPMT		A	
Anticipated Completion	ANTIC_COMPL		A	
Certification	CERT		A	
Collection	COLL	*	A	O
Deferment Type	DFR_TYPE		A	O
Enrollment	ENRL		A	
Expected Student Enrollment	EXP_STU_ENRL	*	A	O
FDLP Servicer	FDSLP_SVR		A	O
FDLP Servicer Branch Holder	FDSLP_SVR_BR_HOL	*	A	O
FDLP Servicer Submittal History	FS_SBMTL_HIS		A	
FDLP Servicer Submittal Run Error	FS_SBMTL_RUN_ERROR		A	
Financial Profile	FINANC_PROF	*	A	O
Guaranty Agency	GA		A	O
GA Aggregate	GA_AGG		A	
GA Submittal History	GA_SBMTL_HIS		A	
GA Submittal Run Errors	GA_SBMTL_RUN_ERR	*	A	
GA Summary	GA_SUM		A	
Insurance Claim Payment	INSUR_CL_PMT	*	A	O
Insurance Claim Refund	INSUR_CL_RFD		A	O
IRS Offset	IRS_OFF		A	O
Lender	LEN	*	A	O
Lender Branch	LEN_BR		A	O
Lender Branch Aggregate	LEN_BR_AGG	*	A	
Lender Branch Holder	LEN_BR_HOL	*	A	O
Lender Branch Holder Servicer	LEN_BR_HOL_SVR	*	A	O
Lender Branch Servicer	LEN_BR_SVR		A	O
Lender Branch Servicer Aggregate	LEN_BR_SVR_AGG		A	
Lender Branch Transition History	LEN_BR_TRAN_HIS		A	O
Loan	LOAN	*	A	O
Loan Cancellation	LOAN_CAN	*	A	O
Loan Capitalized Interest	LOAN_CPTL_INT		A	
Loan Consolidation Application	LOAN_CNSL_APP		A	
Loan Default Rate Default	LOAN_DEF_RT_DEF		A	
Loan Deferment	LOAN_DFR	*	A	O
Loan Disbursement	LOAN_DIS	*	A	O
Loan External Identification	LOAN_EXTL_ID	*	A	
Loan Forgiveness	LOAN_FRGV		A	
Loan Grouping	LOAN_GP	*	A	O
Loan Guarantor	LOAN_GUA	*	A	O

Table Description	Table Name	High Volume	Database	
Loan Maturity Date History	LOAN_MAT_DT_HIS	*	A	O
Loan Origination Support	LOAN_ORIGN_SUPP		A	
Loan Refund	LOAN_RFD		A	O
Loan Repayment Plan	LOAN_RPMT_PLAN	*	A	O
Loan Status	LOAN_STAT	*	A	O
Loan Supplement	LOAN_SUPP	*	A	
Loan Transfer Log	LOAN_TRAN_LOG		A	
Loan Type	LOAN_TYPE		A	O
Loan Type Group	LOAN_TYPE_GP		A	
Loan Unpaid Refund Discharged	LOAN_UNPD_RFD_DCHG		A	
National Aggregate	NATL_AGG		A	
Output Distribution	OPUT_DISTR		A	
Organization Contact	ORG_CON		A	
PCA SPA Notification	PCA_SPA_NOTIF	*	A	O
Pell Grant	PELL_GRT	*	A	O
PLUS Borrower	PLUS_BOR		A	O
PLUS Borrower Loan	PLUS_BOR_LOAN		A	O
PLUS Borrower Name	PLUS_BOR_NM		A	O
PLUS Borrower Social Security Number	PLUS_BOR_SSN		A	O
Portfolio	PORT		A	
Portfolio Participation	PORT_PARTIC		A	
Portfolio Participation Forced	PORT_PARTIC_FCD		A	
Portfolio Status	PORT_STAT		A	
Portfolio Unresolved Student	PORT_UNRL_STU		A	
Preferred School	PREF_SCH	*	A	O
Prescreening Applicant	PRSCRN_APPL	*	A	
Prescreening Result	PRSCRN_RSLT	*	A	
Prescreening Result Loan	PRSCRN_RSLT_LOAN	*	A	
Prescreening Result Pell	PRSCRN_RSLT_PELL		A	
Region Aggregate	REG_AGG		A	
Reinsurance Claim Payment	REINSUR_CL_PMT	*	A	O
Reinsurance Claim Refund	REINSUR_CL_RFD		A	O
Repurchased Loan	RPCH_LOAN		A	O
Rate	RT		A	
School	SCH		A	O
School Branch	SCH_BR		A	O
School Branch Aggregate	SCH_BR_AGG	*	A	
School Branch Code Cross Reference	SCH_BR_CODE_XREF		A	
School Branch Direct Loan Origination	SCH_BR_DLOAN_ORIGN		A	
School Branch DUNS Number	SCH_BR_DUNS_N0		A	
School Branch Program	SCH_BR_PROG		A	
School Branch Holder Servicer	SCH_BR_HOL_SVR	*	A	O
School Branch Status History	SCH_BR_STAT_HIS		A	
School Branch Servicer	SCH_BR_SVR		A	O
School Branch Servicer Aggregate	SCH_BR_SVR_AGG		A	

Table Description	Table Name	High Volume	Database	
School Branch Transmittal History	SCH_BR_TRAN_HIS		A	O
School Origination History	SCH_ORIGN_HIS		A	O
School Submittal History	SCH_SBMTL_HIS		A	
School Submittal Run Errors	SCH_SBMTL_RUN_ERR	*	A	
SPA Payment	SPA_PMT		A	O
SSCR Cycle	SSCR_CYCLE		A	
SSCR Error Detail	SSCR_ERR_DET		A	
SSCR Event	SSCR_EVT		A	
SSCR Letter Generation	SSCR_LTR_GEN		A	
SSCR Performance	SSCR_PERF		A	
SSCR Process Waiver	SSCR_PROC_WAIV		A	
SSCR Regeneration	SSCR_REGEN		A	
SSCR Roster Current Value	SSCR_ROS_CURR_VAL		A	
SSCR Requested Return Summary	SSCR_RQTD_RTN_SUM		A	
SSCR Return History	SSCR_RTN_HIS		A	
SSCR Return Summary	SSCR_RTN_SUM		A	
SSCR Scheduled Distribution	SSCR_SCHED_DISTR		A	
SSCR Transmittal Detail	SSCR_TRSMTL_DET	*	A	
SSCR Transmittal History	SSCR_TRSMTL_HIS		A	
State Aggregate	ST_AGG		A	
Student	STU	*	A	O
Student Address	STU_ADD	*	A	
Student Branch	STU_BR	*	A	O
Student Branch Identifier	STU_BR_ID		A	
Student Demographics	STU_DEM	*	A	O
Student Name	STU_NM	*	A	O
Student Social Security Number	STU_SSN	*	A	O
Student Status	STU_STAT	*	A	O
Student Status Unresolved	STU_STAT_UNRESLV	*	A	
Supplemental Reinsurance Payment	SUPP_REINSUR_PMT	*	A	O
Transfer Monitoring Alert Run History	TM_ALERT_RUN_HIS		A	
Transfer Monitoring Event	TM_EVT		A	
Transfer Monitoring Inform Run Errors	TM_INFRM_RUN_ERROR		A	
Transfer Monitoring Inform Run History	TM_INFRM_RUN_HIS		A	
Transfer Monitoring School Transfer Profile	TM_SCH_XFER_PROF		A	
Transfer Monitoring Student	TM_STU		A	
Transfer Student Alert	TM_STU_ALERT		A	
Transfer Monitoring Student Alert Loan	TM_STU_ALERT_LOAN		A	
Transfer Monitoring Student Alert Name	TM_STU_ALERT_NM		A	
Transfer Monitoring Student Alert Overpayment	TM_STU_ALERT_OVRPMT		A	
Transfer Monitoring Student Alert Pell	TM_STU_ALERT_PELL		A	
Validation Translation Table	VAL_TRSL_TAB		A	O

Legend: A=Active Database, O=Online Abstract, High Volume=In excess of 5 million records

Figure 10–2, Database Tables

10.1.2 How NSLDS Data Is Organized

The NSLDS database was built with DB2, a relational database management system. In a relational database, data is organized into tables consisting of rows and columns. Each table stores information about one entity type. An entity type can be a collection of people, places, things, or events that share common definitions, relationships, and attributes. Examples of Title IV entity types include schools, students, loans, and GAs.

The rows of a table in a relational database represent the specific instances (or entities in relational database terms) of an entity type. They are similar to records in a traditional file structure. The School table, for example, contains one row for each school eligible (or previously eligible) to participate in Title IV programs.

The columns (attributes) of a table contain data values that describe the entity. Section 10.2 provides the following information for a detailed listing of NSLDS database tables and their attributes of for each table:

- Table Name and Database Location (Active or Online Abstract)
- Table Description
- Attribute/Column Name
- Attribute/Column Description
- Data Type (Text, Numeric, Date, Timestamp, Time)
- Field Length
- Indicators of Pre-Established Indexes

Columns (attributes) are similar to fields or data elements in a traditional file structure record. For example, the columns in the School table include the following:

- **CERT_TYPE**—A flag indicating whether an institution is eligible and certified to participate in Title IV program
- **CODE**—A six-digit ED code for uniquely identifying a school
- **NM**—The name of the school

Each intersection of a row and column is called a cell. The cell stores a value of the specific information defined by the intersection of row and column. The following are two examples:

Example 1: Date of Birth Cell—The Student table has a *row* for each person who is, or was, a student with Title IV aid, and a *column* for each attribute relating to individual students, such as name, Social Security Number, and date of birth. The cell that represents the intersection of the row for the student, John Doe, and the Date of Birth column might contain the value 1974-05-12 or May 12, 1974, as shown in Figure 10–3.

Title IV Aid Students		
Name of Student	Date of Birth	SSN
Alice Deer	March 20, 1949	123-45-6789
John Doe	May 12, 1974	234-56-7891
Walter Stag	December 7, 1969	345-67-8910

Figure 10–3, Date of Birth Cell

The attribute (or attributes) that uniquely identifies a row in a table is called the identifier or primary key. To select a specific row from a table, you must specify the identifier. The identifier representing a specific row must be different from every other identifier in that table.

Example 2: School Branch Identifier Cells—A School Branch is uniquely identified by its six-digit main campus identifier and its two-digit location code in the School Branch table. The six-digit main campus code 777777 and the two-digit location identifier 03, when combined as identifiers for a row, uniquely identify one institution. Other institutions can also be identified by the same main campus code 777777, but they must have a location code other than 03 to co-exist in the same School Branch table, as shown in Figure 10–4.

School Branch		
School Code	School Branch Code	School Branch Name
777777	00	University Of Education - Main Branch
777777	01	University Of Education – Location 01
777777	02	University Of Education – Location 02
777777	03	University Of Education - Location 03
777777	04	University Of Education - Location 04

Figure 10–4, School Branch Identifier Cells

In a relational database, information about related entities is often divided into separate tables. For instance, a PLUS borrower is related to one or more loans, but to store a date of birth with every PLUS loan would take up too much space. Instead, personal information such as the date of birth of a PLUS borrower is placed in one table, and all loan information for that borrower in another. The two are then connected, or joined, with a relationship.

The relational database environment supports relationships through linkages, called *joins*, on the attributes shared by tables.

10.1.3 Table Relationships and Keys

The NSLDS database consists of many tables that are related to one another as parents and children are; therefore, they are called *parent tables* and *dependent tables*. The relationship between a parent table and a dependent table is just like a familial relationship. A parent can have one or more children, and every child has a parent. A parent table can be a dependent to another parent table. A dependent table can be the parent to another dependent table. Therefore, these relationships are consistent for grandparents through grandchildren.

These relationships are defined by keys (primary and foreign) that point from one table to another. Primary and foreign keys for all tables available to users are presented in Section 10.3. Using primary and foreign keys properly will allow you to search the enormous amount of data stored in NSLDS more quickly.

10.1.3.1 Parent Tables and Primary Keys

The most significant characteristic of the parent table is its primary key. The primary key (identifier) is constructed from one or more columns (attributes) of the parent table. The primary key is a pointer to one unique row (occurrence) in the parent table. The sequence of the columns that compose the primary key is very important. A primary key always has an index defined upon its column(s) in the same exact order. In addition, the more columns of a primary key that are supplied an *equals to (=) value* in the key sequence, the quicker and more efficiently the query produces results from the table.

10.1.3.2 Dependent Tables and Foreign Keys

The dependent (child) table has a unique characteristic—a foreign key. A foreign key is a pointer to the primary key of the parent table. The columns of the foreign key match one-for-one the columns of the primary key of the parent table. The column names may be spelled differently, but the column characteristic must be identical. Column characteristics are the type (such as character, integer, date, or time), length, and scale of the data in the column. The dependent table foreign key may exist one or more times in the dependent table, but always has one—and only one—parent row (primary key) in the parent table.

10.1.4 Table Indexes

When writing queries to access various tables (entities) and particular columns (attributes) in those tables, you are directing the computer to search the entire database to retrieve specific information. This process is time consuming and expensive in terms of computer resources. To search and retrieve more efficiently, indexes were created to reduce the search effort. Section 10.4 lists the indexes available for writing queries. These indexes are also marked in Section

10.2 to show which tables and attributes have indexes, and which are primary and secondary. Also, the numbering indicates the sequence order of the columns (attributes) in the index.

An index is constructed of one or more column(s) from the associated table. Indexes are either *unique* or *non-unique*. A unique index must always exist for a primary key; it serves as a pointer to one unique row (occurrence) of the associated table. Likewise, a non-unique index is a pointer to one or more row(s) of the associated table.

Figure 10–5 shows three selected tables (center box) in juxtaposition with their related unique indexes (left box) and with a single non-unique index (right box). These unique indexes would clearly point to one and only one occurrence in the associated table, but the non-unique index (School Code and School Branch Code) could point to many occurrences in the Student Branch table.

Unique Index—Column List		Table—Column List		Non-Unique Index—Column List	
Index Name	Column Name	Table Name	Column Name	Index Name	Column Name
SCSCH101	CODE	SCH	CERT_TYPE CODE DEG_GRT_IND ETNC_CODE NM PROG_LGTH SCH_TYPE ST SYS_CR_DATE	none	
STSTBRI1	STU_NO STU_SEQ_NO SCH_CODE SCH_BR_CODE	STU_BR	SYS_CR_DT ANTIC_CDMP_DT STU_NO VAL_IND STU_SEQ_NO SCH_CODE SCH_BR_CODE	STSTBRI2	SCH_CODE SCH_BR_CODE
LOLDIS1	LOAN_NO ID_STU_SEQ_NO LOAN_SEQ_NO DT	LOAN_DIS	AMR_DIFF CUM_AMT CURR_LEN_CODE DT ID_STU_SEQ_NO LOAN_NO LOAN_SEQ_NO SYS_CR_DT	LOLOAN1	NO ID_STU_SEQ_NO SEQ_NO

Figure 10–5, NSLDS Tables Relative to Unique and Non-Unique Indexes

The sequence of the columns that compose an index is very important. In addition, the more columns of the index that can be supplied with an *equals to (=) value* in key-sequence order, in a search request, the faster the answer is retrieved from the table.

Effective indexes are illustrated below:

1. An index on a table is similar to the index of a textbook or encyclopedia. If you are looking for the page (unique)—or pages (non-unique)—to find the definition of a certain word within the book, you could read quickly or search the book looking for sections where the word is used. But, you would probably do better using the index (which is in

alphabetical order) to locate the word, retrieve the number(s) of the page(s) on which it is used, then go to the page(s) to find the definition of the word. It is the same when you ask the computer to find something. Although a computer can read extremely fast, it is still much more efficient to use an index.

2. Another example of an index is the card catalog system in your local library. If you are looking for a particular book, you go to the library card catalog (index), look up the location of the book using its author and title or Library of Congress cataloging number, and then go retrieve the book from the appropriate shelf location identified by the catalog.

To see the importance of the column sequence of an index, just imagine the same search for the library book assuming you only had the author's name. Even though you have the library card catalog, searching for and retrieving the book could be very time consuming because you might have to go through all of the cards under that author's name in the catalog looking for the book title you seek. Once you find the card with the matching author's name and book title, you only have to go to the shelf to locate the book. So supplying the author's name as the first part of the search is useful, but knowing the title enables you to find the book faster.

All of these principles are the same when you want to direct the computer to retrieve data. The more request values with an exact match, *equals to* (=), of the index columns in key sequence, the faster and more efficiently the computer can retrieve your data. Knowing the Library of Congress cataloging number for a book is like knowing a primary key (identifier) value for a table—it enables you to go right to where it is.

10.1.5 Data Model

The NSLDS data model help users better understand the basic structure of the NSLDS and thus be able to use its system capabilities more efficiently. The NSLDS data model was constructed using Advantage:Gen Computer Aided Software Engineering (CASE) tool. The Advantage:Gen data model is based on a hierarchy of objects. Higher level objects decompose into lower level objects; for example, a Advantage:Gen model name decomposes into subject areas, and subject areas decompose into entities and their relationships. The data model groups subject areas, and subject areas group closely related entities. Entities are collections of business data on which business functions operate. Data model diagrams depict relationships as lines that connect entities. Model names, subject areas, and entities are depicted as squares.

Business analysts use Advantage:Gen's data model diagrams to check the logical consistency of a database, while database designers use them to produce a physical database using DB2, ORACLE, or a similar product. The NSLDS data model encompasses the seven main subject areas described in Figure 10–6.

Subject Area	Description
Students	Individuals whose education or partial education is funded through Title IV aid programs.
Schools	Institutions of higher education that participate in Title IV aid programs.
Lenders	Financial institutions that lend money under Title IV aid programs.
Guaranty Agencies	Agencies that guarantee Title IV loans for lenders. These agencies also aid in the administration of collections of defaulted loans, reinsurance of defaulted loans, and disbursement of Supplemental Preclaim Assistance for the FFEL program.
Loans	Money borrowed for funding a pre-approved educational program including disbursements, cancellations, deferments, refunds, loan aggregates, and uncollectables. These do not include grants.
FDLP Servicer	Organization that services FDLP loans for ED.
Default Rates	The creation of default rates for lenders, schools, and GAs related to the organizations' loans in the FFEL program.

Figure 10–6, Data Model Subject Areas

10.1.6 System Hierarchy

The Batch table and figures that follow identify the individual programs and screens, which make up the NSLDS and depict their logical and hierarchical organization.

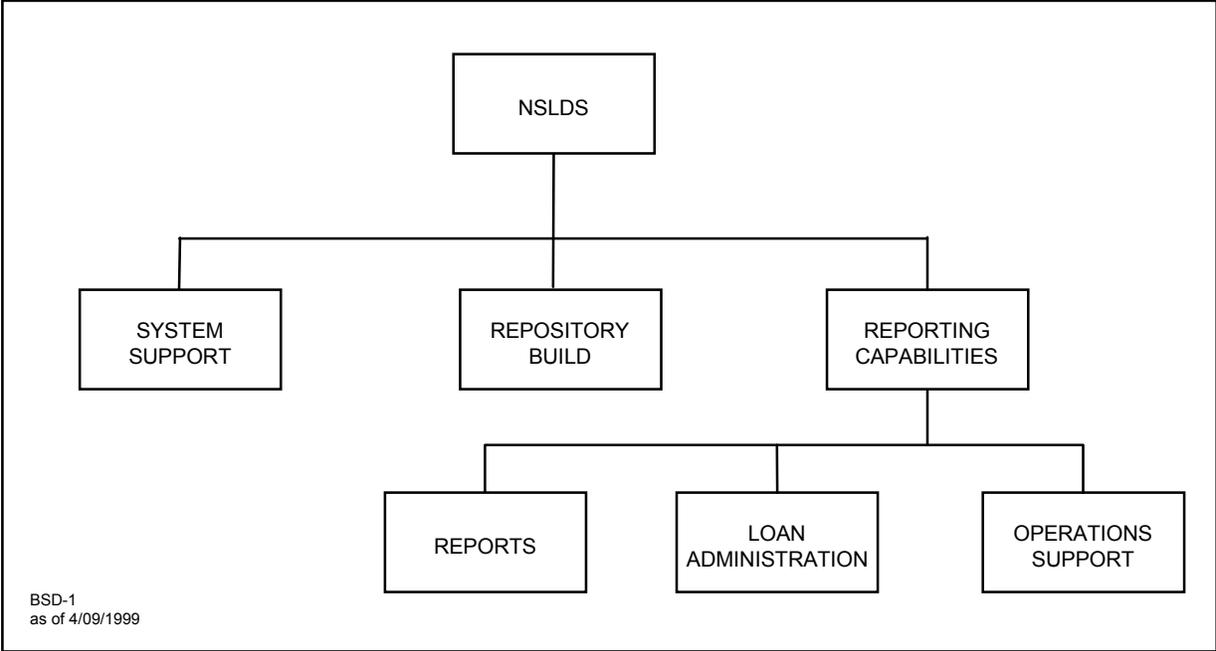


Figure 10–7, Business System Diagram, High Level

(Replace with **foldout** NSL-0835.flo)

Figure 10–8, NSLDS Batch Program Architecture

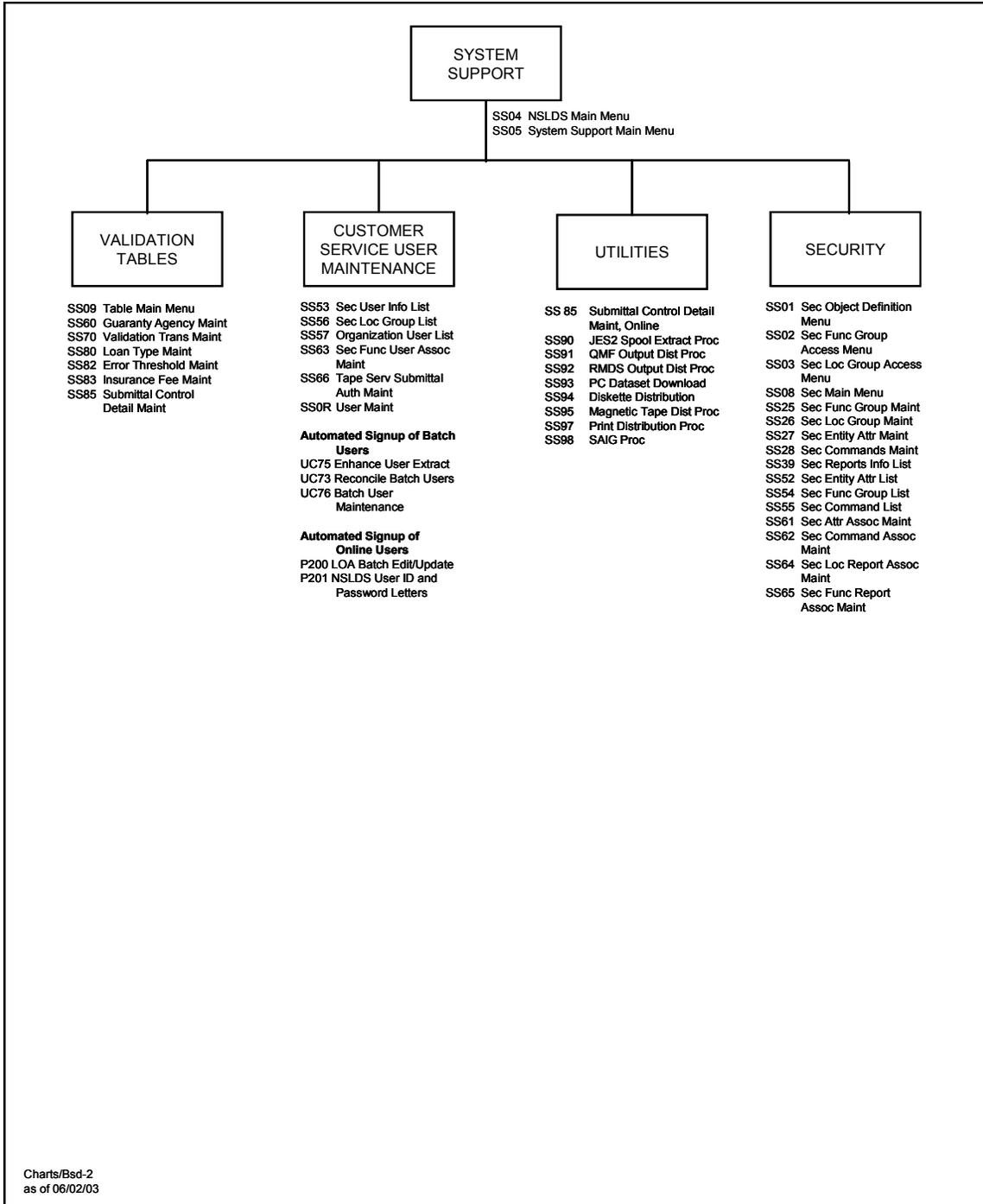


Figure 10–9, Business System Diagram, System Support

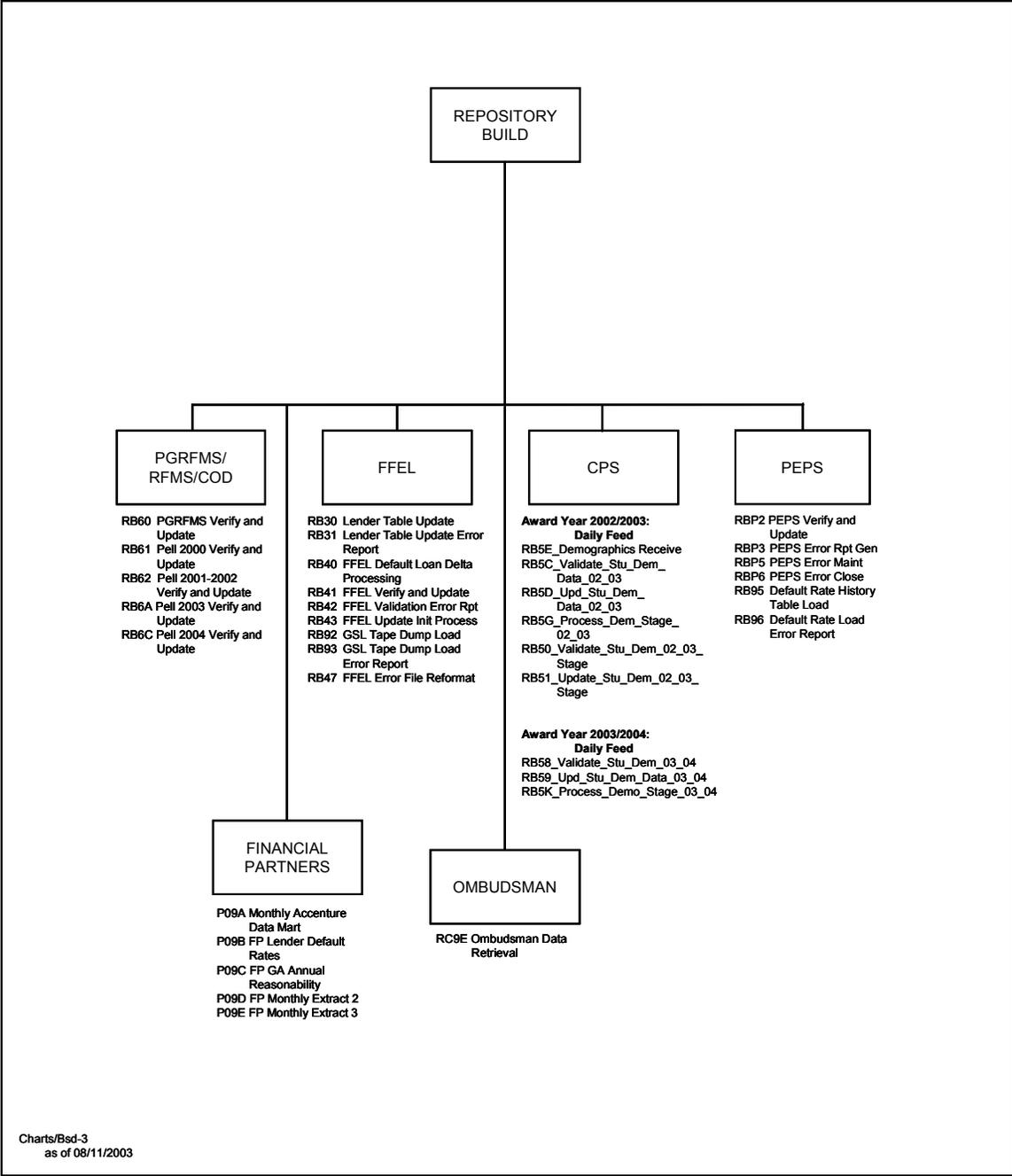


Figure 10-10, Business System Diagram, Repository Build 1: Repository Archival and ED Systems

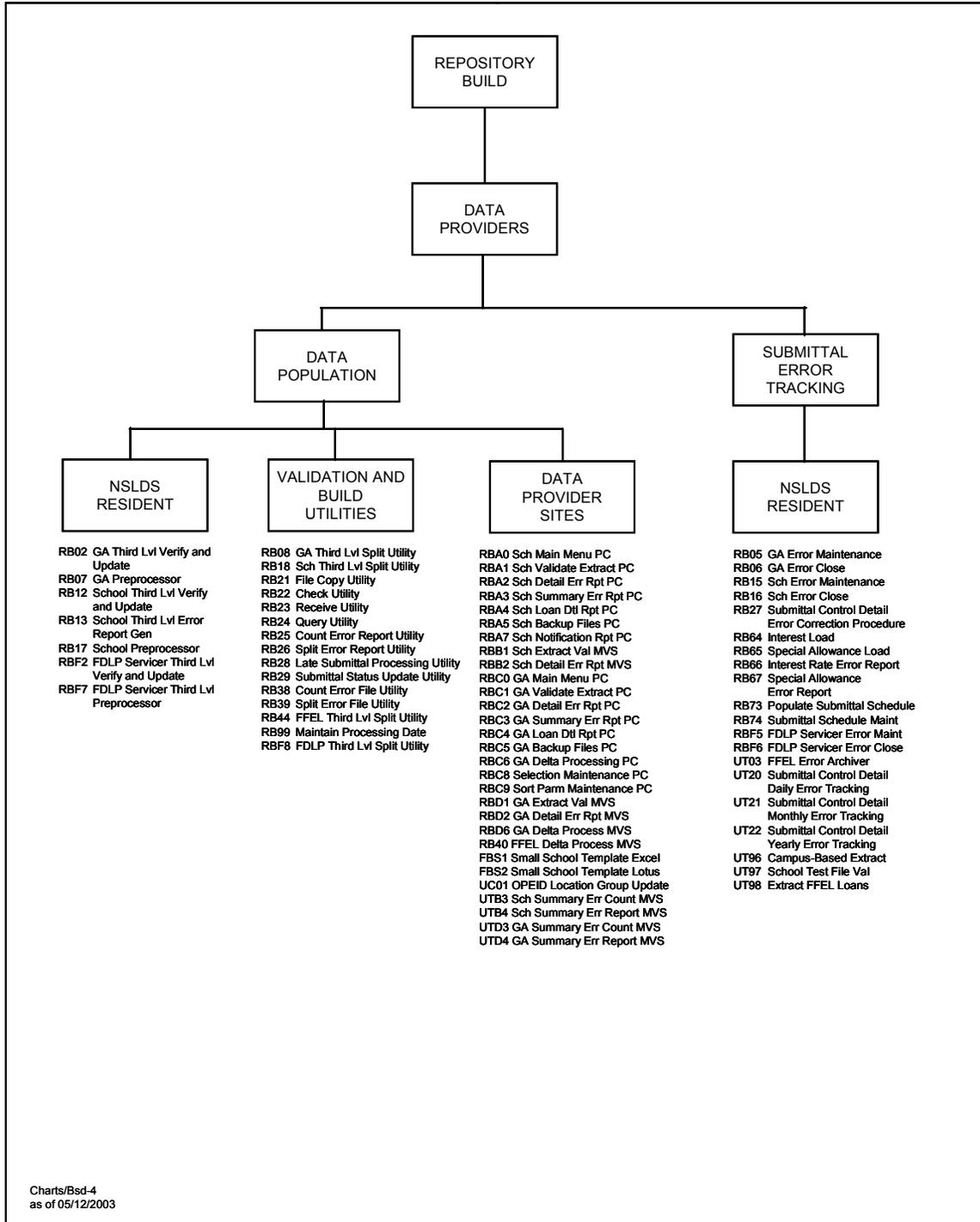


Figure 10–11, Business System Diagram, Repository Build 2: Data Providers

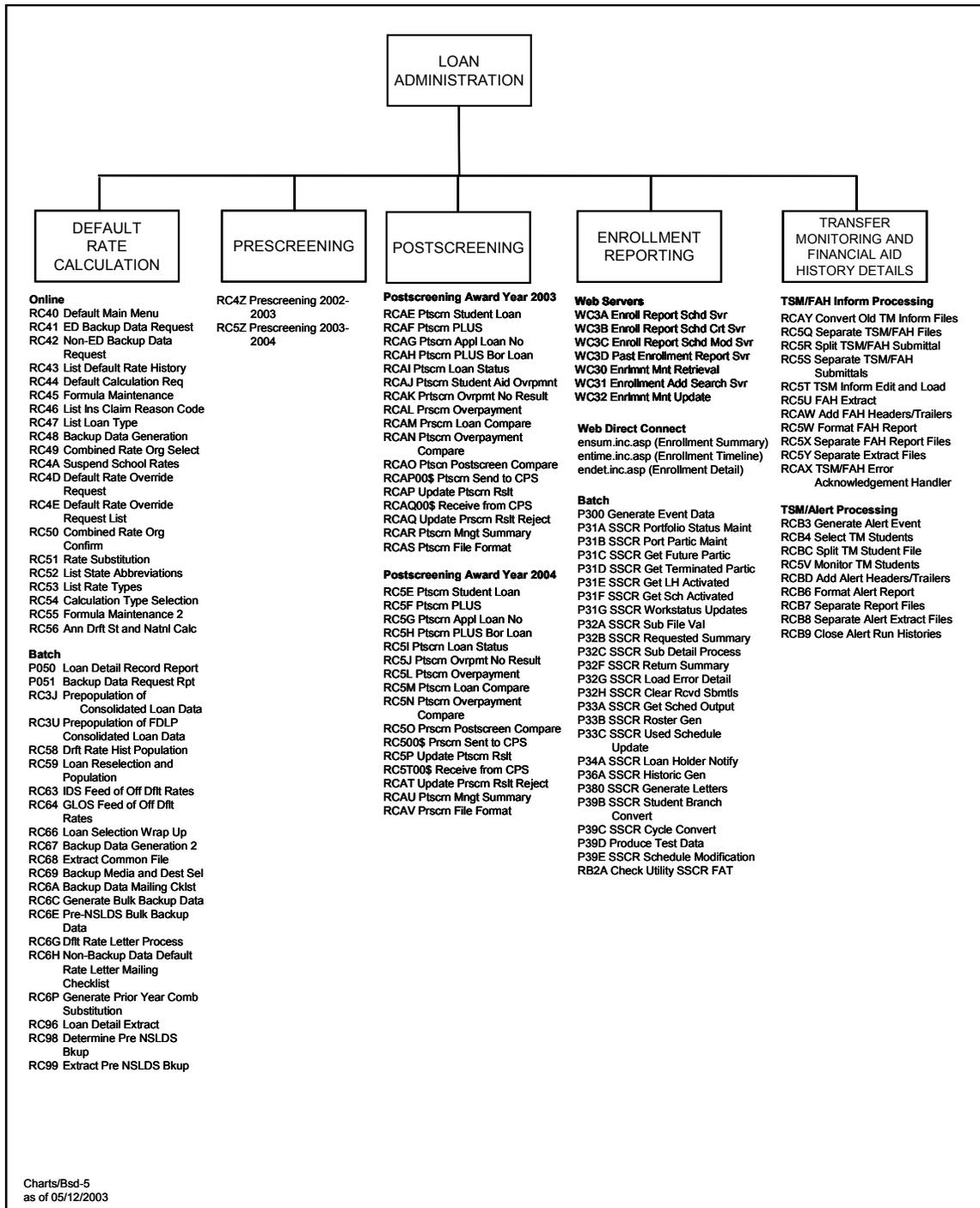


Figure 10–12, Business System Diagram, Reporting Capabilities 1: Loan Administration

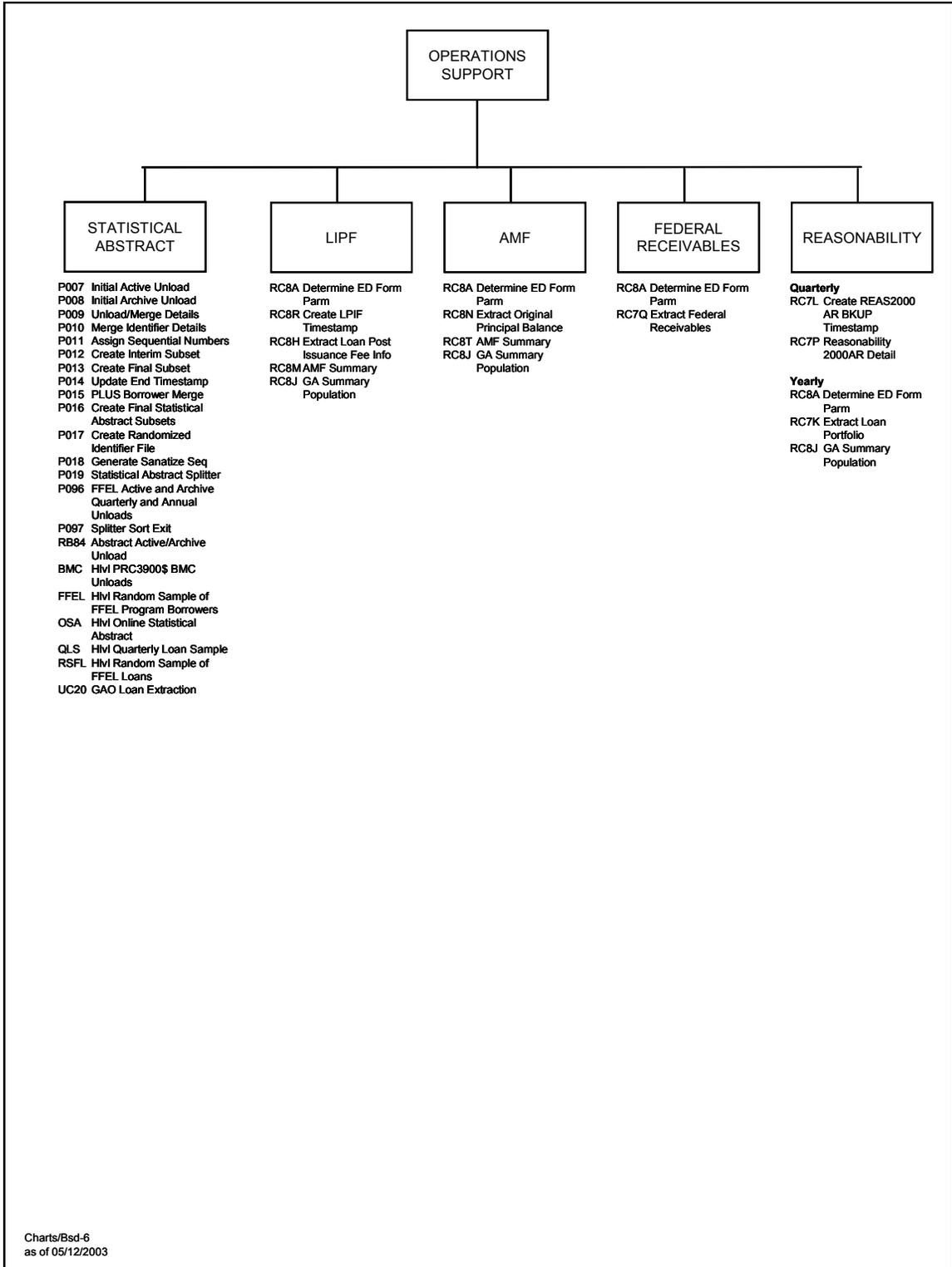


Figure 10–13, Business System Diagram, Reporting Capabilities 2: Operations Support

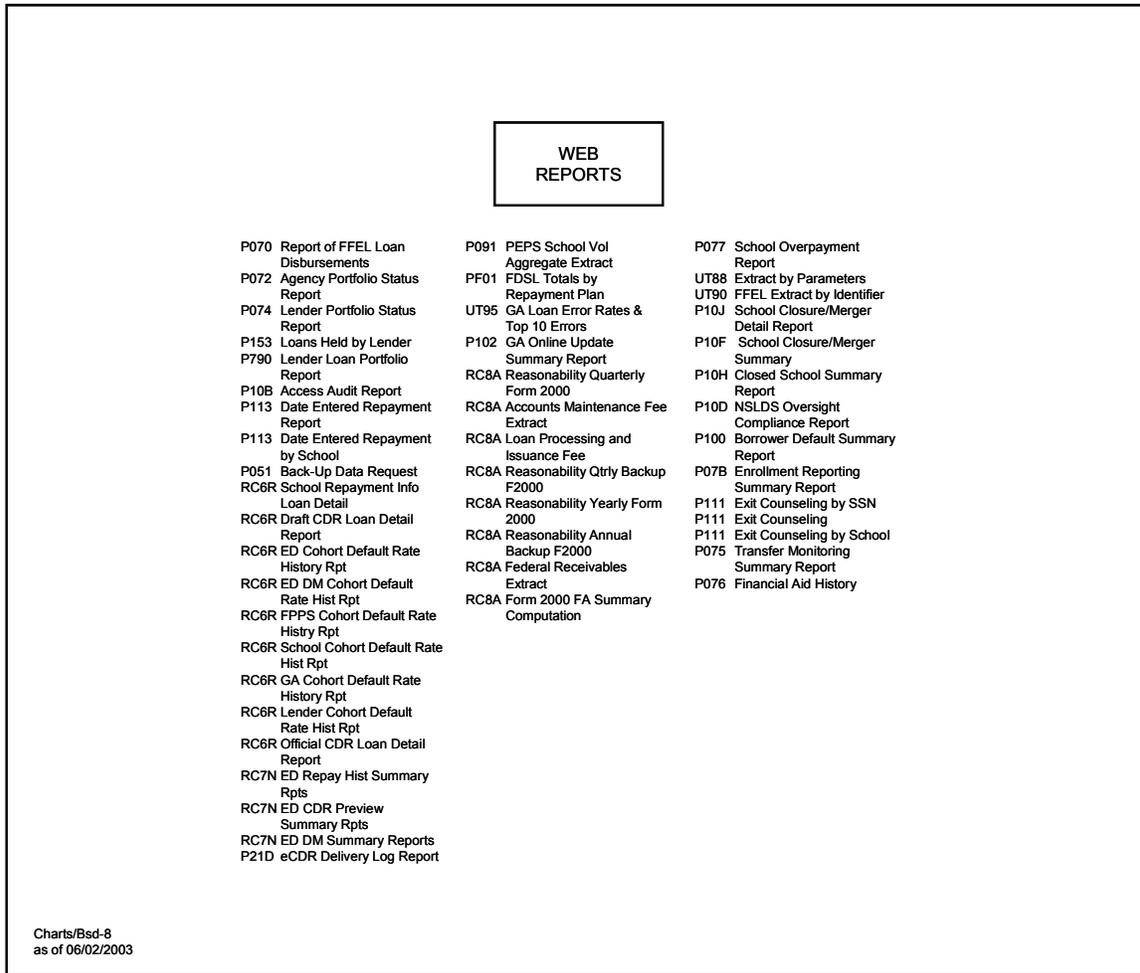


Figure 10–14, Business System Diagram, Reporting Capabilities 3: Reports