Banner Technical Training
Financial Aid Technical Training Workbook

Release 8.0 - April 2008
Updated 2/12/2008

SUNGARD HIGHER EDUCATION

What can we help you achieve?

SunGard Higher Education

4 Country View Road
Malvern, Pennsylvania 19355
United States of America
(800) 522 - 4827

Customer Support Center website

http://connect.sungardhe.com

Distribution Services e-mail address

distserv@sungardhe.com

Other services

In preparing and providing this publication, SunGard Higher Education is not rendering legal, accounting, or other similar professional services. SunGard Higher Education makes no claims that an institution's use of this publication or the software for which it is provided will insure compliance with applicable federal or state laws, rules, or regulations. Each organization should seek legal, accounting and other similar professional services from competent providers of the organization's own choosing.

Revision History Log

Publication Date	Summary
4/30/2008	New version that supports Banner Financial Aid 8.0 software.

Notice of rights

Think before you print.

Table of Contents

Introduction

Workbook goal

Participants in this course will be able to:

- identify Banner Financial Aid forms and tables
- query the Banner Financial Aid Technical tables
- follow key Financial Aid Technical processes
- identify and read reports, processes, procedures and scripts in Banner
- identify the Banner Financial Aid naming conventions, form types, directories and their contents, database structure, processes and procedures
- follow the basics of how Job Submission, Population Selection, and Letter Generation work.

Intended audience

The target audience for this training course is the Technical staff, "power users" and Financial Aid managers who may use or teach others about Banner Financial Aid Technical tables and processes.

Prerequisites

To complete this workbook, you should have

- completed OR101 (Introduction to Oracle)
- completed the *Banner 8 Fundamentals* Education Practices CBT or equivalent training in Banner navigation

Client Responsibilities

The client must complete several tasks before the training consultant arrives at the site to conduct the course.

Before you set up the current module, the following elements must be defined in the Banner Financial Aid System:

- The Banner Financial Aid System should be installed and readily available
- Training accounts should be accessible so that every class participant can connect to the Unix/VAX/NT machine
- Training accounts should be accessible so that every class participant can connect to SQL*Plus and select from any Banner table
- Participants should have access to the Banner Technical Reference Manual Financial Aid Supplement

If you are unable to complete these tasks before the course is scheduled, please contact your account consultant for assistance.

Introduction

Objectives

The attendees will receive a lecture and demonstration that describes the Banner Financial Aid Technical module.

At the end of this course, participants will be able to:

- identify Banner Financial Aid Technical forms and tables
- query the Banner Financial Aid Technical tables
- follow key Financial Aid Technical processes
- identify and read processes, procedures, and scripts in Banner Financial Aid Technical.

Topics include SunGard Higher Education services, Banner naming conventions, data directories and their contents, database structures, Job Submission, Population Selection, Letter Generation, conversions and resources for data and help.

Financial Aid Technical Training Overview

Introduction

This section provides a basic overview of Banner and its common conventions.

Objectives

At the conclusion of this chapter, participants will be able to:

- identify Banner products
- identify common Banner object naming conventions
- distinguish between Banner form types
- properly format various types of data in Banner
- work with PIDMs.

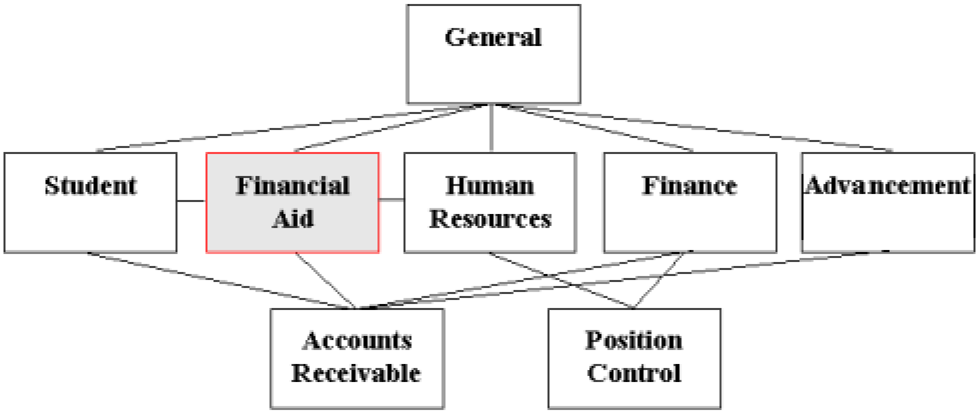
SunGard Higher Education's Education Centers

Benefits

- Provide a centralized location for training in Banner and Oracle
- Provide an environment where SunGard Higher Education clients can focus on learning
- Provide an opportunity to interact with colleagues from other institutions

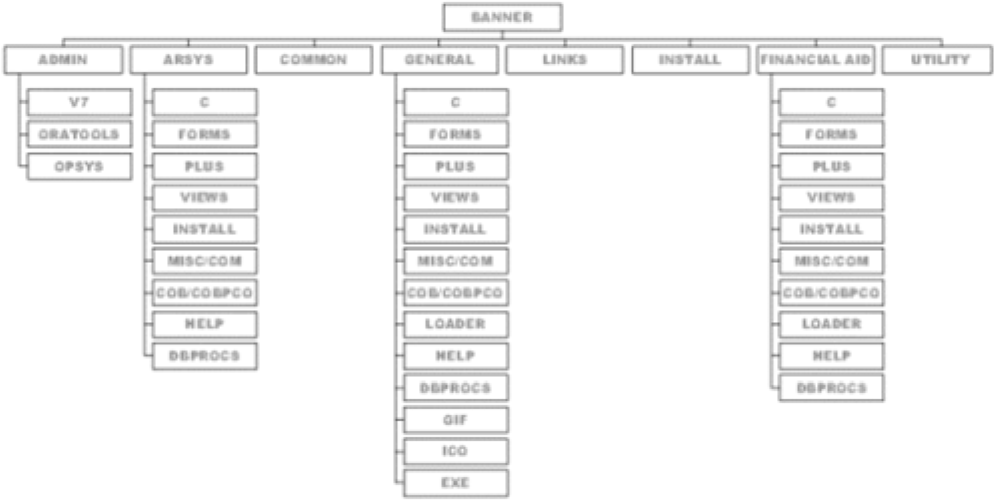
Banner Products

Diagram



Financial Aid Directories

Diagram



Naming Conventions

Naming conventions

All Banner objects adhere to naming conventions. The names of all Banner forms (except menu forms), reports, processes and tables are seven characters long, with each character representing a position location.

Objects include forms, tables, processes, etc.

Example:

RBRBCMP (Budget Component Rule Form)

- R -> Financial Aid
- B -> Budgeting
- R -> Rule
- BCMP -> (acronym)

Position 1

Identifies the primary Banner product owning the form, report, process, or table. Some common examples follow:

A	Advancement	**R**	Financial Aid
F	Finance	**S**	Student
G	General	**T**	Accounts Receivable
N	Position Control	**V**	Voice Response
P	Human Resources/ Payroll/Personnel	**H**	New Products (Web)

Position 2

Identifies module that owns the form, report, process, or table. The letter assignments will vary by Banner System. The following table lists the letter assignments when position 1 = *R* (Financial Aid).

B	Budgeting	**J**	Student Employment	**R**	Requirements Tracking
C	Record Creation	**L**	Logging	**S**	Student System Shared Data
E	Electronic Data Exchange (EDE)	**N**	Need Analysis	**T**	Validation Form/ Table
F	Funds Management	**O**	Common Functions		
H	History and Transcripts	**P**	Packaging and Disbursement		

Position 3

Identifies the type of form, report, process, or table.

A	Application form	**P**	Process or Procedure
B	Base table/Batch COBOL process	**Q**	Query form
E	Either Batch or Online	**R**	Rule table, repeating table, or report/process
I	Inquiry form	**T**	General maintenance temporary table
O	Online COBOL process	**V**	Validation form/table or view

Positions 4 – 7

A unique four-character acronym identifying the form, report, process, or table.

Client-developed items

W, Y, or Z can be used as the first character for client-developed (non-Banner) applications built to coexist with Banner applications.

For client-developed reports or modules used within a Banner application, the SunGard Higher Education System identifier should be used as the first character (i.e., R = Financial Aid) and W, Y, or Z should be used as the second character.

For more information, refer to Chapter 1 of the General Technical Reference Manual.

Form Types

Menu

- Lists all related forms
- Outlines the System

Application/ Functional

- Allows one to enter, update, and query

Validation

- Lists all possible values for a given field
- Information may be entered

Rule/Control

- Defines how variables, objects, and applications can be used

Query

- Allows information to be queried and returned to another form

Inquiry

- Used only to look up information

Validation Forms

Description

Validation forms always have *TV* as their 2nd and 3rd characters.

All validation forms must have:

- a code
- a description
- an activity date.

Validation tables

Table and form names are the same.

Examples:

- Budget Type Validation table (RTVBTYP)
- Budget Group Validation table (RTVBGRP)
- Holds Validation table (RTVHOLD)

Query Forms

Description

The 3rd character of a query form is always a *Q.* Query forms can **only** be reached when called by another form.

Query forms cannot be used to change information; they are used strictly for lookups.

Examples

- Fund Determination Form (RFQDETC)
- Fund Query Form (RFQFUND)
- Disbursement Validation Query Form (ROQPVAL)
- Award Disbursement Query Form (RPQAWRD)

Inquiry Forms

Description

The 3rd character of an inquiry form is always an *I.*

Inquiry forms cannot be used to change information; they are used strictly for lookups. They may be accessed from any menu.

Examples

- Applicant Need Analysis Application Inquiry Form (RNIAPPL)
- Applicant Data Log Inquiry Form (ROIALOG)
- Aid Year Inquiry Form (ROIAIDY)
- Budget Inquiry Form (RBIBUDG)

Database Dictionary

Description

The database dictionary is a set of tables and views used as a read-only reference about the database.

It is one of the most important parts of an Oracle database, because it contains:

- Names of schema objects (tables, views…)
- Names of Oracle users
- Privileges and roles granted to each user

Maintenance

The database dictionary is maintained internally.

- System table space
- Select only

In SQL*Plus, describe dictionary (or *desc dict*).

Ownership

SYS owns all base dictionary tables.

- Users have access to views of the data dictionary
- No user should alter any object contained in the SYS schema

During normal operation, Oracle verifies the existence of objects and checks whether users have proper access to them.

catalog.sql

- Creates the database dictionary and public synonyms
- Grants public access to the synonyms

Retrieval

The database dictionary is divided by users. To retrieve basic information, use:

- ALL: Lists all objects that a user has been granted access to
- USER: Lists all objects that a user owns
- DBA: Lists all objects in the database
- ALL_TABLES: Descriptions of tables
- ALL_COL_COMMENTS: Comments on columns of accessible tables
- ALL_TAB_COLUMNS: Lists columns contained in all tables
- ALL_TAB_COMMENTS: Comments on tables
- ALL_USERS: Information on all users in database
- ALL_VIEWS: Lists views accessible to user

Table and module ownership

For more complete information, refer to Oracle Server Reference or select *table_name* from *dict* in SQL*Plus.

Product	**Table owner in Oracle**	**Module owner in Banner**
Accounts Receivable	TAISMGR	TAISUSR
Advancement	ALUMNI	ADISUSR
General	GENERAL	GENERAL
Finance	FIMSMGR	FIMSUSR
Payroll	PAYROLL	HRISUSR
Position Control	COSNCTL	HRISUSR
Financial Aid (Resources)	FAISMGR	FAISUSR
Student	SATURN	SAISUSR

Referential Integrity

Definition

Referential integrity is a condition by which a dependent table's foreign key must have either a null entry or a matching entry in a related table.

Oracle enforces relationships between tables through the use of primary and foreign key constraints.

Banner uses naming conventions for these objects.

Note: Keys are implemented as constraints.

Primary key

A candidate key, selected as a unique entity identifier, which enforces unique, non-null keys.

- PK_ppppppp is the Banner convention for primary keys, where
  - **PK** stands for Primary key
  - **ppppppp** is the Primary key table name

Primary key example

Example: PK_STVTERM

- Alter Term Code Validation table (STVTERM)
- Add constraint PK_STVTERM
- Primary key (column_name)

Foreign key

An attribute in one table whose values must match the primary key in another table or whose values must be null.

A table can have any number of foreign keys.

Usage:

- Referencing validation tables
- Application hierarchy

Foreign key - validation

Validation tables:

- FK + n + "_" + fffffff + "_INV_" + ppppppp + "_code" is the Banner convention for Foreign keys, where:
  - **FK** stands for Foreign key
  - **n** is a one-up number which distinguishes potential duplicate Foreign key names in a given table
  - **fffffff** is the Foreign key table name
  - **ppppppp** is the Primary key table name

Example: `FK3_RPRATRM_INV_STVTERM_CODE`

- Alter Applicant Award by Term table (RPRATRM)
- Add constraint FK3_RPRATRM_INV_STVTERM_CODE
- Foreign key (sfrstcr_code,)
- References STVTERM (STVTERM_CODE)

Foreign key - hierarchy

Application hierarchy:

```
"FK"+1up#+"_"+foreign_table+"_INV_"+ Primary_table_"KEY"
FK3_RPRATRM_INV_RPRAWRD_KEY
```

STVTERM

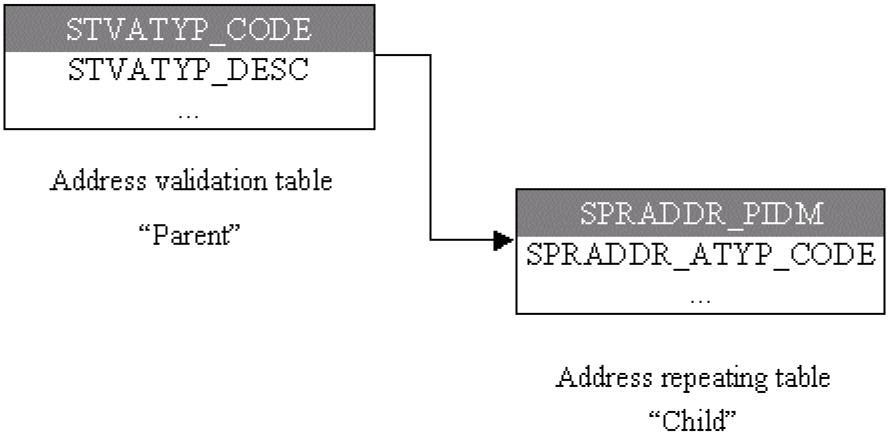
Before a term code may be entered into RPRATRM, it must exist in STVTERM.

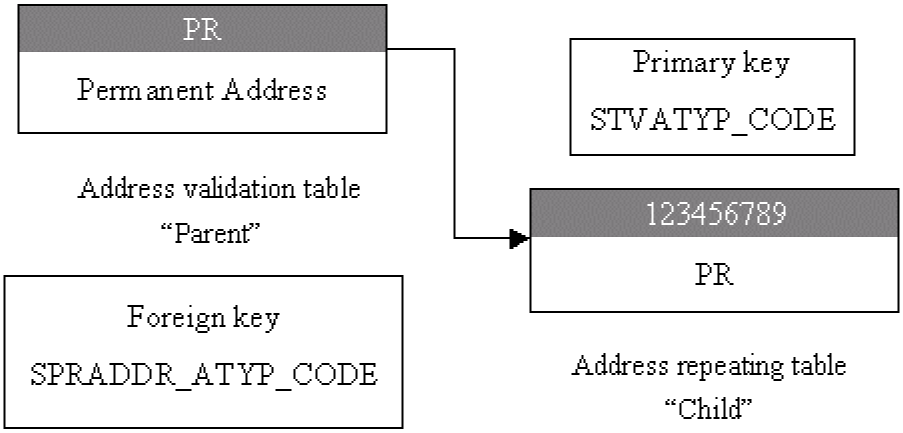
STVTERM codes can not be deleted if they exist in other tables.

Parent and child rows

The Child row has the foreign key constraint and the Parent row has the primary key constraint.

The Parent row cannot be deleted when a Child row exists.





Data Standardization

Components

- IDs
- Names
- Addresses
- Dates

Identification Numbers

Persons

Enter an identification number of your choice, or use the identification number that the System generates on any identification form (xPAIDEN).

Non-persons

Use the System-generated identification number.

Names

Basic format

Prefix	First Name	Last Name	Suffix
Mr.	William	Smith-Jones	III

Case

Use conventional upper- and lower-case letters.

Last name considerations

Do not enter spaces within prefixed last names:

- MacArthur O'Connor VanHusen
- St.John deBolt DuShen

Do not enter spaces within hyphenated last names:

- Smith-Jones Cochram-Ashley

First initials/ middle name

For names using a first initial and middle name, enter a period and one space between the initial and the name. (Example: R. Maureen Smith)

Prefixes/Suffixes

Use periods after prefixes and suffixes where applicable:

- Miss Mrs. Mr. Jr.

Addresses

Pound sign (#)

Do not use a pound sign (#) when completing an **Address** field if your organization uses Banner Letter Generation. (Banner Letter Generation recognizes a pound sign as a formatting command.)

Code definitions

County codes are defined on the County Code Validation Form (STVCNTY).

State and province codes are defined on the State/Province Code Validation Form (STVSTAT). (State and Province names are entered into the **State/Prov** fields.)

Dates

Date format selection

You may select the date format via the **Date Format** radio button on the Installation Control Form (GUAINST).

Date formats (MDY, DMY, YMD)

- MDY November 5, 1996 is entered as 11/05/99
- DMY November 5, 1996 is entered as 05/11/99
- YMD November 5, 1996 is entered as 99/11/05

Notes

- If you enter only part of the date, the rest of the current date defaults
- If you are including a date in query criteria, always include the century
- You can enter a dash (-) instead of a slash (/)
- For Job Submission dates, use *DD-MON-YYYY* format on the Process Submission Control Form (GJAPCTL)

PIDMs

What is a PIDM?

A "Primary Identification Master," or PIDM, uniquely identifies the records of one person or non-person entity in Banner. It is an internal key field used to identify and store records, and is assigned automatically when new records are created.

A PIDM is generated from an Oracle Sequence called PIDM_SEQUENCE through a function named GB_COMMON.F_GENERATE_PIDM.

How to get the next available PIDM

```
Select pidm_sequence.curval from DUAL;
Select pidm_sequence.nexval from DUAL;
```

Database Tool Set for SQL

Introduction

This chapter lists common SQL scripts used in Banner Financial Aid.

Objectives

At the conclusion of this chapter, participants will be able to view common SQL scripts.

SQL*Plus Scripts

Scripts

The following scripts are used to examine the database structure using SQL*Plus.

check_col_ comments.sql

```
select column_name,comments
  from all_col_comments
 where table_name like upper('&Table_Name%')
 order by column_name
```

check_tab_ comments.sql

```
select table_name,comments
  from all_tab_comments
 where table_name like upper('&Table_Name%')
   and table_name not like '%$%'
 order by table_name
```

check_cons.sql

```
select constraint_name,column_name,
       position,table_name
  from all_cons_columns
 where table_name like upper('&TableName%')
                 order by table_name, position
```

check_cons_ all.sql

```
select all_cons_columns.constraint_name,
        all_cons_columns.column_name,
        all_cons_columns.position,
           decode  (all_constraints.constraint_type,'R',
       'Referencial Integrity', 'C','Not
        Null/Check','P','Primary Key',
       'U','Unique','V','With Check Option',
       'O','With Read Only') ctype,
       all_constraints.status,
       all_cons_columns.table_name
  from all_cons_columns, all_constraints
 where all_cons_columns.column_name
       like upper('&TableName%')
   and all_cons_columns.constraint_name =
       all_constraints.constraint_name
   and all_cons_columns.table_name =
       all_constraints.table_name
 order by table_name, all_constraints.constraint_name,
       position
```

self-generating.sql

```
select 'select distinct '||column_name||' from '
        ||table_name||';'
  from dba_tab_columns
 where table_name like upper('&tables%')
   and column_name like upper('%&column%')
```

Self Check

Directions

Use the information you have learned in this workbook to complete this self check activity.

Exercise 1: Data Dictionary

Write a simple script to use as a tool to list information about Banner Financial Aid tables, to include table_name, table_comment. The script should include ability to ask user for name of table and should accept partial input.

Save your script to a temporary directory on your computer for future use.

Exercise 2: Referential Integrity

Write a simple script to use as a tool to examine constraints placed on tables by Banner designers.

Save your work.

Exercise 3: Referential Integrity Challenge

Write a script that will show constraint name, column name, position, type of constraint, constraint status and table name.

Exercise 4: General Person

Write a simple query to get ID, first name, middle name, last name, and the Change indicator for persons who have had changes made to their ID on the Identification Form (SPAIDEN) records.

Exercise 5

Write a simple query to get PIDM, ID, first name, middle initial, last name, and Social Security number for persons who have had changes made to their name records.

Self Check – Answer Key

Exercise 1: Data Dictionary

Write a simple script to use as a tool to list information about Banner Financial Aid tables, to include table_name, table_comment. The script should include ability to ask user for name of table and should accept partial input.

Save your script to a temporary directory on your computer for future use.

```
select table_name, comments
  from all_tab_comments
 where table_name like ('%BASE')
   order by table_name
```

Exercise 2: Referential Integrity

Write a simple script to use as a tool to examine constraints placed on tables by Banner designers.

Save your work.

```
select constraint_name, column_name,
       position, table_name
  from all_cons_columns
 where column_name like upper('&TableName%')
 order by table_name, position
```

Exercise 3: Referential Integrity Challenge

Write a script that will show constraint name, column name, position, type of constraint, constraint status and table name.

```
select all_cons_columns.constraint_name,
       all_cons_columns.column_name,
       all_cons_columns.position,
decode (all_constraints.constraint_type,'R','Referencial
       Integrity', 'C','Not Null/Check','P','Primary
       Key','U','Unique','V','With Check Option','O',
       'With Read Only') ctype, all_constraints.status,
       all_cons_columns.table_name
  from all_cons_columns, all_constraints
 where all_cons_columns.column_name
       like upper('&TableName%')
   and all_cons_columns.constraint_name =
       all_constraints.constraint_name
   and all_cons_columns.table_name =
       all_constraints.table_name
 order by table_name, all_constraints.constraint_name,
       position
```

Exercise 4: General Person

Write a simple query to get ID, first name, middle name, last name, and the Change indicator for persons who have had changes made to their ID on the Identification Form (SPAIDEN) records.

```
SELECT  spriden_id,
        spriden_first_name,
        spriden_mi,
        spriden_last_name,
          spriden_change_ind
  FROM    spriden
 WHERE  spriden_change_ind = 'I'
   AND  spriden_entity_ind = 'P'
```

Exercise 5

Write a simple query to get PIDM, ID, first name, middle initial, last name, and Social Security number for persons who have had changes made to their name records.

```
select spriden_pidm,
       spriden_id,
       spriden_first_name,
       spriden_mi,
       spriden_last_name,
       spbpers_ssn,
       spriden_change_ind
  from spriden, spbpers
 where spriden_change_ind = 'N'
   and spriden_entity_ind = 'P'
   and spriden_pidm = spbpers_pidm
```

Banner 8.0 Common Enhancements

Introduction

This section provides an overview of the common enhancements in Banner General 8.0, which will also affect Banner Financial Aid 8.0.

Objectives

At the conclusion of this chapter, participants will be able to:

- identify enhancements to PIN security
- identify enhancements related to supplemental data
- identify enhancements made for Internationalization purposes
- identify enhancements made for partial data logging.

PIN Maintenance

Enhancements

Banner General 8.0 includes enhancements for user PIN (password) security.

- PINs are now stored only in encrypted form
- Institutions can set and enforce standards for strong passwords
- A new, more secure PIN reset mechanism has been established

New forms

- GOAQSTN – PIN Questions Form

Changed forms

- GUAPPFR – Enterprise PIN Preferences Form
- GOATPAC – Third Party Access Form
- GOATPAD – Third Party Access Audit Form

Please refer to the *Banner General 8.0 Release Guide* for more information on these changes.

Internationalization

Enhancements

Banner General 8.0 includes enhancements designed to support the use of Banner in different countries and in different languages. These Internationalization enhancements will reduce the amount of custom modifications necessary to make Banner usable under those circumstances.

Unicode Support

Banner now supports the Unicode international character set through the character standard UTF8. The Oracle database will be converted to UTF8 as part of the Banner 8.0 installation process.

Additional IDs

The Additional Identification Table (GORADID) allows storage of unlimited extra IDs for a person in Banner. Each Additional ID must be assigned an ID type, which is set up on the Additional Identification Type Validation (GTVADID) form and table.

A new **Additional ID** table on the General Person Identification Form (SPAIDEN) displays Additional ID information.

Expanded fields

Many fields, including the following, have been expanded on Banner tables and forms so as to accommodate longer data values.

- Name
- Address
- Telephone number
- E-mail address
- ID
- Currency amount
- Currency rate

Please refer to the *Banner General 8.0 Release Guide* for more information on these changes.

Supplemental Data Engine

Enhancements

The Supplemental Data Engine allows storage of additional data that are not part of the existing Banner data model. Examples of the types of data affected are comments fields to record miscellaneous notes or data that has been translated into various languages.

No customization of Banner forms or tables is needed to capture and use additional data with SDE. The new data is displayed in a popup window, the Supplemental Data Window, and is stored in a supplemental data table. Because no customization is needed, supplemental data is generally not impacted by Banner upgrades.

Each supplemental data record created through SDE is tied to a specific Banner table, so any forms using that table will be able to access the same data through the Supplemental Data Window. Through SDE you can create additional fields associated with a specific Banner table but stored in a separate table, the Supplemental Data Table.

Although it is natural to think of supplemental data in terms of extra fields on Banner forms, SDE is tied to forms only indirectly.

There are several limitations of SDE, including the following.

- Not all tables, blocks, and forms work with SDE.
- The combination of SDE with Virtual Private Database (VPD) for use in Multi-Enterprise Processing is currently not supported.
- Masking is not currently supported with SDE.

Please refer to the *Banner General 8.0 Release Guide* for more information on these changes.

Supplemental User ID/Activity Date and Logging

Enhancements

Banner Financial Aid user-defined table logging functionality was first introduced with the Early Decision Release in 2003. This logging support allows you to choose which Banner Financial Aid tables and columns to log, and which actions to log, through the Logging Control Form (ROALOGC).

Specifically, on ROALOGC you can activate logging on the table by clicking the **Active?** check box to the right of the table description. You can choose to Log Inserts; Log Updates; Log Deletes; or Activate All Columns of the desired table. You may choose any combination of these options.

If you prefer to log specific columns, rather than all columns, navigate to the second window of this form. Here, you can place a check mark next to the columns you want to log under the Log heading. You may choose any combination of columns to log.

Logging results appear on the Logging Activity Inquiry Form (ROILOGA), which opens in Query mode. Use the first window to query on any field except the **Name** field. Fields 1 – 5 display the data in the row of the table in which the data was changed. The second window displays the exact column name that was changed, what it was prior to the change, and what it was changed to. (This includes inserts and deletions assuming you set it up this way on ROALOGC).

Tables

Tables with user-defined logging prior to this release are:

- RBBABUD
- RBRACMP
- RORSTAT
- RPRADSB
- RPRATRM
- RPRAWRD
- RRRAREQ

This enhancement adds user-defined table logging functionality to the Banner Financial Aid Budget, Student Employment, and Short Term Credit modules. The additional tables are:

Budgeting Tables

- RBRMIXD
- RBRCOMP
- RBRGTYP
- RBRGMSG
- RTVBGRP
- RTVBTYP
- RTVCOMP

Short Term Credit Tables

- RPBSTAB
- RPRASTC
- RPRASTD
- RPRSTAC
- RTVCINT
- RTVRPSR

Student Employment Tables

- RJBJOBT
- RJBPLBD
- RJBSEDR
- RJRPAYL
- RJRPLRL
- RJRSEAR
- RJRSERF
- RJRSEWH
- RTVAUST
- RTVRFST

In order to implement the addition of user-defined logging to these tables, the user ID and activity date were added or modified to utilize database triggers for the update of these values.

Please refer to the *Banner General 8.0 Release Guide* for more information on these changes.

Partial Data Masking

Enhancements

The ability to partially mask a field, which was introduced initially in Banner 7.0, has been extended in Banner 8.0 to character fields. You can allow a specified number of characters at one side of a field to remain readable while masking the remainder of a value.

To support partial character masking, two new fields (**Partial Character Mask** and **Partial Unmasked Length**) have been added to the Data Display Mask Rules Form (GOTDMSK).

Please refer to the *Banner General 8.0 Release Guide* for more information on these changes.

Review of Financial Aid Process

Introduction

This chapter provides a broad overview of the Banner Financial Aid Process and its components.

Objectives

At the conclusion of this chapter, participants will be able to

- view the overall Banner Financial Aid process flow
- view and identify components of a wide variety of Financial Aid components, including Applicant Record Creation, Data Load, Need Analysis, Requirement Tracking, Shared Data, Electronic Data Exchange, Funds Management, Budgeting, Packaging and Disbursement, Award History and Transcripts
- identify various processes that are part of the overall Financial Aid process.

Financial Aid Process

Student applies for aid

The student will fill out and submit:

- Free Application for Federal Student Aid (FAFSA).
    - Student competes for Financial Aid on an ANNUAL Basis
    - Output from FAFSA is the Student Aid Report (SAR)
    - Student receives SAR in the mail
    - Institution receives SAR electronically (ISIR)
    - Expected Family Contribution (EFC)
- Institutional application
- Application for admission to the institution

Financial Aid reviews paperwork

- File Processing
- Verification
- Request additional documentation
- Tracking Letter
- Documents required for packaging or disbursement
- Paperwork completed – 'File Complete'
- SAP review

Packaging – awarding of aid

- Federal Pell Grant determined by EFC and budget
- Campus-based aid (SEOG, Perkins Loan, and Federal Work Study) based on institutional packaging guidelines
- Scholarships – merit and need
- Loans – subsidized, unsubsidized and Plus

Types of Financial Aid

Aid types

- Scholarships
- Work-Study
- Loans (Subsidized, Unsubsidized, Parent, Perkins)
- Grants
  - Federal Pell Grant
  - Federal Supplemental Opportunity Grant (SEOG)
  - State Grants
  - Institutional Grants

Acceptance and Application of Aid

Award letter

An award letter is mailed to a student, who has the option to accept or decline the awarded aid. When the student's acceptance of the aid is entered into the system, the aid is then ready for memo, authorization or payment.

Fee assessment and billing

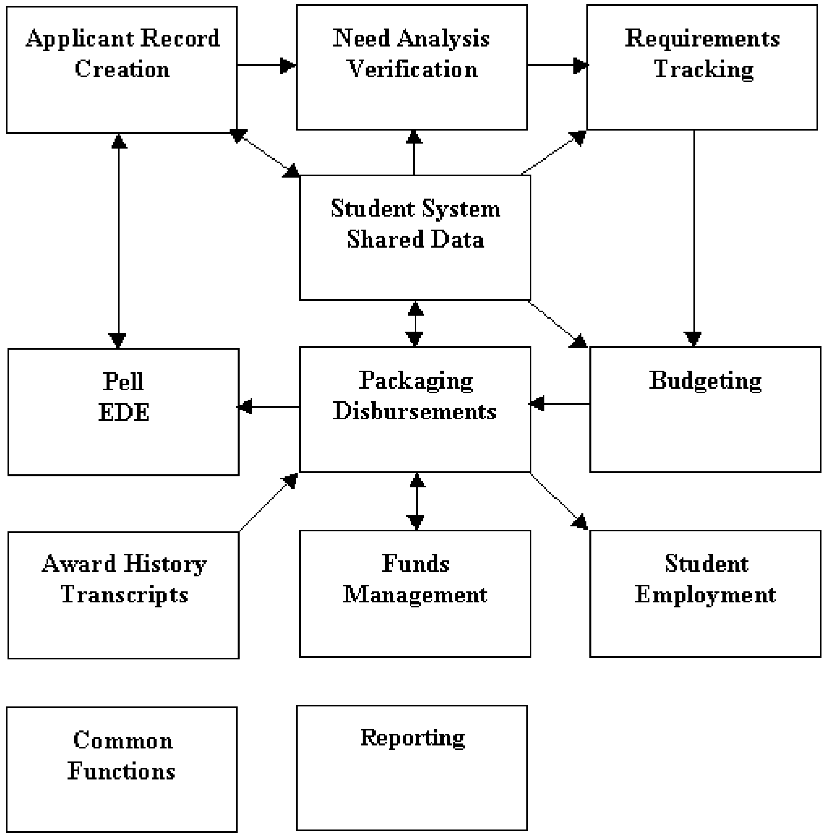
Students register for and are subsequently billed for courses. Aid is then applied to student accounts.

Reporting

- Federal Pell Grant Report
- FISAP (Fiscal Operations Report)
- State Grant Reporting

Financial Aid System Process Flow

Diagram



1

[1] Financial Aid TRM Supplement, pages 1-2 and 1-3 (12 of 156).

Applicant Record Creation

Applicant record creation

Financial Aid applicant records can be created through several processes, including MDE tape loads, Pell Electronic Data Exchange, and manual creation using Record Creation/Need Analysis forms such as RRAAREQ, RNANAxx and RPAIDEN.

Applicant record creation forms

Seven-character name	Proper name of form, process, report, or table
ROASTAT	Applicant Status Form
ROARMAN	Financial Aid Record Maintenance
RNANAxx	Need Analysis Form
RNAPRxx	Need Analysis Processing Form
RNAVRxx	Need Analysis Document Verification Form
RNARSLT	Need Analysis Results Form
RRAAREQ	Applicant Requirements Form
RBAABUD	Applicant Budget Form
RPAAWRD	Award Form
RSIHIST	Summary Academic History Inquiry Form
RSIENRL	Financial Aid Enrollment Inquiry Form
RSISTDN	Basic Student Information Inquiry Form
RSIAPPL	Admissions Application Inquiry Form
RSIAREV	Account Detail Review Inquiry Form
RSIDEGR	Degrees and Other Formal Awards Inquiry Form
RSIHOUS	Housing Information Inquiry Form
RSIPCOL	Prior College Inquiry Form
RSIHSCH	High School Inquiry Form
RSITEST	Test Score Inquiry Form
RSIVETN	Veteran Certification Inquiry Form
RSIINTL	International Student Inquiry Form
RSICONT	Third Party Contract Inquiry Form
RSIEXPT	Billing Exemption Inquiry Form

Applicant Record Creation Data Load Procedures

Procedures

- Load electronic media from external sources
- Stored in temporary ORACLE tables
  - By processes RCBCTxx (CSS) or RCBTPxx (EDE)
  - Used by the RCPMTCH process
- RCRTPxx process completes load

For more information on these procedures, refer to Chapter 2 of the Banner Financial Aid User Manual.

Prepare for the Data Load Process

RCRINFR

Set up the Data Source Code Validation Form (RCRINFR):

- This form is aid-year-specific, and most values are delivered already populated.

- To set up this form, find the source code(s) used at your institution. Define the rest of the columns for that source. This includes setting a requirements code (tracking requirement) that can be updated when the data is loaded, which is a supplemental requirement that can be used for CSS. The related tracking requirement status is also defined.

- The **Contact Type** and **Recruiting Source** fields must be populated for the Data Load process to run to completion. Even though you may not be creating recruitment records, this data is required for Part 3 of the Data Load process (RCRTPxx) to complete successfully.

RCRTPTR

Build the Interface Data Translation Rules Form (RCRTPTR)

- Determine the conversion values on the Interface Data Translation Rules Form (RCRTPTR).

- The converted data is data that is contained on the incoming data file.

- The fields that you must define are **Major Code, Marital Status** and **SBGI Code**. These fields may contain data that the institution uses and this is the opportunity to make the incoming data match the institutionally defined validation tables for these fields.

ROAINST

Build the Institutional Financial Aid Options Form (ROAINST)

- The following fields must all be defined prior to running the data load for the year: **Current Term Code**, **Start Year**, **End Year**, **EFC Proration**, **Default Aid Period**, **Primary MDE**, **SBGI Conversion**, **Initial SAP Code**, **Update Trans#/EFC**, **Applicant Received Date Indicator**, and the **MDE** codes, if applicable.

RCRDTSR

- Establish the matching source and parameter set for the aid year and data source used for the processing of individual records from the Financial Aid Suspended Record Maintenance Form (RCRSUSP).

Destination directory

Regardless of what source your data is being received from and in which format, the data must be loaded to the $DATA_HOME/finaid directory for UNIX or DATA$HOME for VMS. (Ensure that you have write privileges to the appropriate directory).

Filenames

The disk file name (data load file) must be renamed to match what the processes are expecting.

Rename the file to:

- xxyyesar.tap (EDE ISIR diskette or download)
- xxyyesat.tap (EDE ISIR tape)
- xxyycssd.tap (CSS diskette or download)
- xxyycsst.tap (CSS tape)

This step must be completed every time there is a new input file to be loaded to the system.

Run the Data Load Process

Data Load steps

- Run RCBTPxx (Part 1 of Data Load for federal data) or
- Run RCBCTxx (Part 1 of Data Load for CSS PROFILE data only)
- Run RCPMTCH (Part 2 of Data Load)
- Run RCRTPxx (Part 3 of Data Load)

  It is important to run the RCPMTCH and RCRTPxx steps as close together as possible. This decreases the chances of someone adding new students online, which would make the New status (as determined by RCPMTCH) incorrect.
- Run the RCPDTMP Process

Need Analysis/Verification Forms and Tables

Forms

The Need Analysis process performs calculations that determine the contribution an applicant can make toward his/her expenses. The Verification process validates certain information provided on an aid application against the same data from other documents.

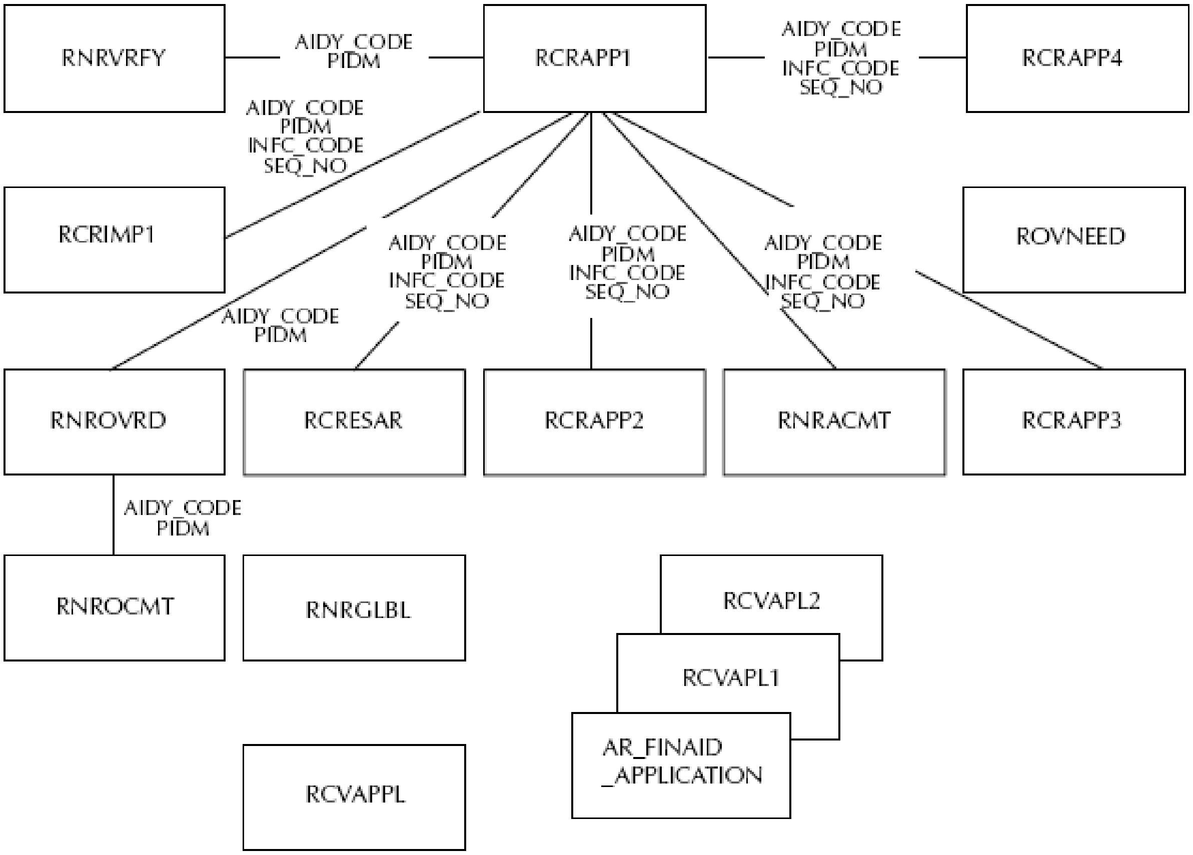
Seven-character name	Proper name of form, process, report, or table
RNANAxx	[Year-specific] Need Analysis Form
RNAPRxx	[Year-specific] Need Analysis Processing Form
RNASUxx	[Year-specific] Need Analysis Supplemental Form
RNARSLT	Need Analysis Result Form
RNAOVxx	[Year-specific] Applicant Override Form
RNIAPPL	Applicant Need Analysis Application Inquiry Form
RNINSLD	Applicant Student Loan Data Inquiry Form
RNINAIQ	Calculated Need Analysis Detail Inquiry Form
RNIMSxx	[Year-specific] Miscellaneous Results Inquiry Form
RNRGLxx	INAS Global Policy Options Rules Form
ROASTAT	Applicant Status Form
ROAPELL	Applicant Pell Grant Form
RNAVRxx	[Year-specific] Need Analysis Document Verification Form
RNAPURG	Application Purge Form
RNASLxx	[Year-specific] Student Loan Data System Form

Tables

Seven-character name	Proper name of form, process, report, or table
RNRACMT	Financial Aid Application Comments
RNRGLBL	INAS Global Policy Options Table
RNROCMT	Financial Aid Application Override Comments
ROROVRD	Applicant Need Analysis Override Table
RNRVRFY	Needs Analysis Verification Table

Need Analysis/Verification Table Relationships

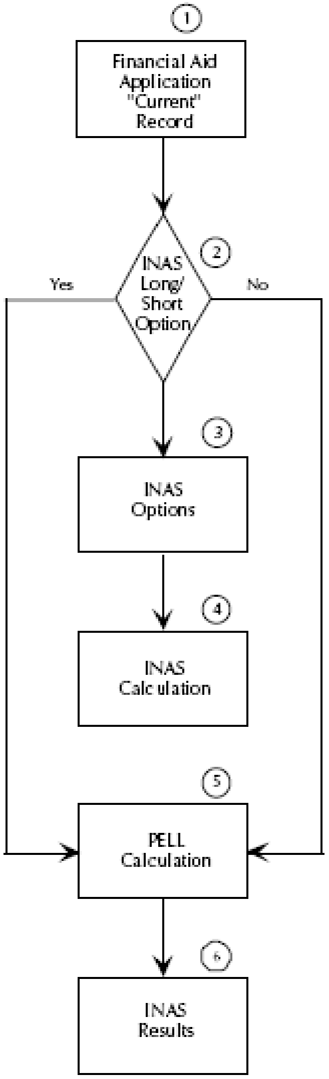
Diagram



[2] Banner Financial Aid TRM 7.1, page 1-7, January 2005.

Need Analysis Online Recalculation Process Flow

Diagram



[3] Banner Financial Aid User Guide 7.9, pg. 5-4.

Requirement Tracking Forms and Tables

Forms

The Requirements Tracking module monitors the request for and receipt of certain documents Financial Aid applicants are often required to provide, so that their applications can be fully processed.

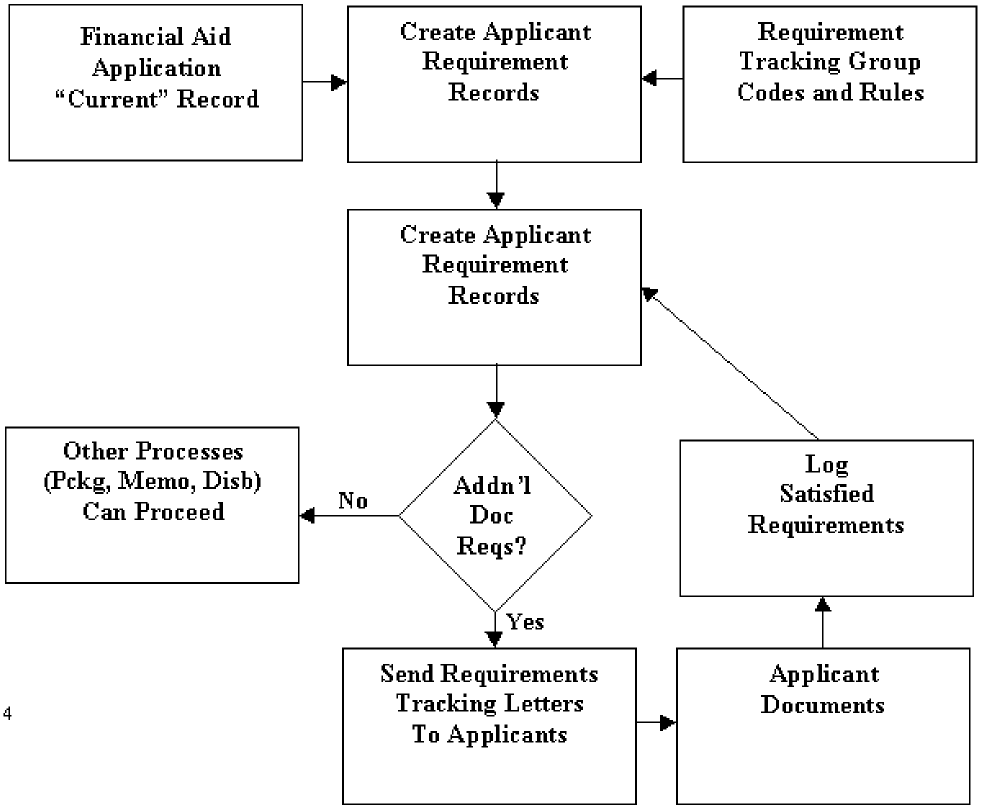
Seven-character name	**Proper name of form, process, report, or table**
RRAAREQ	Applicant Requirements Form
ROARMAN	Financial Aid Record Maintenance
RRAMASS	Applicant Requirements Mass Entry Form
RHACOMM	Applicant Comments Form
ROAIMMP	Applicant Immediate Process Form

Tables

Seven-character name	**Proper name of form, process, report, or table**
RRRGREQ	Tracking Group/ Requirements Rules
RRRAREQ	Applicant Requirements
RRRTMSG	Tracking Requirement Message Rules
RRVAREQ	Requirements View
RRVREQ1	Object: Access F/A Requirements View

Requirement Tracking Process Flow

Diagram



[4]

[4] Financial Aid User Manual/Requirements Tracking, page 7-4 (4 of 8).

Requirements Tracking Processes and Procedures

RTVTREQ

Requirements Tracking Validation Form (RTVTREQ)

- Define tracking requirement codes.
- Examples might include institutional applications, FAT, tax returns, verification statements, SARs, loan applications, etc.

RTVTRST

Requirements Tracking Status Validation Form (RTVTRST)

- Define tracking requirement status codes.
- Examples might include received, waived, established, satisfied, cancelled, or received incomplete.

RTVTGRP

Requirements Tracking Group Validation Form (RTVTGRP)

- Define tracking groups.
- Examples might include variations of Graduate vs. Undergraduate documents, Selected for Verification documents, Independent vs. Dependent verification groups, etc.

RRRGREQ

Requirements Tracking Group/Requirements Rules Form (RRRGREQ)

- Develop tracking group requirements for each group.
- Indicate which documents are required for each specific tracking group.
- You have the opportunity here to change your defaults for a specific code from the defaults you indicated on RTVTREQ for the code.

  These defaults will only be used if the system brings in the requirements for the specific student. If the requirement is entered manually and the manually entered code is not defined in the student's particular tracking group, the default from the RTVTREQ table will appear.

RORRULE

Financial Aid Selection Rules Form (RORRULE)

- Develop tracking group rules associated with each group.
- Assign criteria for placing applicants into tracking groups. Use the standard Banner data element dictionary table.
- Selection criteria may come from the Banner Student System, Financial Aid System, or any other Banner system which has been installed.

RRRTMSG

Requirements Tracking Code/Message Rules Form (RRRTMSG)

- Develop specific messages for each tracking code, if desired.

RORGRPS

The Automatic Group Assignment Process (RORGRPS) can be run in batch mode or in online mode. It allows you to process a single applicant (in online mode), to use a previously defined population selection, to process all applicants, or to process only applicants in *Wait* status. You can assign the applicant or applicants to budgeting groups, tracking groups, or packaging groups, or any combination of the three group types.

Note**:** When processing more than one group type, RORGRPS will process the group based on the order entered for the parameter. For example, if you want to process tracking, budgeting, packaging groups in that order, you would enter *TBP*. If you enter *BPT* for parameter 02, RORGRPS will process budgeting first, then packaging, and tracking group last.

RORRULE and ROOGSQL

ROOGSQL

The Online Group Assignment Selection Criteria Generation process (ROOGSQL) creates a SQL statement from the budgeting, tracking and packaging group selection rules created by the Financial Aid Selection Rules Form (RORRULE).

Student System Shared Data Forms

Shared data forms

The Banner Financial Aid System is designed to share information with the institution's Student Information System (the Banner Student System or another Student system).

Data is shared with the Banner Student System through a series of views that contain information that the Financial Aid system may require for certain processes. This information is available through a series of inquiry-only forms listed on the Student System Shared Data menu. Student System data is shared through these views, rather than via direct access of the Banner Student System tables.

Thus, in the event that Banner Financial Aid is installed as a stand-alone product in an environment in which Banner Student is not installed, these views can be created as tables by the institution and periodically populated with data from its Student System.

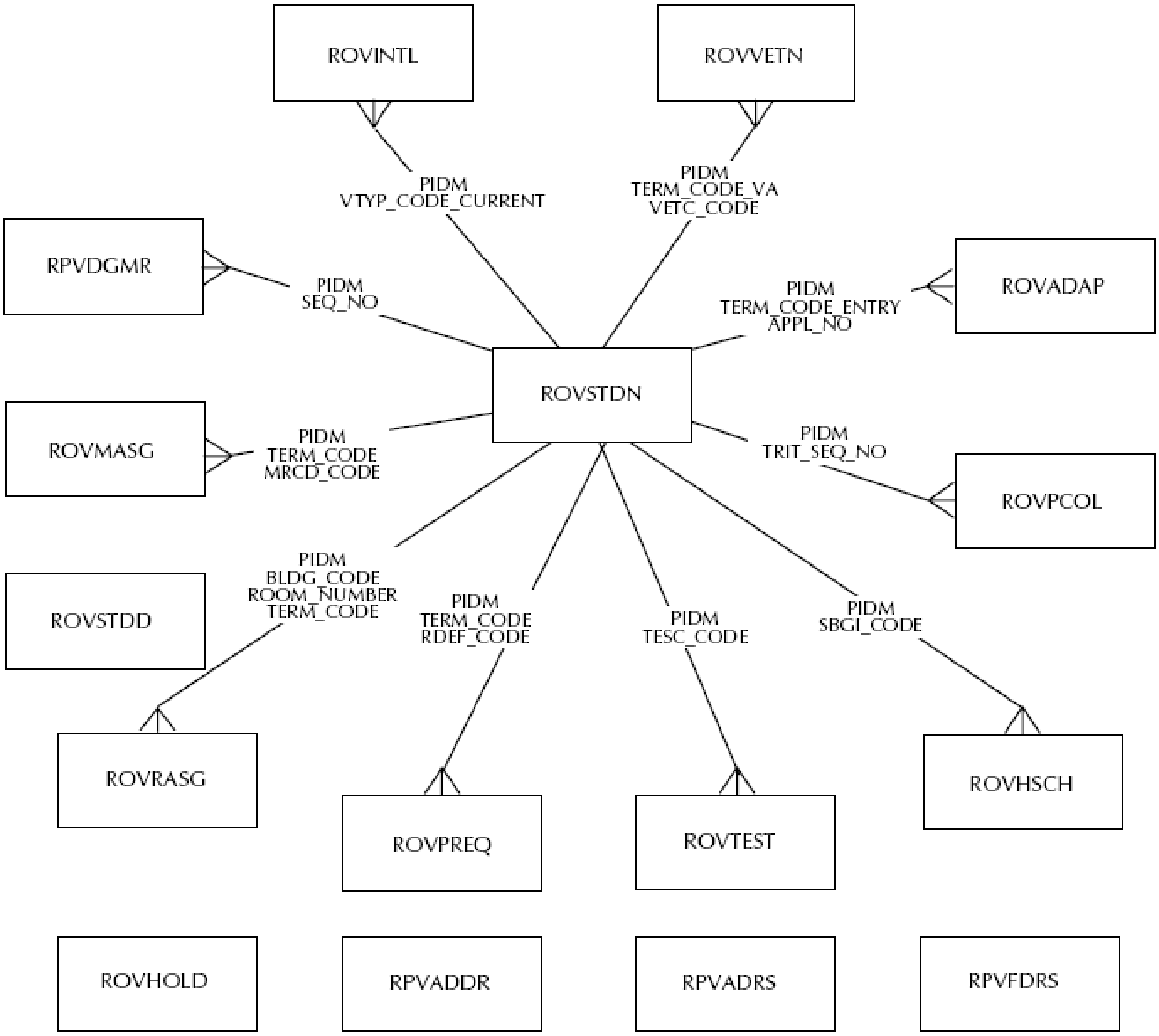
Seven-character name	**Proper name of form, process, report, or table**
RSIHIST	Summary Academic History Inquiry Form
RSIENRL	Financial Aid Enrollment Inquiry Form
RSISTDN	Basic Student Information Inquiry Form
RSIAPPL	Admissions Application Inquiry Form
RSIAREV	Account Detail Review Inquiry Form
RSIDEGR	Degrees and Other Formal Awards Inquiry Form
RSIHOUS	Housing Information Inquiry Form
RSIPCOL	Prior College Inquiry Form
RSIHSCH	High School Inquiry Form
RSITEST	Test Score Inquiry Form
RSIVETN	Veteran Certification Inquiry Form
RSIINTL	International Student Inquiry Form
RSICONT	Third Party Contract Inquiry Form
RSIEXPT	Billing Exemption Inquiry Form

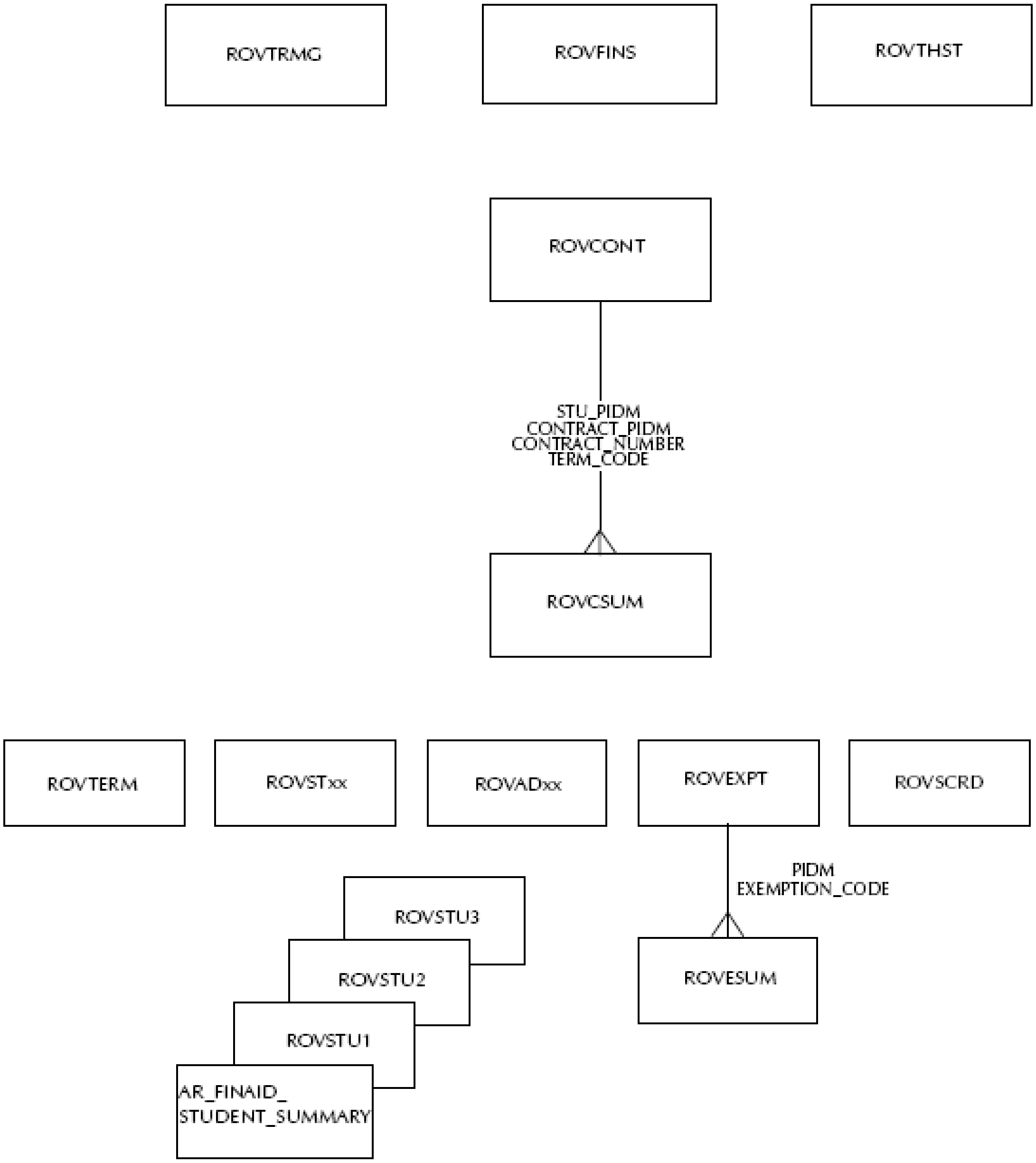
Student System Shared Data Tables and Views

Seven-character name	**Proper name of form, process, report, or table**
RORENRL	Enrollment Status Financial Aid Captured data
ROVSTDN	Basic Student Information
ROVADAP	Admissions selection
ROVADxx	Admission information
ROVDGMR	Degree information

Student System Shared Data Process Flow

Diagram



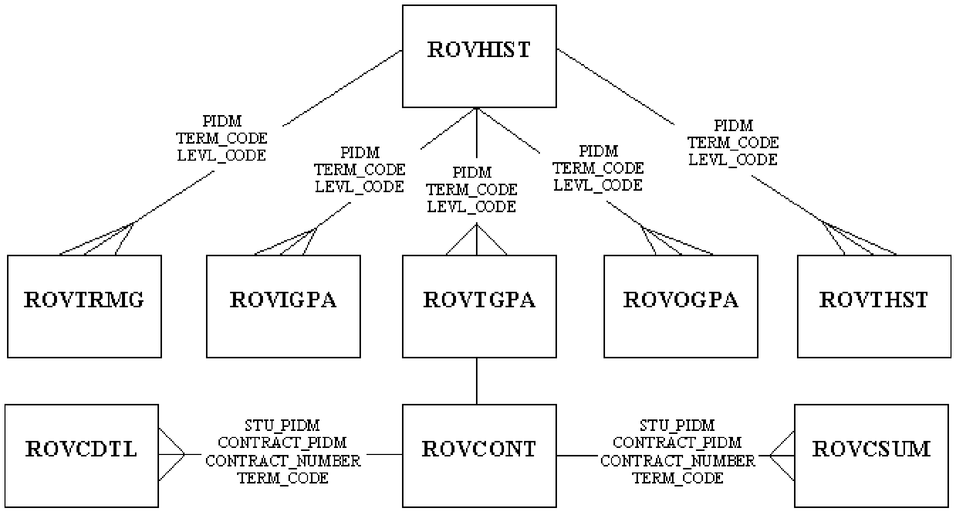


[5]

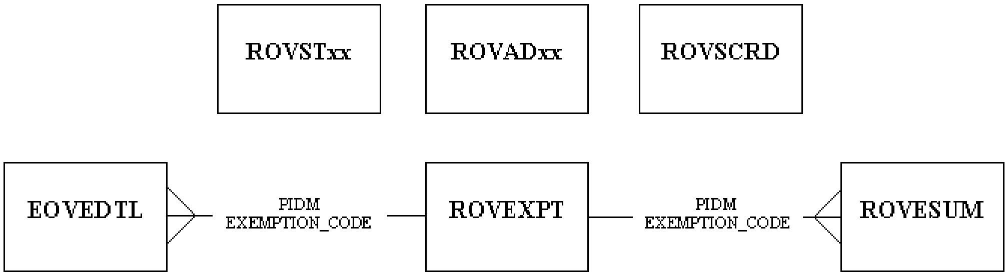
[5] Banner Financial Aid TRM Supplement 7.1, page 1-24/1-25.

Student System Process Flow

Diagram



[6]



[7]

[6] Financial Aid TRM Supplement, page 2-30 (84 of 156).

[7] Financial Aid TRM Supplement, page 2-30(84 of 156)

Student System Shared Data Processes and Procedures

RSRENRL

Financial Aid Enrollment Hours Process (RSRENRL)

- Add credit and billing hours from student enrollment to Financial Aid
- C process, used to 'capture' or 'freeze' hours

Electronic Data Exchange Forms

EDE

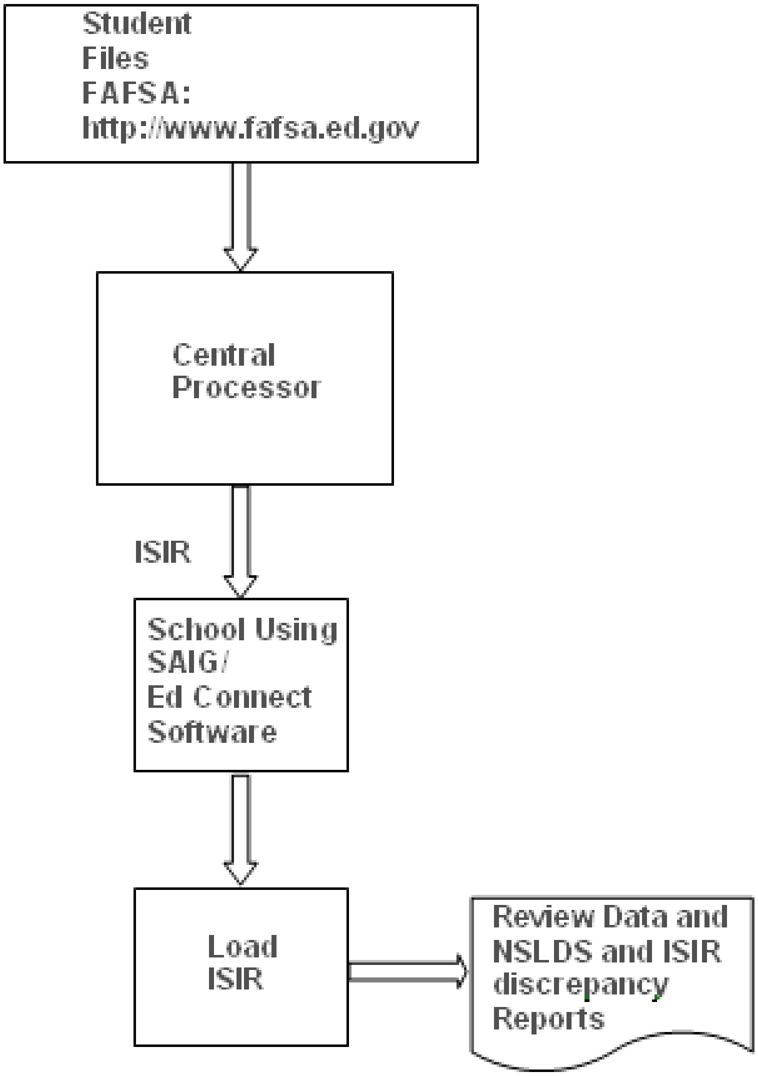
The EDE program allows institutions to share information electronically with the Central processor.

Seven-character name	Proper name of form, process, report, or table
REACORR	ISIR Correction/Request Form
RPIBATC	Batch Control Form
READRCT	EDE Directory Form
RNIMSxx	[Year-specific] Miscellaneous Results Inquiry Form
RTVRJCT	Rejection Code Validation Form
REAORxx	[Year-specific] Pell Origination/Acknowledgement Form
READIxx	[Year-specific] Pell Disbursement/Acknowledgement Form
REISSOA	Summary Statement of Account Inquiry Form
REIDSOA	Detail Statement of Account Inquiry Form
RERRDRQ	Data Request Record Form
REIRMRR	Multiple Reporting Record Inquiry Form
REIYTDS	Year to Date Pell Summary Inquiry Form

Seven-character name	Proper name of form, process, report, or table
RCRAPPxx RCRESAR	Applicant ISIR Tables
RPIBATC	Batch Control Form Table
REACORR	EDE Corrections Table
REICODD	XML COD Document Control Table
READIxx	YTD Disbursement Record Table
REAORxx	Pell Origination Table

ISAR Data Receipt Process Flow

Diagram

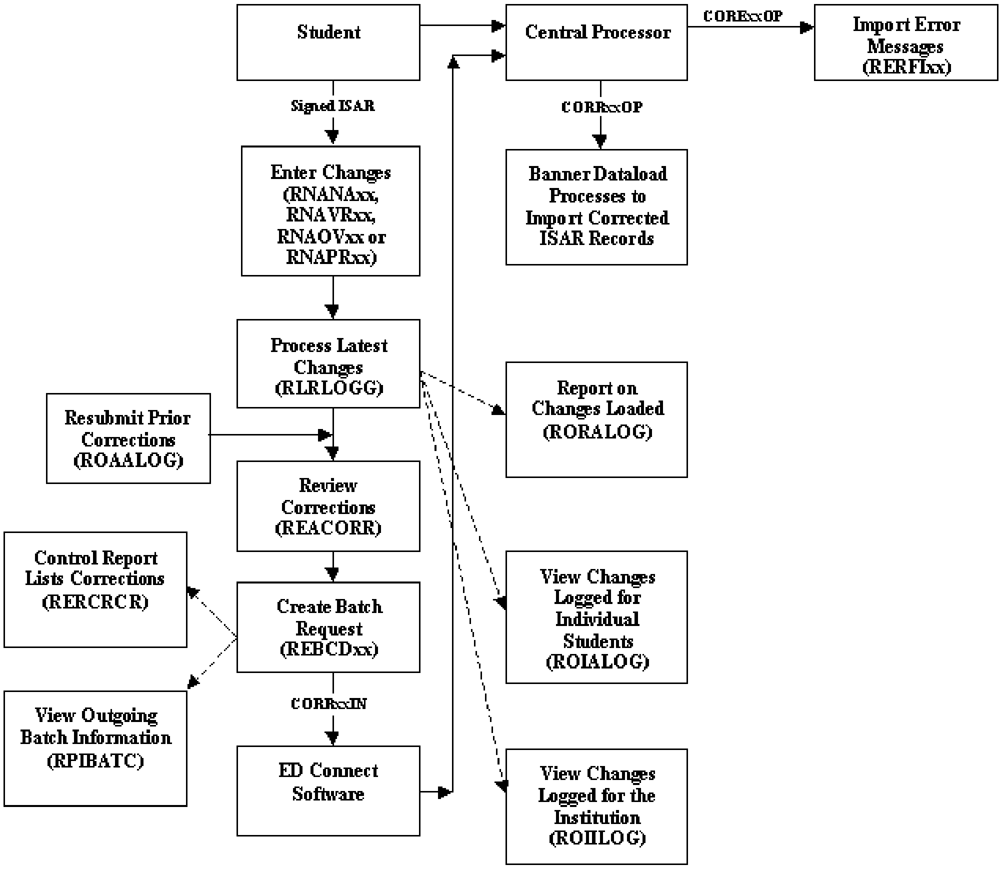


Note: Institutions no longer have to use the SAIG/EdConnect software. Clients can download ISIR files from their mailbox via the Internet. Files are saved to a local directory, ftp'd to the appropriate Banner directory, and then ISIRs are loaded to Banner.

Some schools do still use the EdConnect software, but it is in addition to loading the ISIR files to Banner.

EDE ISAR Data Correction Process Flow

Diagram


Student
Central Processor
COREcxOP
Import Error Messages (RERFIxx)
Signed ISAR
CORRxxOP
Enter Changes (RNANAxx, RNAVRxx, RNAOVxx or RNAPRxx)
Banner Dataload Processes to Import Corrected ISAR Records
Process Latest Changes (RLRLOGG)
Report on Changes Loaded (RORALOG)
Resubmit Prior Corrections (ROAALOG)
Review Corrections (REACORR)
Control Report Lists Corrections (RERCRCR)
Create Batch Request (REBCDxx)
View Changes Logged for Individual Students (ROIALOG)
View Outgoing Batch Information (RPIBATC)
CORRxxIN
ED Connect Software
View Changes Logged for the Institution (ROIILOG)

Budgeting Forms, Tables and Reports

Forms

Budgeting maintains data regarding an applicant's cost of attending an institution.

Seven-character name	**Proper name of form, process, report, or table**
RBAABUD	Applicant Budget Form
ROARMAN	Financial Aid Record Maintenance
RHACOMM	Applicant Comments Form
ROAIMMP	Applicant Immediate Process Form

Tables

Seven-character name	**Proper name of form, process, report, or table**
RBBABUD	Applicant Budget Table
RBRACMP	Applicant Budget Component Table

Processes, procedures and reports

There are no Budgeting processes or procedures.

There are, however, two reports:

- Applicant Budget Report (RBAABUD)
- Budget Component Report (RBRBCMP)

Funds Management Forms and Tables

Forms

Funds Management maintains data related to Financial Aid funding sources.

Seven-character name	Proper name of form, process, report, or table
RFRMGMT	Fund Management Form
ROAMGMT	Funds Management
RFIBUDG	Fund Budget Inquiry Form
RFIFFID	Federal Fund ID Inquiry Form
RPIFEDR	Federal Rules Inquiry Form
RFRDEFA	Default Award and Disbursement Schedule Rules Form
RFRASCH	Fund Award and Disbursement Schedule Rules Form
RORRULE	Financial Aid Selection Rules Form
RFRBASE	Fund Base Data Form
RTVFSRC	Fund Source Code Validation Form
RTVFTYP	Financial Aid Fund Type Validation Form

Major tables

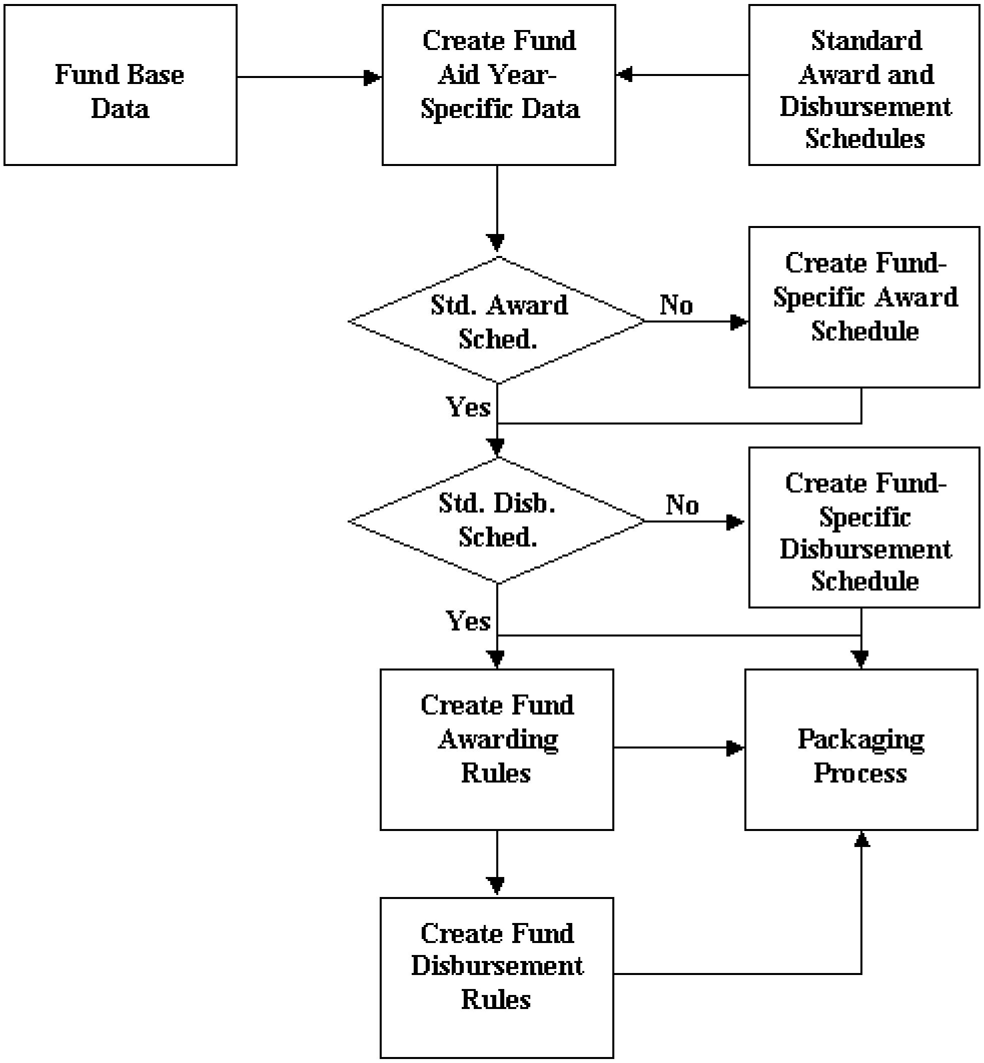
Seven-character name	Proper name of form, process, report, or table
RFRBASE	Fund Base Data Table
RFRBCMP	Fund/Budget Component Table
RFRFFID	Federal Fund ID Table

Processes and procedures

There are no processes in Funds Management other than procedures for the functional staff to follow. They may need assistance with RORRULEs.

Funds Management Process Flow

Diagram


Fund Base Data
Create Fund Aid Year-Specific Data
Standard Award and Disbursement Schedules
Std. Award Sched.
No
Create Fund-Specific Award Schedule
Yes
Std. Disb. Sched.
No
Create Fund-Specific Disbursement Schedule
Yes
Create Fund Awarding Rules
Packaging Process
Create Fund Disbursement Rules

Packaging and Disbursement Forms and Tables

Packaging and disbursement

The Packaging/Disbursements processes match an applicant's need with funds, determine an appropriate "package" for the applicant and record the disbursement of the funds.

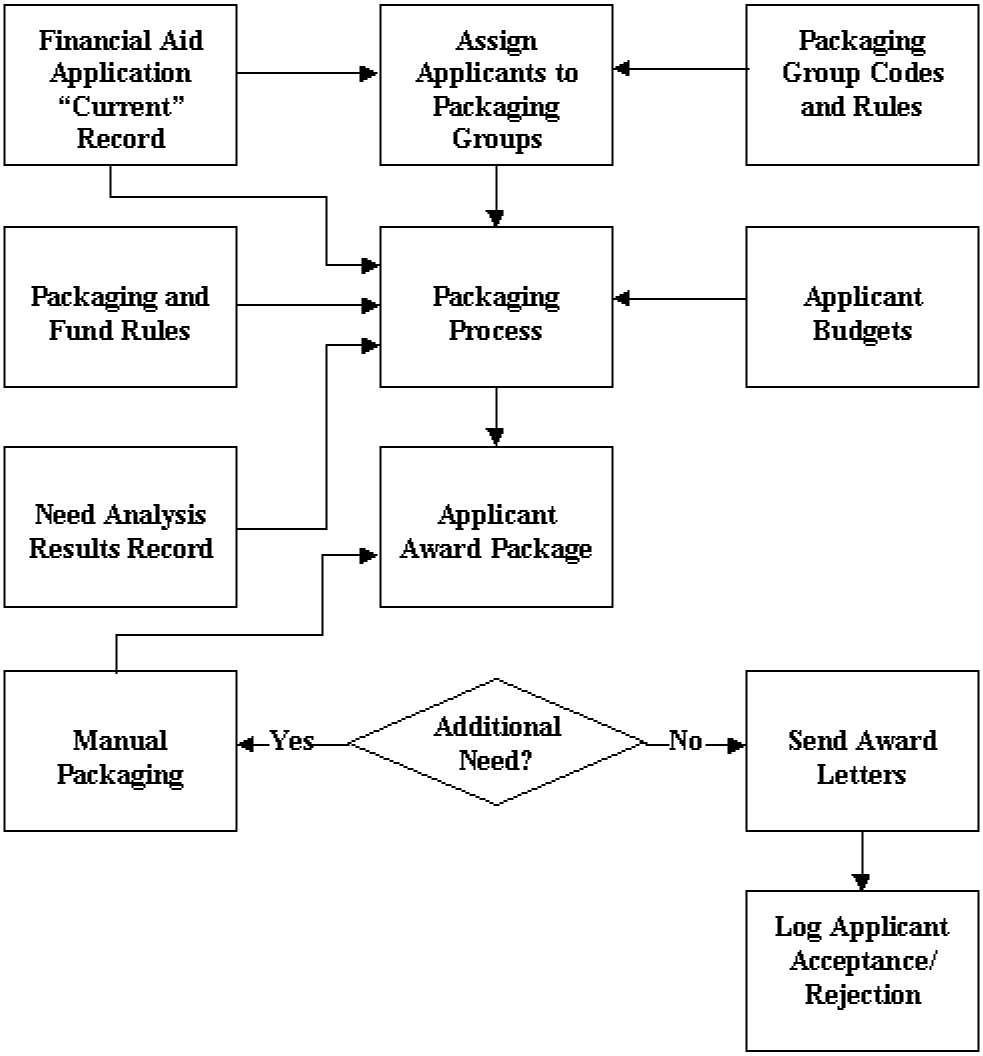
Seven-character name	**Proper name of form, process, report, or table**
RPAAPMT	Package Maintenance Form
RPAAWRD	Award Form
ROARMAN	Financial Aid Record Maintenance
RPAMACC	Award Acceptance Mass Entry Form
RPAMOFF	Award Offer Mass Entry Form
RPAARSC	Resource Form
RHACOMM	Applicant Comments Form
ROAIMMP	Applicant Immediate Process Form
RPIARPY	Contracts and Exemptions Payment Inquiry Form
RPIATRM	Applicant Award by Term Form
ROIDISB	Disbursement Results Form

Major tables

Seven-character name	**Proper name of form, process, report, or table**
RPRAWRD	Applicant Award table
RPRATRM	Applicant Award by Term table
RPRLAPP	Loan Application table

Packaging Process Flow

Diagram


Financial Aid Application "Current" Record
Assign Applicants to Packaging Groups
Packaging Group Codes and Rules
Packaging and Fund Rules
Packaging Process
Applicant Budgets
Need Analysis Results Record
Applicant Award Package
Manual Packaging
Yes
Additional Need?
No
Send Award Letters
Log Applicant Acceptance/ Rejection

Algorithmic Packaging

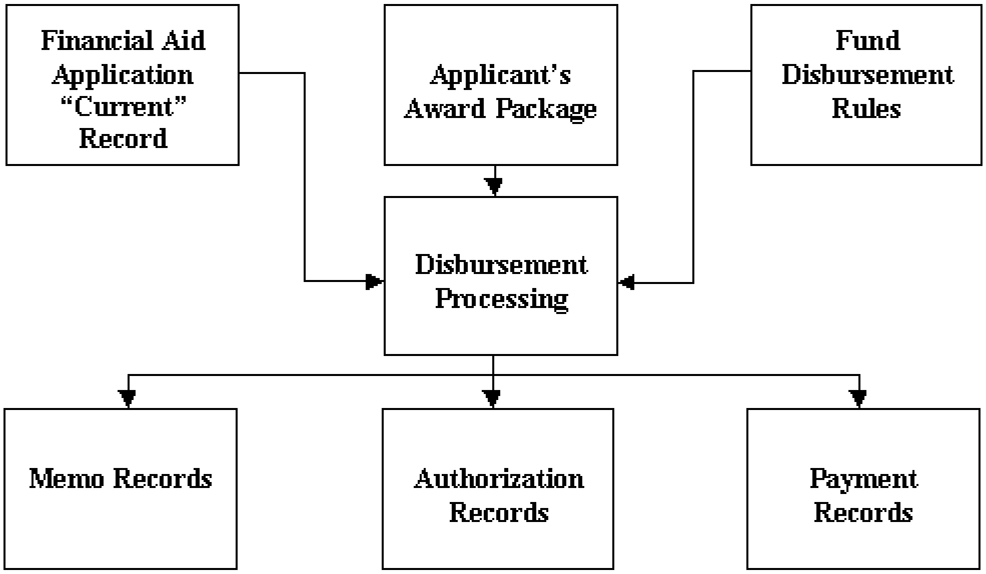
Algorithmic packaging allows clients to use a business rules engine to automatically package students in accordance with the institution's awarding philosophy. In addition to the algorithmic packaging business rules solution, many other changes have been made to enhance the packaging process. Forms and processes have been added or modified to provide a more complete means of automatically packaging students through the use of new algorithmic packaging forms in conjunction with pre-existing Banner Financial Aid rules forms.

Algorithmic Packaging forms, processes, reports and tables

Seven-character name	**Proper name of form, process, report, or table**	**Purpose**
RPRALGR	Algorithmic Packaging Rules Form/Table	Use algorithmic packaging to award funds
RTVALGR	Algorithmic Packaging Rule Code Validation Form/Table	Define algorithmic packaging rule codes for use on RPRALGR, RPRPCKR and RPRGFND
RPRFCAT	Packaging Group Fund Category Maximums Rules Form/Table	Place fund category maximums on packaging groups
RTVFCAT	Fund Category Validation Form/Table	Define fund categories for use on RFRBASE
RPIFAWD	Fund Award Inquiry Form	View all students who have a particular fund for a particular aid year
RFPFBRL	Fund Balance Roll Process	Roll processes fund balances forward from one aid year to the next aid year
RPIFTRM	Fund Award Term Inquiry Form	View all students who have a particular fund for a particular term
RPTNEED	Temporary Packaging Need Table	Columns used in algorithmic packaging rules; rows deleted after use, no indexes or constraints
RFRFCAT	Fund Category Table	Stores fund code, fund category, activity date, user ID, data origin

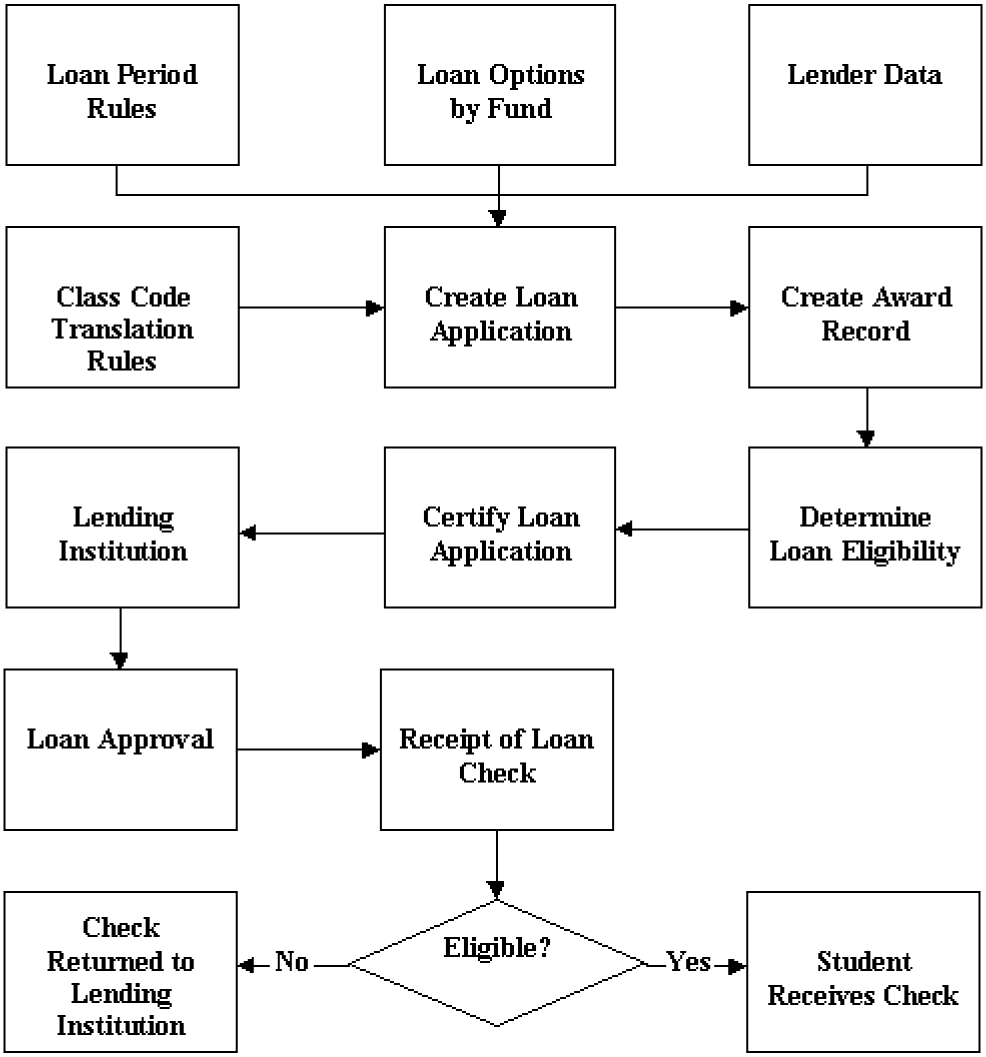
Disbursement Process Flow

Diagram



Loan Process Flow

Diagram



Packaging Process (RPEPCKG)

Packaging process

- Actual or Simulation
- RPBPDRV creates reports:
  - **RPBAWRD.out** – The Award Report that results from the packaging run
  - **RPBFUND.out** – The Fund Report that results from the packaging run

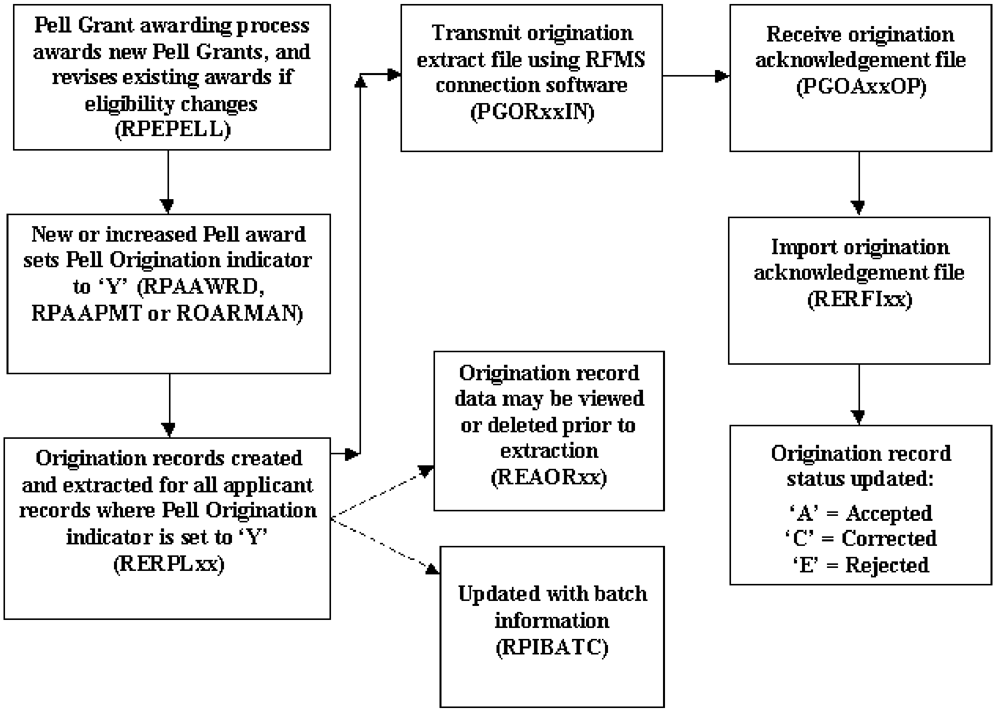
Pell Grant Calculation Process (RPEPELL)

RPEPELL

- Online, batch or from database trigger
- Triggers when certain data changes on the following forms:
- RNARSLT, RBAABUD, RNASUxx, RNANAxx and ROAPELL

RFMS Pell Grant Origination Record Processing

Diagram


Pell Grant awarding process awards new Pell Grants, and revises existing awards if eligibility changes (RPEPELL)
New or increased Pell award sets Pell Origination indicator to 'Y' (RPAAWRD, RPAAPMT or ROARMAN)
Origination records created and extracted for all applicant records where Pell Origination indicator is set to 'Y' (RERPLxx)
Transmit origination extract file using RFMS connection software (PGORxxIN)
Origination record data may be viewed or deleted prior to extraction (REAORxx)
Updated with batch information (RPIBATC)
Receive origination acknowledgement file (PGOAxxOP)
Import origination acknowledgement file (RERFIxx)
Origination record status updated:
'A' = Accepted
'C' = Corrected
'E' = Rejected

Online Award Validation Process (RPOVLDT)

RPOVLDT

This process validates all awards within an applicant's package to ensure that no changes made to the applicant or the fund(s) invalidate the award(s).

Running this process ascertains whether the applicant is eligible for an individual fund being added to the applicant's package either online or during the Packaging process.

The process also validates that the applicant is eligible for system or manual disbursements from the awards within the applicant's package.

Batch Validation Process (RPBVLDT)

RPBVLDT

This process validates all awards within an applicant's package to ensure that no changes made to the applicant or the fund(s) invalidate the award(s).

Running this process ascertains whether the applicant is eligible for an individual fund being added to the applicant's package either online or during the Packaging process.

The process also validates that the applicant is eligible for system or manual disbursements from the awards within the applicant's package.

Disbursement Process (RPEDISB)

RPEDISB

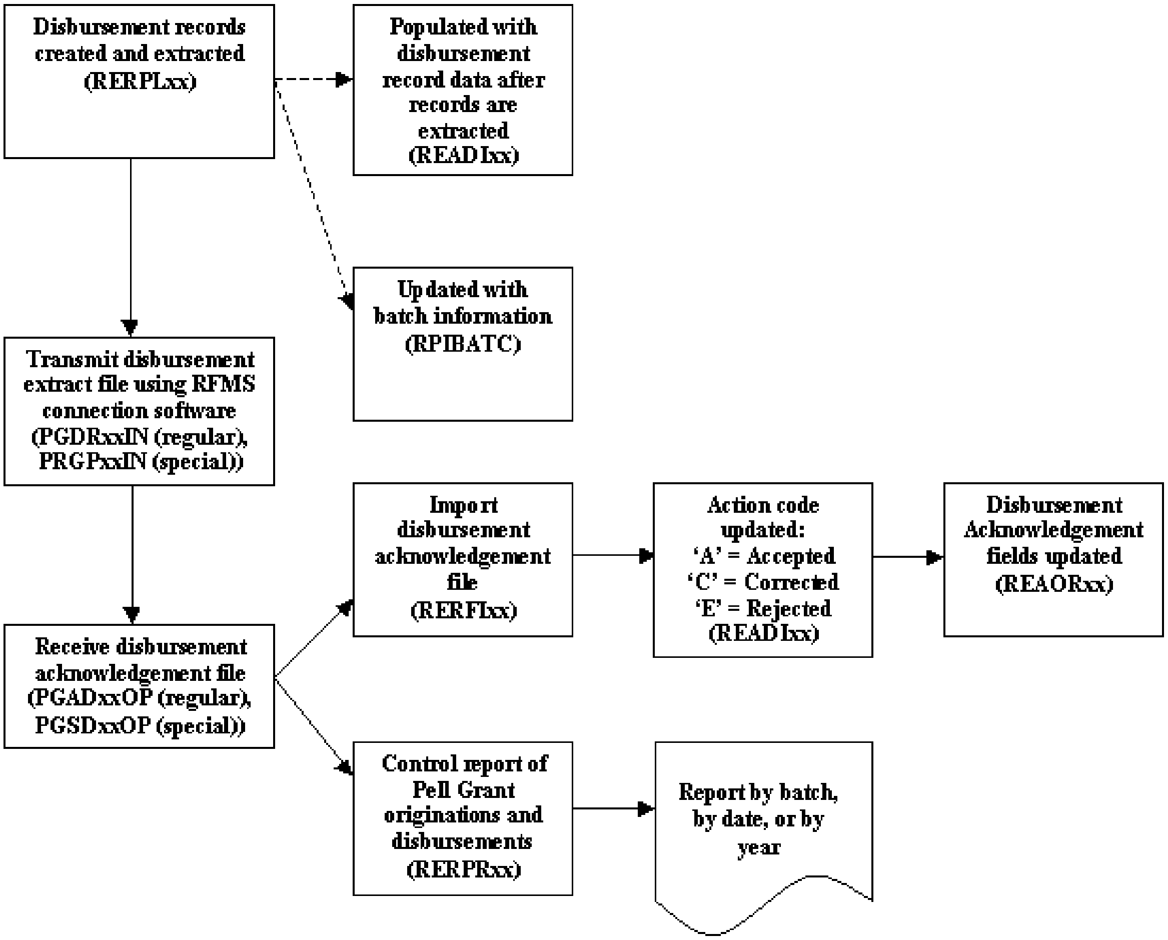
This process passes the amount of deferred Financial Aid available to an applicant within a specific term to the Banner Student System. This is displayed as a memo transaction in the Student Billing module.

The process passes the amount of Financial Aid available to be disbursed to an applicant within a specific term to the Banner Student System. This is displayed as an authorization in the Student Billing module.

The process also passes the amount of Financial Aid scheduled to be disbursed to an applicant within a specific term to the Banner Student System. All scheduled disbursements with a date less than or equal to the processing date that have not been disbursed will be processed. These are displayed as payment transactions in the Student Billing module.

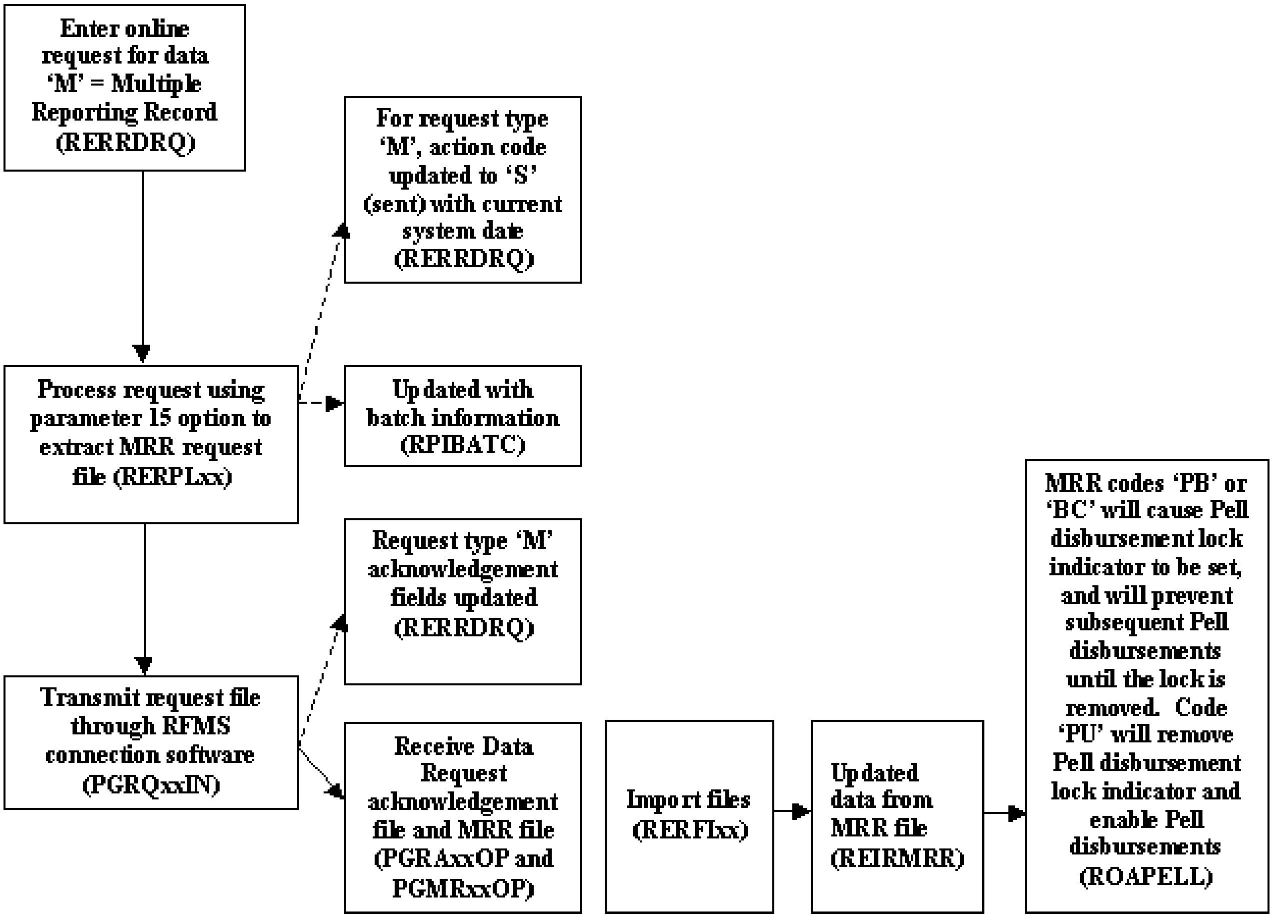
RFMS Pell Grant Disbursement Record Processing

Diagram


Disbursement records created and extracted (RERPLxx)
Populated with disbursement record data after records are extracted (READIxx)
Updated with batch information (RPIBATC)
Transmit disbursement extract file using RFMS connection software (PGDRxxIN (regular), PRGPxxIN (special))
Import disbursement acknowledgement file (RERFIxx)
Action code updated: 'A' = Accepted 'C' = Corrected 'E' = Rejected (READIxx)
Disbursement Acknowledgement fields updated (REAORxx)
Receive disbursement acknowledgement file (PGADxxOP (regular), PGSDxxOP (special))
Control report of Pell Grant originations and disbursements (RERPRxx)
Report by batch, by date, or by year

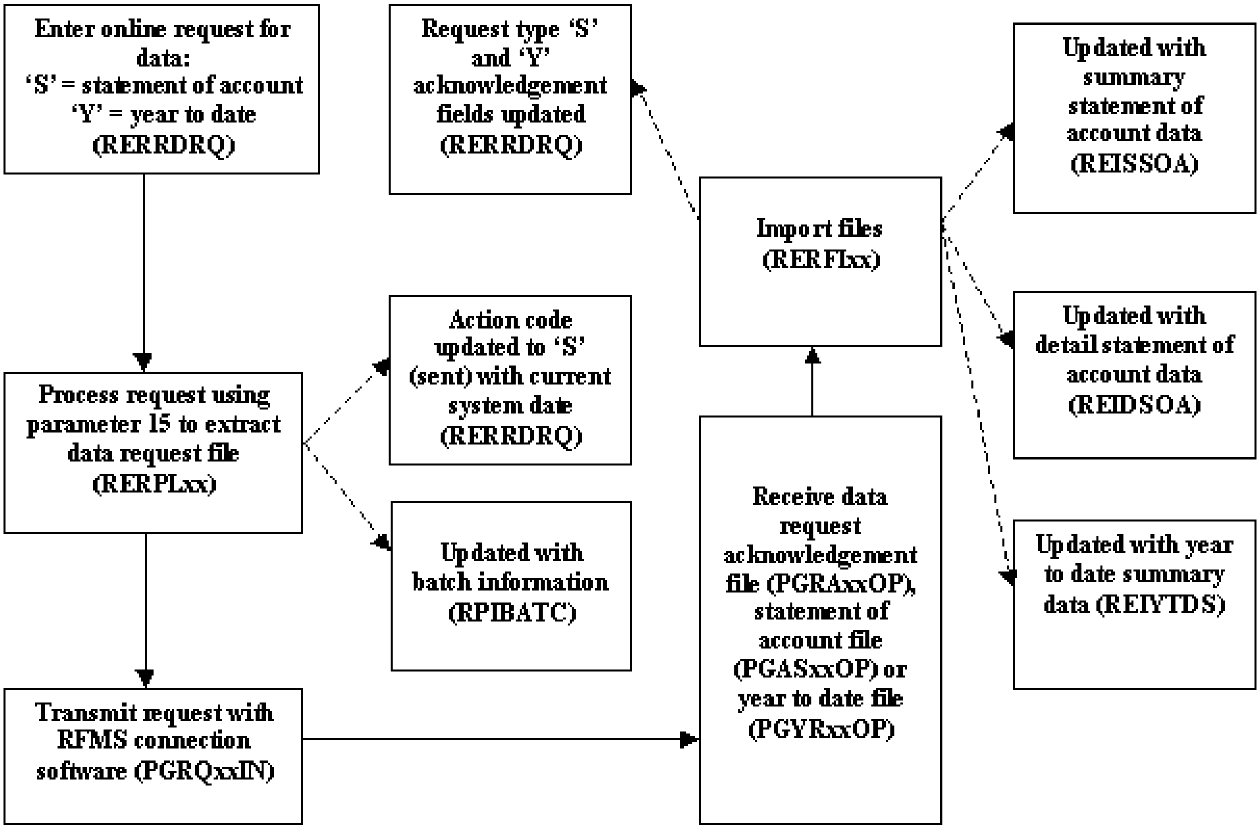
RFMS Multiple Reporting Record (MMR) Processing

Diagram


Enter online request for data 'M' = Multiple Reporting Record (RERRDRQ)
For request type 'M', action code updated to 'S' (sent) with current system date (RERRDRQ)
Process request using parameter 15 option to extract MRR request file (RERPLxx)
Updated with batch information (RPIBATC)
Request type 'M' acknowledgement fields updated (RERRDRQ)
Transmit request file through RFMS connection software (PGRQxxIN)
Receive Data Request acknowledgement file and MRR file (PGRAxxOP and PGMRxxOP)
Import files (RERFIxx)
Updated data from MRR file (REIRMRR)
MRR codes 'PB' or 'BC' will cause Pell disbursement lock indicator to be set, and will prevent subsequent Pell disbursements until the lock is removed. Code 'PU' will remove Pell disbursement lock indicator and enable Pell disbursements (ROAPELL)

RFMS Statement of Account, Year to Date Record Processing

Diagram


Enter online request for data:
'S' = statement of account
'Y' = year to date
(RERRDRQ)
Request type 'S' and 'Y' acknowledgement fields updated (RERRDRQ)
Updated with summary statement of account data (REISSOA)
Import files (RERFIxx)
Action code updated to 'S' (sent) with current system date (RERRDRQ)
Updated with detail statement of account data (REIDSOA)
Process request using parameter 15 to extract data request file (RERPLxx)
Receive data request acknowledgement file (PGRAxxOP), statement of account file (PGASxxOP) or year to date file (PGYRxxOP)
Updated with batch information (RPIBATC)
Updated with year to date summary data (REIYTDS)
Transmit request with RFMS connection software (PGRQxxIN)

Common Functions Forms

Forms

These are general functions that support all of the other Financial Aid processes.

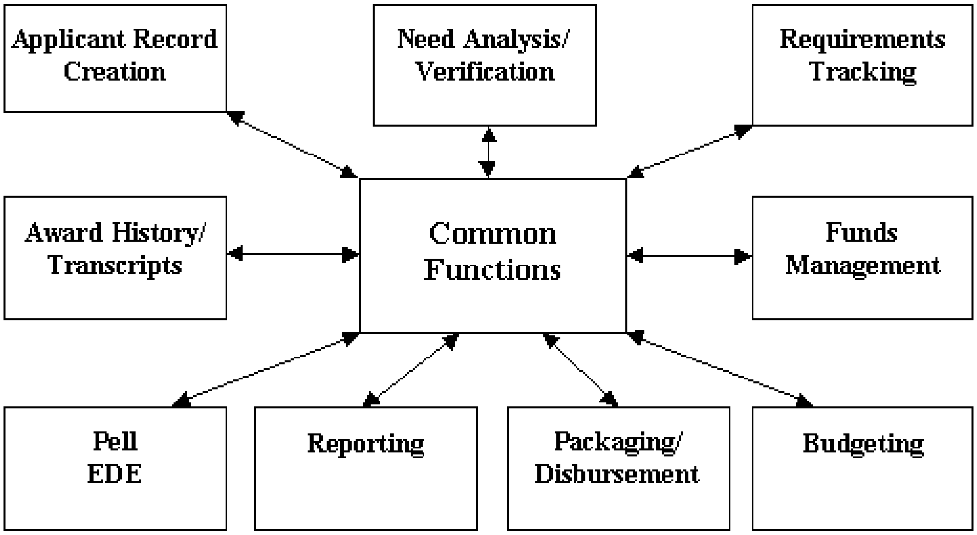
Seven-character name	Proper name of form, process, report, or table
RPAIDEN	Financial Aid Identification Form
ROAIDEN	Person Name/ID Search Form
ROAPERS	Financial Aid General Person Form
ROIAIDY	Aid Year Inquiry Form
ROIILOG	Data Log Inquiry Form
ROISAPR	SAP Code Inquiry Form
ROISARI	SAR ID Inquiry Form
ROAFSAP	FISAP Person Maintenance Form
ROIALOG	Applicant Data Log Inquiry Form
ROAALOG	Applicant Data Log Application Form

Major tables

Seven-character name	Proper name of form, process, report, or table
RORSTAT	Applicant Status Table
RORPELL	PELL Grant Payment Schedule Table
RORHOLD	Applicant Holds Table
RORDVAL	Audit Log Form/Data Code Table
ROBALOG	Audit Log Table
ROBUSDF	Applicant User-Defined Data Table
RORCMPL	Selection Identifier Compiled Statements
RORSAPR	Satisfactory Academic Progress Rules Table
RORMESG	Financial Aid Messages
RORENRL	Financial Aid Enrollment Table
RORCRHR	Credit Hours Table
ROBINST	Financial Aid Institution Table
RORGSQL	Generated SQL Selection Identifier Table
RORCOLS	All Financial Aid Columns Table

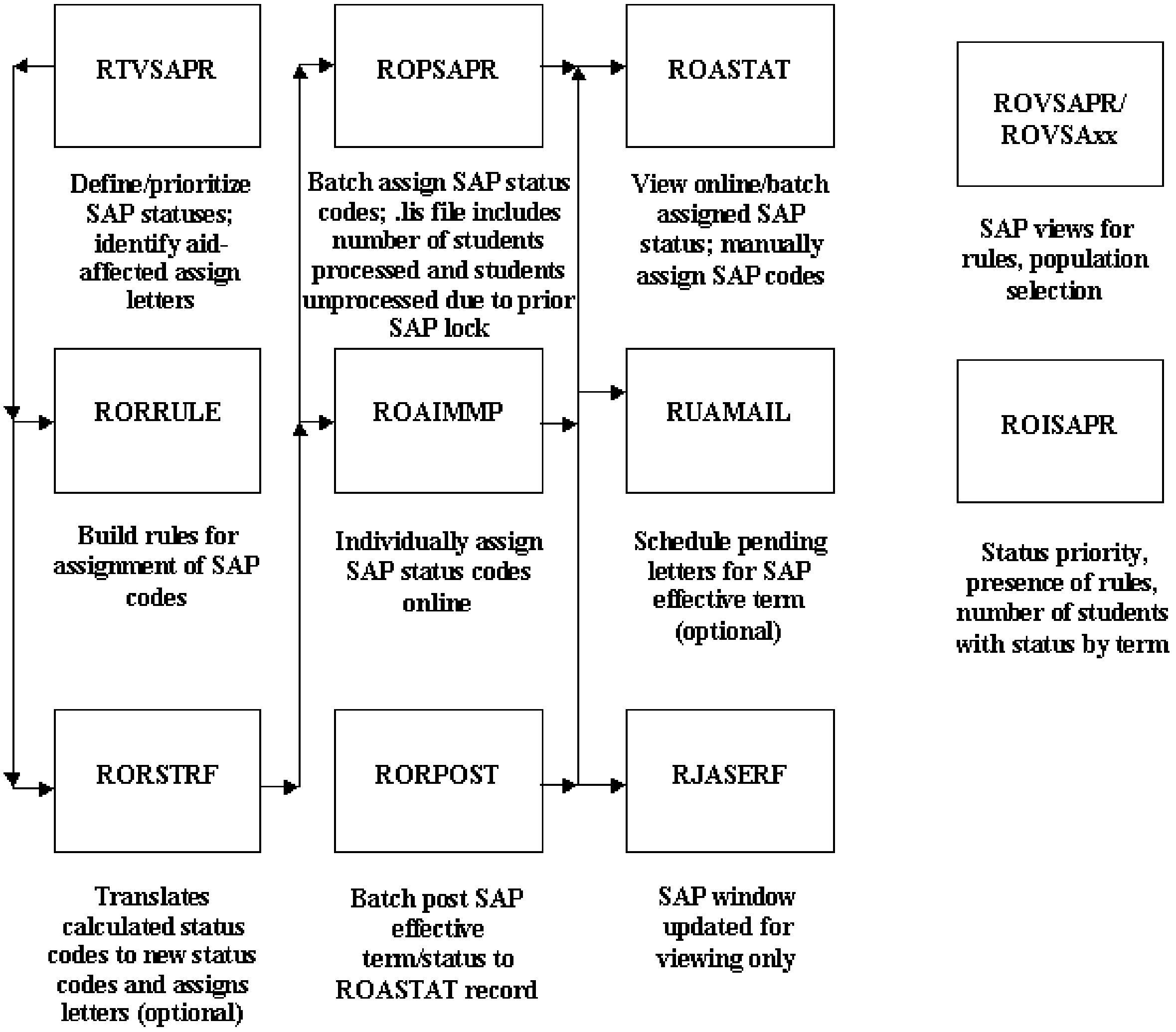
Common Functions Process Flow

Diagram



Satisfactory Academic Progress Process

Diagram


RTVSAPR
Define/prioritize SAP statuses; identify aid-affected assign letters
RORRULE
Build rules for assignment of SAP codes
RORSTRF
Translates calculated status codes to new status codes and assigns letters (optional)
ROPSAPR
Batch assign SAP status codes; .lis file includes number of students processed and students unprocessed due to prior SAP lock
ROAIMMP
Individually assign SAP status codes online
RORPOST
Batch post SAP effective term/status to ROASTAT record
ROASTAT
View online/batch assigned SAP status; manually assign SAP codes
RUAMAIL
Schedule pending letters for SAP effective term (optional)
RJASERF
SAP window updated for viewing only
ROVSAPR/ ROVSAxx
SAP views for rules, population selection
ROISAPR
Status priority, presence of rules, number of students with status by term

Reporting

Reporting module

The Reporting module contains standardized reports that extract data from all other Financial Aid modules.

Commonly used reports

Financial Aid Roll Rule Report (ROPROLL)

- Create the scripts that rolls year rules
- By aid year (Group, Fund, Budgeting, Tracking, Packaging)

FISAP Report (RORFSxx)

- Prints FISAP report

Other reports to know

- Disbursement Process (RPEDISB)
- Pell Awarding Process (RPEPELL)
- Packaging Process (RPEPCKG)
- [Year-specific] Need Analysis Process (RNEINxx)
- INAS

Award History and Transcripts

Forms

This module maintains historical applicant data and produces Financial Aid transcripts.

Seven-character name	**Proper name of form, process, report, or table**
RHRTRNS	Transcript Rules Form
RHAPBAW	Pre-Banner Award Summary Form
RHAPTRN	Prior Institution Transcript Form
RHIAFSH	Fund Sequence History Inquiry Form
RHACOMM	Applicant Comments Form
RHATINF	Transcript Information Form
RHARQST	Financial Aid Transcript Request Form

Tables

Seven-character name	**Proper name of form, process, report, or table**
RHBTINF	Institution Transcript Information Table
RHBTRNS	Prior Institution Transcript Base Table
RHRAWRD	Prior Institution Transcript Award Table
RHRCOMM	Applicants Comments

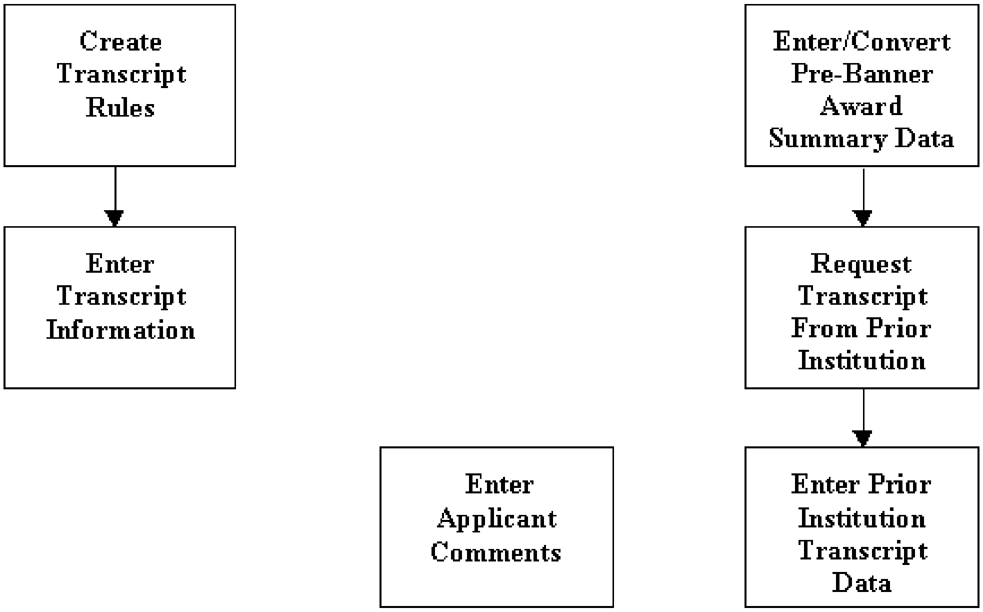
Processes and procedures

Financial Aid Transcript (RHRTRAN)

- Request federal student aid on a student who has attended a previous post-secondary institution

Award History and Transcripts Process Flow

Diagram


Create Transcript Rules
Enter Transcript Information
Enter/Convert Pre-Banner Award Summary Data
Request Transcript From Prior Institution
Enter Applicant Comments
Enter Prior Institution Transcript Data

Student Employment

Forms

This module maintains data related to student employment financial assistance.

Seven-character name	Proper name of form, process, report, or table
RJASEAR	Student Employment Authorization Form
RJASERF	Student Employment Referral Form
RJASEME	Student Employment Mass Entry Form
RJISEWH	Student Employment Work History Form

Major tables

Seven-character name	Proper name of form, process, report, or table
RJBPLBD	Placement Base Data Table
RJBJOBT	Job Title Table
RJRSEAR	Student Employment Authorization Table
RJRSERF	Student Employment Referral Table
RJRSEWH	Student Employment Work History Table

Processes and procedures

Payroll Load Process (RJRLOAD)

- Interface payroll data from Banner Human Resources to Banner Financial Aid

Logging

Forms and tables

- Institution Financial Aid Options Form (ROAINST) (audit logging)
  - Budget logging
  - Needs Analysis logging
  - Packaging logging
  - Pell Correction logging
- Audit Log table (ROBALOG)

New Year Roll Process

Financial Aid New Year Roll Process (ROPROLL)

Steps required to roll selected aid year specific information from one aid year to the new aid year include:

- creation of an aid year ROAINST instance
- running the ROPROLL process from Job Submission
- manually reviewing and editing rules and validation tables FOLLOWING the roll.

Important Information

BEFORE your IT Department can possibly install the appropriate Banner New Aid Year release, you MUST first create an appropriate aid year instance of ROAINST.

Very Important Considerations

If a row already exists in a table for the aid year to which you roll data, an error message indicates that rows currently exist for the table; only non-existing rows are rolled.

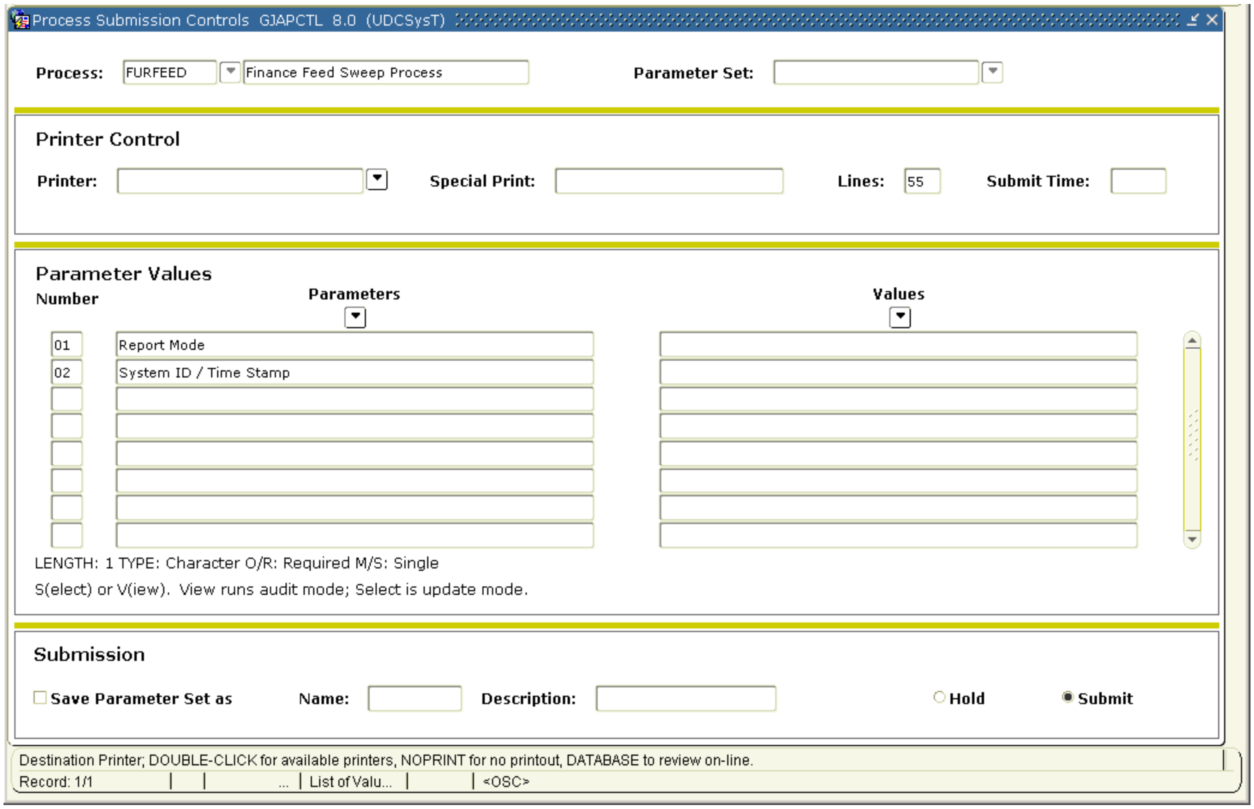
Use caution when copying or manually entering aid year data for a module that you wish to automatically roll from the prior aid year by running ROPROLL.

ROPROLL Decisions

What data do you want to roll automatically?

Parameter		Value
01	From Aid Year Code	ex. 0708
02	To Aid Year Code	ex. 0809
03	Perform roll of Group Rules	Y or N
04	Perform roll of FUND data	Y or N
05	Perform roll of BUDGET data	Y or N
06	Perform roll of TRACKING data	Y or N
07	Perform roll of PACKAGING data	Y or N
08	Perform roll of COMMON data	Y or N
09	Perform roll of EMPLYMENT data	Y or N

Banner form



Process Submission Controls GJAPCTL 8.0 (UDCSysT)

Process: FURFEED Finance Feed Sweep Process

Parameter Set:

Printer Control

Printer: Special Print: Lines: 55 Submit Time:

Parameter Values

Number	Parameters	Values
01	Report Mode	
02	System ID / Time Stamp	

LENGTH: 1 TYPE: Character O/R: Required M/S: Single

S(elect) or V(iew). View runs audit mode; Select is update mode.

Submission

□ Save Parameter Set as Name: Description: ○ Hold ● Submit

Destination Printer; DOUBLE-CLICK for available printers, NOPRINT for no printout, DATABASE to review on-line.

Record: 1/1 | | ... | List of Valu... | | <OSC>

Common functions forms and data

- ROAINST
- RTVTREQ RTVPGRP
- RTVTRST STVLEVL
- STVTERM RORUSER
- RTVAPRD RORTPRD
- RTVSAPR RORUSER
- RTVTGRP RORPOST
- RTVBGRP RCRINFR

The following examples are based on a 2007-2008 new year roll.

Record creation module

- ROAINST – Set up Institution Financial Aid Options
- RCRINFR - Create Data Source Matching Rules
- RTVINFC - Update Data Source Code
- RCRTPTR – Review for accuracy
- RPROPTS – Review Packaging and Disbursement options

Need Analysis module

- RPROPTS – Check Packaging Fields for New Aid Year
- ROAINST – Check for completion of 0708 Institutional Parameters
- RORTPRD – Update with 0708 Aid Periods and Terms
- RFRBASE – Review and Update Funds for 0708
- RFRMGMT – Review Funds Management Forms for 0708
- RPRCLSS – Review STVLEVL, STVCLAS and Banner Financial Aid Codes

Tracking validation tables and rules

- ROAINST – Review 0708 Tracking Parameters
- RTVTGRP – Review Tracking Group Validation Form
- RTVTREQ – Review and Edit Requirements Tracking Validation Form for 0708
- RRRGREQ – Update 0708 Tracking Group/Requirements and Rules
- RORRULE – Create Group Assignment Rules for 0708 Using the Copy Function
- RORMESG – Associate any new messages for 0708 to Tracking Requirement Codes

Budgeting validation tables and rules

- ROAINST – Create 0708 Budget Defaults
- RTVAPRD – Confirm Aid Periods for 0708
- RTVBTYP – Confirm 0708 Budget Types
- RBRCOMP – Update Budgets for 0708
- RTVBGRP – Confirm 0708 Budget Groups
- RBRGTYTP – Confirm 0708 Budget Types
- RORMESG – Associate Budget Messages created on RTVMESG with Budget Groups
- RORRULE – Create 0708 Budget Group Assignment Rules

Funds Management validation tables and rules

- ROAINST –Set Pell Fund code Defaults
- RFRBASE – Update FUNDS for the 0708 Aid Year
- RFRMGMT – Build 0708 Aid-year Specific Data and Rules for Each Fund You Intend to Use (Initial Page of RFRMGMT Form does NOT roll)
- RFRDEFA – Create Term Award and Disbursement Schedules for 0708 by Aid Period
- RFRASCH – Create OPTIONAL Fund/Term Award and Disbursement Schedules for the 0708 Aid Year
- RORRULE – Create OPTIONAL Fund Awarding Rules for 0708 by Fund Code

Packaging and disbursement

- ROAINST – Confirm that Full-Time, Half-Time, ¾-Time Values are associated with EACH STVTERM Code and EACH STVLEVL code for 0708 (pg. 4)

- RPROPTS – Create 0708 Packaging and Disbursement Options

- RTVPGRP – Confirm 0708 Packaging Groups

- RPRGFND – Associate 0708 Packaging Groups with Funds, indicating the Priority Order and Respective Minimum and Maximum Awards

- RPRCLSS – Create Class Code Translations for Disbursements

- RORRULE – Create Packaging Group Assignment Rules for 0708 if using Automated Packaging

- RPRDEFR – OPTIONAL Set-up for Institutions Using Self-Help, Gap or Equity Packaging

- RPRPCKR – Create Self-Help, Gap or Equity Packaging Rules if pertinent to Packaging Philosophy

- RPRAUDT – Create OPTIONAL rules for Audit Grading Mode if using Adjusted Enrollment Hrs.

- RORRULE – Create 0708 OPTIONAL Group/Fund Awarding Rules

- RPREXPT / RPRCONT – Create OPTIONAL Exemption Interface Rules and/or Contract Interface Rules

- RPRFSRC – Create OPTIONAL Fund Source Maximum Rules for 0708

- RPRLNDA - If Hold/Release processing is to be utilized

Algorithmic Packaging

- RFPFBRL – Roll process for fund balances from one aid year to the next

Loan module

- ROAINST – Update 0708 Institutional Loan Parameters
- RPRLOPT – Create 0708 Packaging and Disbursement Options for Each Loan Fund
- RPRLPRD – Create 0708 Loan Periods
- RFRMGMT – Verify that the Loan Process Indicator is Checked for Each Loan Fund to be Awarded in 0708
- RPRLNDR – Review Lender Information and Update for 0708

EDE module

- ROAINST – Update EDE Corrections Indicator form, and Complete Pell, COD and Common line Windows for 0708
- RORDATA – Verify Presence of this SGHE-Delivered Table. Modifications to RORDATA Are NOT Recommended

SUMMARY – Three Steps to Roll Aid Year Data

- Create an '0708' instance of ROAINST prior to installation of new year release
- Run the ROPROLL process from Job Submission
- If desired run the roll process for fund balances, RFPFBRL
- Review/Edit each of the Financial Aid modules listed in the New Year Start-up Checklist

For more details on the ROPROLL process, refer to the *Banner Financial Aid New Aid Year Start-Up Checklist.*

Security in Financial Aid

Security

In general, no forms should be required that are not already included in Financial Aid security, such as student accounts, admissions status, or others.

Which forms and menus will be given to different levels of Financial Aid workers is reviewed in a table in the **Table Indexes** section of this workbook.

PIN Maintenance enhancement

Banner General 8.0 includes an enhancement for user PIN (password) security. Please refer to the *Banner General 8.0 Release Guide* for more information on these changes.

Makefiles

Purpose

Makefiles maintain, update, and regenerate groups of programs.

Important flag

`-f MakeFile` - Specifies a makefile to read instead of the default makefile

Commands

```
make -f sctproc.mk {program_name}
make -f sctprocb.mk {program_name}
```

Compile All Programs

C compiles

Rescmplc.shl

COBOL compiles

Rescmpl.shl

Rescmprr.shl (Profile)

Rescmp.shl (Inas)

Self Check

Directions

Use the information you have learned in this workbook to complete this self check activity.

Exercise 1: Applicant record creation

Other than data load, how can new records be created?

Explain what the process RCPDTMP does.

Exercise 2: RORRULE

Write a rule to capture in a tracking group those individuals that have a 'C' code on their current records. You will need to create a tracking group and tracking code. To complete this assignment, you should be able to run RORGRPS and place a requirement on student records that fit the above category.

Exercise 3: Packaging and disbursement

Write a script that will list the student ID, aid year, fund code, and Paid Award Amount for a student whose ID and aid year are entered as parameters.

Exercise 4: SAP History

Write a script to select ID, name, progress code, and term for each student coded in Banner for Satisfactory Academic Progress.

Some students are coded for more than one term.

If students have more than one term, select the most current term and select that term's progress code.

The academic progress table is RORSAPR.

Exercise 5

Write a script to list the student ID, aid year, term code, fund description, and offered award amount for the term of a student whose ID and term code is entered as a parameter. Allow for the use of wild card entry for both ID and term code.

Exercise 6: Database Structure

Write a script to return those students that have different EFCs. Script should return student ID, formatted name, EFCs and indicate if record has a 'lock'. Prompt the user for aid year.

Exercise 7: Database Structure Advanced

Write a script to select ID, name, unmet need, EFC, FWS work preference, and total paid awards for each student. Use wild card selection for aid year.

Exercise 8: Database Structure Advanced

Modify your script from the previous exercise to select only students that indicated they wanted FWS for 2007-08.

Exercise 9: Database Structure Advanced

Modify your script from the previous exercise to select students where they have any awards paid for aid year.

Self Check – Answer Key

Exercise 1: Applicant record creation

Other than data load, how can new records be created?

- **New records can be created on RRAAREQ, RNANAxx, or ROASTAT.**

Explain what the process RCPDTMP does.

- **It cleans out the temporary data load tables.**

Exercise 2: RORRULE

Write a rule to capture in a tracking group those individuals that have a 'C' code on their current records. You will need to create a tracking group and tracking code. To complete this assignment, you should be able to run RORGRPS and place a requirement on student records that fit the above category.

1. **`Create code on RTVTREQ`**
2. **`Create Group on RTVTGRP`**
3. **`Find field on RNARSLT: RCRAPP2_ELIGIBILITY_IND = '1'`**
4. **`Write RORRULE:`**

   ```
           RCRAPP2_ELIGIBILITY_IND = '1' and
           RCRAPP1_CURR_REC_IND = 'Y'
   ```
5. **`Save, Compile, Execute Group to test.`**
6. **`Run RORGRPS.`**

Exercise 3: Packaging and disbursement

Write a script that will list the student ID, aid year, fund code, and Paid Award Amount for a student whose ID and aid year are entered as parameters.

```
select spriden_id id,
       rprawrd_aidy_code aidyr,
       rprawrd_fund_code fund_code,
      '$'||rprawrd_paid_amt pd_amount
  from spriden, spbpers, rprawrd
 where spriden_pidm = rprawrd_pidm
   and spbpers_pidm = spriden_pidm
   and spbpers_dead_ind is null
   and spriden_change_ind is null
   and spriden_entity_ind = 'P'
   and spriden_id = '&id'
   and rprawrd_paid_amt is not null
   and rprawrd_aidy_code = '&aidyr';
```

Exercise 4: SAP History

Write a script to select ID, name, progress code, and term for each student coded in Banner for Satisfactory Academic Progress.

Some students are coded for more than one term.

If students have more than one term, select the most current term and select that term's progress code.

The academic progress table is RORSAPR.

```
SELECT spriden_id, spriden_last_name, spriden_first_name, x.rorsapr_term_code,
       x.rorsapr_sapr_code
  FROM spriden, spbpers, rorsapr x
 WHERE x.rorsapr_term_code = (SELECT MAX (y.rorsapr_term_code)
                                FROM rorsapr y
                               WHERE y.rorsapr_pidm = x.rorsapr_pidm)
   AND x.rorsapr_pidm = spriden_pidm
   AND x.rorsapr_pidm = spbpers_pidm(+)
   AND spbpers_dead_ind IS NULL
   AND spriden_change_ind IS NULL
   AND spriden_entity_ind = 'P'
```

Exercise 5

Write a script to list the student ID, aid year, term code, fund description, and offered award amount for the term of a student whose ID and term code is entered as a parameter. Allow for the use of wild card entry for both ID and term code.

```
select spriden_id  id,
       rpratrm_aidy_code aidy,
       rpratrm_term_code term_code,
      rfrbase_fund_title fund_desc,
      '$'||rpratrm_offer_amt  offered_amt
  from rfrbase, spbpers, rpratrm, spriden
 where spriden_pidm = rpratrm_pidm
   and rpratrm_pidm = spbpers_pidm
   and rpratrm_fund_code = rfrbase_fund_code
   and spbpers_dead_ind is null
   and spriden_change_ind is null
   and spriden_entity_ind = 'P'
   and spriden_id like '&id'
   and rpratrm_term_code like '&term_code'
 order by spriden_id, rpratrm_aidy_code,
          rpratrm_term_code, rpratrm_fund_code
```

Exercise 6: Database Structure

Write a script to return those students that have different EFCs. Script should return student ID, formatted name, EFCs and indicate if record has a 'lock'. Prompt the user for aid year.

```
select spriden_id ID,
       f_format_name(spriden_pidm,'LF30') Name,
       rcvappl_pell_pgi SAR_EFC,
       rcvappl_pgi Pell_EFC,
       rcvappl_tfc SYS_EFC,
       rcvappl_used_trans_no Trans,
       rorstat_lock_ind Locked
  from rcvappl, rorstat, spriden
 where rcvappl_pidm = rorstat_pidm
   and rorstat_pidm = spriden_pidm
   and spriden_change_ind is null
   and spriden_entity_ind = 'P'
   and rcvappl_aidy_code = rorstat_aidy_code
   and rcvappl_infc_code = 'EDE'
   and rcvappl_aidy_code = '&Aid_Year'
   and rcvappl_curr_rec_ind = 'Y'
   and (rcvappl_tfc <> rcvappl_pgi or
        rcvappl_tfc <> rcvappl_pell_pgi or
        rcvappl_pgi <> rcvappl_pell_pgi)
 order by name;
```

Exercise 7: Database Structure Advanced

Write a script to select ID, name, unmet need, EFC, FWS work preference, and total paid awards for each student. Use wild card selection for aid year.

```
select  spriden_id "ID", spriden_last_name|| ', '
        ||spriden_first_name "NAME",
        rorstat_pri_sar_pgi "EFC",
        rorstat_unmet_need "UNMET_NEED",
        rcrapp4_work_study "WORK",
        sum(rprawrd_paid_amt) "PAID_AMT"
  from  spriden, spbpers, rprawrd, rorstat, rcrapp4,
        rcrapp1
where  rcrapp1_aidy_code = '&aidy_code'
   and  rcrapp1_curr_rec_ind = 'Y'
   and  rcrapp1_pidm = rcrapp4_pidm
   and  rcrapp1_infc_code = rcrapp4_infc_code
   and  rcrapp1_seq_no = rcrapp4_seq_no
   and  rcrapp1_aidy_code = rcrapp4_aidy_code
   and  rcrapp1_pidm = rorstat_pidm
   and  rcrapp1_aidy_code = rorstat_aidy_code
   and  rcrapp1_pidm = rprawrd_pidm
   and  rcrapp1_pidm = spbpers_pidm(+)
   and  rcrapp1_pidm = spriden_pidm
   and  spbpers_dead_ind is null
   and  spriden_change_ind is null
   and  spriden_entity_ind = 'P'
 group  by spriden_id, spriden_last_name,
        spriden_first_name, rorstat_pri_sar_pgi,
        rorstat_unmet_need, rcrapp4_work_study
```

Exercise 8: Database Structure Advanced

Modify your script from the previous exercise to select only students that indicated they wanted FWS for 2006-2007.

```
select  spriden_id "ID", spriden_last_name|| ','
        ||spriden_first_name "NAME",
        rorstat_pri_sar_pgi "EFC",
        rorstat_unmet_need "UNMET_NEED",
        rcrapp4_work_study "WORK",
        sum(rprawrd_paid_amt) "PAID_AMT"
  from  spriden, spbpers, rprawrd, rorstat, rcrapp4,
        rcrapp1
 where  rcrapp1_aidy_code = '&aidy_code'
   and  rcrapp1_curr_rec_ind = 'Y'
   and  rcrapp1_pidm = rcrapp4_pidm
   and  rcrapp1_infc_code = rcrapp4_infc_code
   and  rcrapp1_seq_no = rcrapp4_seq_no
   and  rcrapp1_aidy_code = rcrapp4_aidy_code
   and  rcrapp1_pidm = rorstat_pidm
   and  rcrapp1_aidy_code = rorstat_aidy_code
   and  rcrapp1_pidm = rprawrd_pidm
   and  rcrapp1_pidm = spbpers_pidm(+)
   and  rcrapp1_pidm = spriden_pidm
   and  spbpers_dead_ind is null
   and  spriden_change_ind is null
   and  spriden_entity_ind = 'P'
   and  rcrapp4_work_study = '1'
 group  by spriden_id, spriden_last_name,
        spriden_first_name, rorstat_pri_sar_pgi,
        rorstat_unmet_need, rcrapp4_work_study
```

Exercise 9: Database Structure Advanced

Modify your script from the previous exercise to select students where they have any awards paid for aid year.

```
select  spriden_id "ID", spriden_last_name|| ','
        ||spriden_first_name "NAME",
        rorstat_pri_sar_pgi "EFC",
        rorstat_unmet_need "UNMET_NEED",
        rcrapp4_work_study "WORK",
        sum(rprawrd_paid_amt) "PAID_AMT"
  from  spriden, spbpers, rprawrd, rorstat, rcrapp4,
        rcrapp1
 where  rcrapp1_aidy_code = '&aidy_code'
   and  rcrapp1_curr_rec_ind = 'Y'
   and  rcrapp1_pidm = rcrapp4_pidm
   and  rcrapp1_infc_code = rcrapp4_infc_code
   and  rcrapp1_seq_no = rcrapp4_seq_no
   and  rcrapp1_aidy_code = rcrapp4_aidy_code
   and  rcrapp1_pidm = rorstat_pidm
   and  rcrapp1_aidy_code = rorstat_aidy_code
   and  rcrapp1_pidm = rprawrd_pidm
   and  rcrapp1_pidm = spbpers_pidm(+)
   and  rcrapp1_pidm = spriden_pidm
   and  spbpers_dead_ind is null
   and  spriden_change_ind is null
   and  spriden_entity_ind = 'P'
   and  rprawrd_paid_amt <> 0
group  by spriden_id, spriden_last_name,
        spriden_first_name, rorstat_pri_sar_pgi,
        rorstat_unmet_need, rcrapp4_work_study
```

Job Submission

Introduction

This chapter provides an overview of the Job Submission process in Banner.

Objectives

At the conclusion of this chapter, participants will be able to:

- identify the components of Job Submission in Banner
- view Job Submission at the functional and database levels

Job Submission Overview

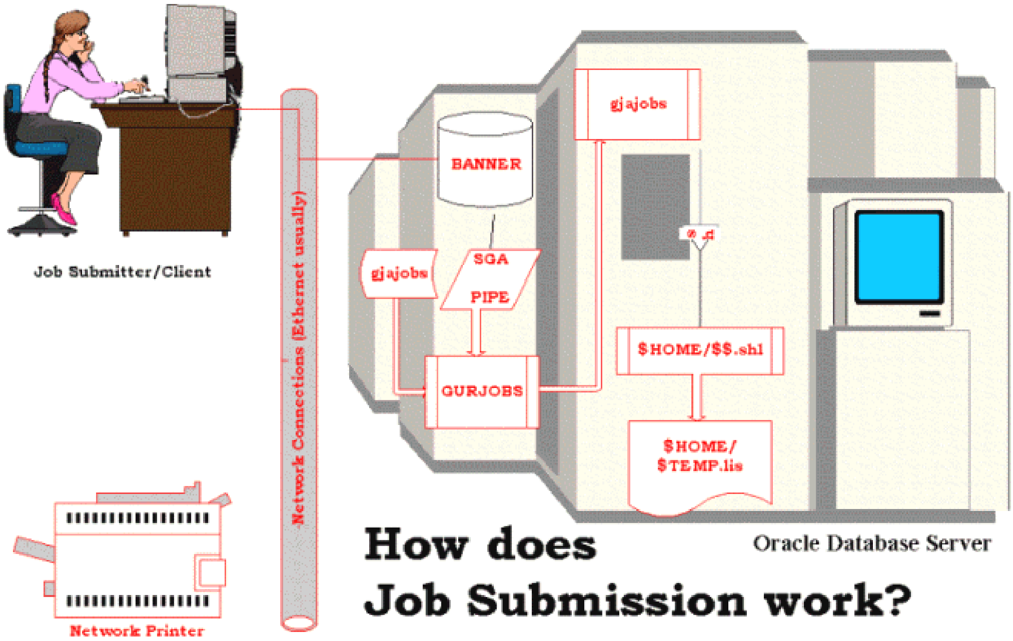
Description

The Job Submission process allows users to identify and define job parameters, work with job scheduling and printing, schedule jobs and control job printing.

How does Job Submission work?

- It defines the parameters used to execute any given process
- It communicates with the database server environment to schedule the process
- It communicates with the database server environment to print the output of the process

Diagram



JOBSUB ID

jobsub is a System user ID, not an Oracle user ID. *jobsub* starts Job Submission table (GURJOBS) for each database.

All reports (.lis files) reside in *jobsub*'s directory.

Start Job Submission-Sleep/Wake

GURJOBS is a PRO*C program created to handle requests in a client-server configuration.

GURJOBS must be running in Sleep/Wake mode prior to Job Submission. The user ID *jobsub* should start this process. All output from Job Submission is owned by *jobsub*.

GURJOBS is dependent upon the Oracle database pipe (dbmspipe.sql).

Establish printers

The Printer Validation Form (GTVPRNT) defines the printers that are accessible to users.

Printers must be accessible from the network. The database server must know about the printer and the print queue.

NOPRINT and DATABASE may be defined as printer codes, but do not route to a printer.

Define processes

The Process Maintenance Form (GJAJOBS) allows for maintenance of overall process attributes.

- Process name
- Process description
- Process type (Pro*C, Pro*COBOL, etc.)

Define process parameters

Parameters for any given process can be maintained dynamically through Banner forms.

- Parameter Definition Form (GJAPDEF)
- Parameter Value Validation Form (GJAPVAL)
- Default Parameter Value Validation Form (GJAPDFT)

Submit job

The Process Submission Control Form (GJAPCTL) creates a one-up number to uniquely identify the job, and passes the user-entered parameters and the one-up number to the table GJBPRUN.

At run time, all processes access GJBPRUN for their parameters.

Review results

The Process Results Form (GJARSLT) allows for the review of the process log.

Output files

All process output (.lis files) reside in the *jobsub* home directory. Log files (.log) also reside in the *jobsub* home directory.

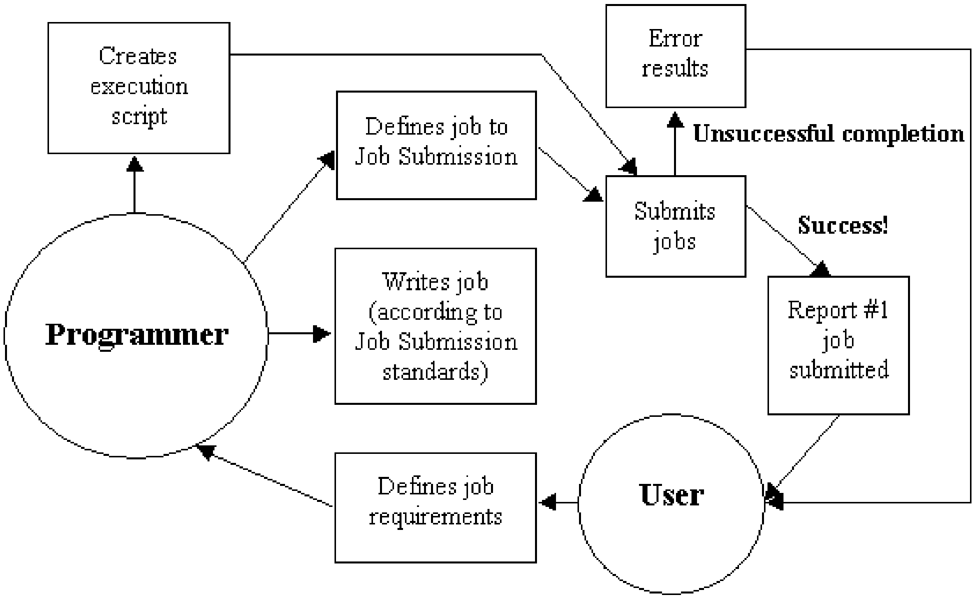
Job Submission Process Flow

Process flow

1. Job Submission Form (GJAJOBS)
    - Define and maintain jobs
2. Parameter Definition Form (GJAPDEF)
    - Name and define parameters
3. Default Parameter Value Definition Form (GJAPDFT)
    - Define default parameters
4. Parameter Value Form (GJAPVAL)
    - Enter parameter values for a job
5. Job Submission Form (GJAPCTL)
    - Run a job
6. Process Results Form (GJARSLT)
    - View the results of a job

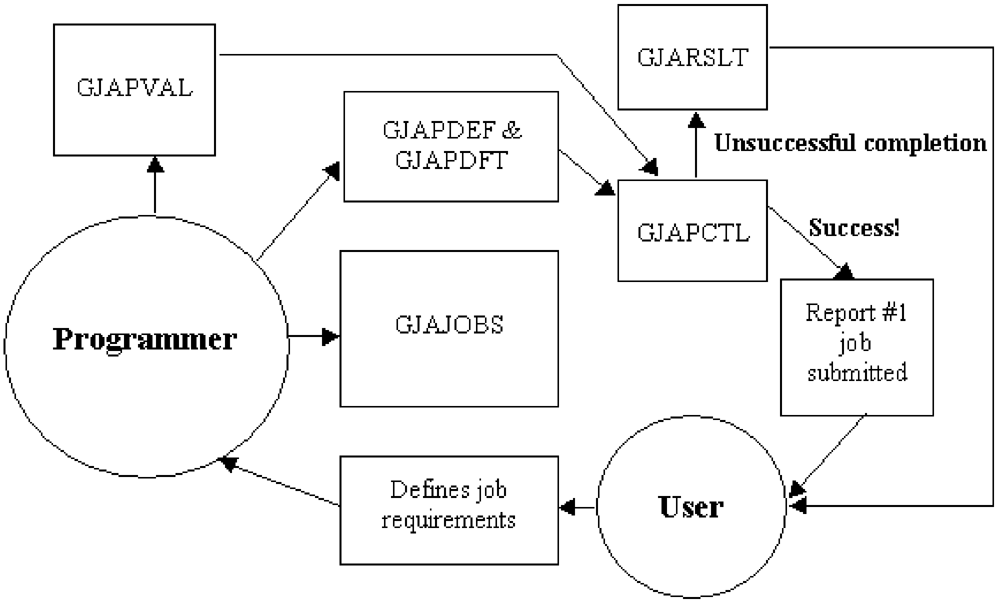
Job Submission at the Functional Level

Diagram



Job Submission at the Database Level

Diagram



Population Selection

Introduction

This chapter discusses the Population Selection process in Banner and its components.

Objectives

At the conclusion of this chapter, participants will be able to:

- identify the Population Selection process in Banner and its major steps
- view the Population Selection process flow

Population Selection Overview

Description

Population Selection allows you to identify and group entities in the database, such as people, vendors, or organizations.

Population Selection extracts a subset of IDs for use in Banner reports and letters, and allows processes to be run using that subset of IDs (such as people, vendors, or organizations). It is used for reporting purposes or for facilitating Letter Generation.

How does population selection work?

It selects the PIDM* for individuals who meet the selection criteria:

```
SELECT  pidm
  FROM  table
 WHERE  criteria
```

*Primary Identification Master

Population Selection involves four basic steps:

- Identify an application
- Define the Population Selection criteria
- Extract the IDs that match the criteria
- Review the Population Selection

Step 1: Identify an application

Application

An application owns a grouping of Population Selections.

The rules established by the parent application will be inherited by the child objects.

- Examples: FINAID, GENERAL, RECRUITING, ADVANCEMENT

Although the application must be defined in the Application Rules table (GLRAPPL), the rules (criteria) at this level are optional.

Step 2: Define criteria

Example

Example: test_female

```
SELECT spbpers_pidm
  FROM spbpers, spriden
 WHERE spriden_pidm = spbpers_pidm
***AND spriden_change_ind IS NULL
   AND spbpers_state_code_birth = 'CA'
   AND spbpers_sex = 'F'
   AND spriden_entity_ind = 'P';
```

GLRSLCT

Population Selection Definition Rules table (GLRSLCT)

- Allows for entry of the specific criteria which distinguishes one population from another
- If the tables are joined by PIDM, the PIDM join is automatic
- If the tables are joined by additional columns, the conditions must be explicitly defined

When the form is exited, the process Parameter Selection Process (GLBPARM) compiles the SELECT statement.

Step 3: Extract the IDs

GLBDATA

Run the Population Selection Extract Process (GLBDATA) from Job Submission.

Population Selection Extract Process (GLBDATA) parameters:

Required?	Parameter	Description
✓	Selection Identifier 1	Enter in all capital letters.
	Selection Identifier 2	Used only for Union, Intersect, or Minus functionality
	New Selection Identifier	Used only for Union, Intersect, or Minus functionality
	Enter description of Selection	Used only for Union, Intersect, or Minus functionality
	Enter (U)nion, (I)ntersect, (M)inus	Used only for Union, Intersect, or Minus functionality
✓	Application	Enter in all capital letters.
✓	Creator ID	Enter in all capital letters.
	Detailed Execution Report	

Step 4: Review population selection

GLAEXTR

Population Selection Extract Data Form (GLAEXTR)

- Review the results of running Population Selection Extract Process (GLBDATA)
- Add other IDs not processed by GLBDATA
- Remove IDs processed by GLBDATA

Population Selection Process Flow

Process flow

Application definition

- GLIAPPL - Application Inquiry Form
- GLRAPPL - Application Rules table

Objects

- GLIOBJC
- GLIOBJT

Variable usage

Variable definitions

- GLRVRBL - Variable Definition Rules table

Population Selection rules

- GLRSLCT - Population Selection Definition Rules table

Manual population creation

- GLAEXTR – Population Selection Extract Data Form

View population

- GLIEXTR - Population Selection Extract Data Inquiry Form

Self Check

Directions

Use the information you have learned in this workbook to complete this self check activity.

Exercise

Write a population selection to extract students who have an EFC between 100 and 200. Extract the students and then run a report (one that's in baseline Banner) to print to the database the extracted students' full names and IDs.

Self Check – Answer Key

Exercise

Write a population selection to extract students who have an EFC between 100 and 200. Extract the students and then run a report (one that's in baseline Banner) to print to the database the extracted students' full names and IDs.

The report to run is the Basic Applicant Report (RORAPLT).

Letter Generation

Introduction

This chapter discusses the Letter Generation process in Banner and its components.

Objectives

At the conclusion of this chapter, participants will be able to:

- identify the Letter Generation process in Banner and its major steps
- create and track letters

Letter Generation Overview

Description

Letter Generation allows you to extract data from the Banner database and merge that data with text, then print the results and log what letter was created and for whom the letter was created.

Letter Generation allows you to extract data from Banner based on a given Population Selection, merge extracted data with text, print the results, and maintain a log of printed letters.

How does Letter Generation work?

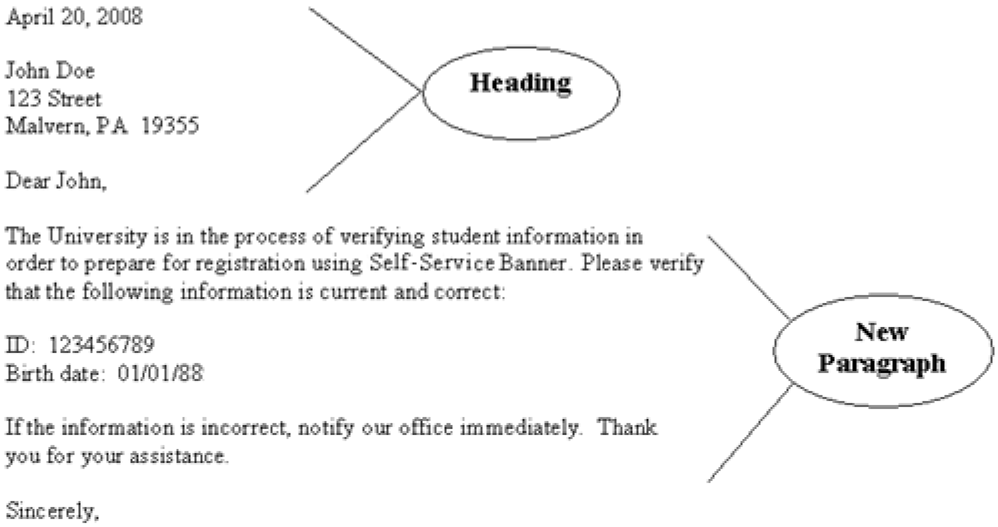
Letter Generation extracts specific data from the PIDMs which are extracted during a Population Selection.

```
SELECT spriden_first_name,
       spriden_last_name
  FROM  spriden
WHERE  pop_sel criteria
```

Next, it merges the extracted data with paragraphs customized for your implementation.

Example letter

Example



April 20, 2008

John Doe
123 Street
Malvern, PA 19355

Dear John,

The University is in the process of verifying student information in order to prepare for registration using Self-Service Banner. Please verify that the following information is current and correct:

ID: 123456789
Birth date: 01/01/88

If the information is incorrect, notify our office immediately. Thank you for your assistance.

Sincerely,

Steps to Create a Letter

Steps

1. Define variables to be used
2. Construct paragraphs
3. Combine paragraphs into letters
4. Extract population
5. Extract letter variables
6. Print letter or create merge file

Step 1: Define Variables

1st Paragraph

```
April 20, 2000          *DATE1

John Doe                *FUNC_PERSON_NAME
123 Street              *FUNC_PERSON_ADDRESS_LINE1 (LINE2, LINE3)
Malvern, PA  19355       *FUNC_PERSON_CITY_STATE_ZIP

Dear John,               *FUNC_PERSON_SALUTATION
```

GLRVRBL

Variable Rules Definition table (GLRVRBL)

- Builds selection criteria for data element
- Owned by an application
- Must have data associated with an ID

2nd paragraph

```
The University is in the process of verifying student information
in order to prepare for registration using Banner.  Please verify
that the following information is current and correct:

ID: 123456789           *ID
Birth date:  01/01/80   *BIRTH_DATE

If the information is incorrect, notify our office immediately.
Thank you for your assistance.

Sincerely,
```

Compilation

When the form is exited, the Parameter Selection Process (GLBPARM) compiles the SELECT statement.

Step 2: Construct Paragraphs

GTVPARA

Paragraph Code Validation Form (GTVPARA)

- Define codes and descriptions for paragraphs

GUAPARA

Letter Generation Paragraph Form (GUAPARA)

- Define the contents of each paragraph
- Paragraphs contain variables, formatting commands, and text
- Paragraphs for merge files contain only variables

Step 3: Combine Paragraphs

GTVLETR

Letter Code Validation Form (GTVLETR)

- Define codes and descriptions for letters

GUALETR

Letter Process Form (GUALETR)

- Combine paragraphs into a letter

Example letter – 1st paragraph

```
*DATE1
#nl
*FUNC_PERSON_NAME
#nl
*FUNC_PERSON_ADDRESS_LINE1 (LINE2, LINE3)
#nl
*FUNC_PERSON_CITY_STATE_ZIP
#nl
#nl
Dear
*FUNC_PERSON_SALUTATION
```

Example letter – 2nd paragraph

```
   #nl
   The University is in the process of verifying student information
in
   #nl
   order to prepare for registration using Banner.  Please
   #nl
   verify that the following information is current and correct:
   #nl
   #nl
   ID:
   #nl
   *ID
   #nl
   Birth date:
   #nl
   *BIRTH_DATE
   #nl
   If the information is incorrect, notify our office immediately.
   #nl
   Thank you for your assistance.
   #nl
   #nl
   Sincerely,
```

Example letter – merge file

```
   *DATE1
   *FUNC_PERSON_NAME
   *FUNC_PERSON_ADDRESS_LINE1 (LINE2, LINE3)
   *FUNC_PERSON_CITY_STATE_ZIP
   *FUNC_PERSON_SALUTATION
   *ID
   *BIRTH_DATE
```

Step 4: Extract a Population

Extraction requirement

A population must have been extracted prior to the next few steps in Letter Generation.

Step 5: Extract Letter Variables

GLBLSEL

Run the Letter Extract Process (GLBLSEL) from Job Submission.

This process extracts data from the database for selected PIDMs and stores it in a collector table.

Step 6: Print the Letter

GLRLETR

Submit the Letter Generation Print Report (GLRLETR) through the Process Submission Control Form (GJAPCTL).

- Removes the selected variables from the Collector table
- Merges that data with text specified in a letter
- Prints the letter or creates the appropriate word processing file

Creating a Letter

Required forms

Application Definition Rules Form (GLRAPPL)

- Create an application (only once)

Object Definition Rules Form (GLROBJT)

- Created once per object

Letter Code Validation Form (GTVLETR)

- Needed for every letter

Variable Definition Rules Form (GLRVRBL)

- Every variable has to be created (only once) *

Paragraph Code Validation Form (GTVPARA)

- Needed for every paragraph *

Letter Generation Paragraph Form (GUAPARA)

- Details needed for each paragraph

*Can be used for multiple letters in the same module

Tracking Letters

Determine how often you will send tracking letters (MIL's) to students and follow this procedure when you want to send them.

Steps

1. Extract the population selection you want to send tracking letters (MIL's) to using **GLBDATA**. The population you want should be built on **GLRSLCT.**
2. Run **GLBLSEL** to gather the appropriate variable information for your population selection.
3. Run **GLRLETR** to print the letter or extract the information to merge into a MS Word Document.
4. Select **Review Output** from the **Options** menu.
5. When the job is done, open .doc file in a browser.
6. Save the file into the folder where your Tracking Letter merge document is.
7. Make sure to save it as a .txt file.
8. Merge the letter.
9. Print the letter.
10. Save the merged letter into Processed Letters folder using date as name (i.e. 2005-03-24)

Don't forget to run **RLRLETR** to reset the Tracking Letter Indicator. This is an easy step to forget because you get busy printing letters and get sidetracked. Run this process as soon as you know your letters have merged successfully, before you start printing.

Conversions

Introduction

This chapter discusses the methods and considerations involved in converting Banner data.

Objectives

At the conclusions of this chapter, participants will be able to

- compare the pros and cons of conversion methods and relevant considerations
- identify the steps involved in a typical conversion
- view how various types of records and transcripts are affected by conversion.

Conversion Methods

Manual conversions

- Slower
- Data entry errors
- Small volume only

Electronic conversions

- Large volumes
- Requires development and testing

Conversion Considerations

Considerations

- Keeping track of PIDM on Legacy System
- What type of data you are bringing across
- Random ID# or SSN?
- Name/Address formatting
    - No number signs: "#"
    - No periods: "."
- Address types
- Multiple IDs on Legacy System?

Timing Considerations

New year conversion

- Recommended
- Starts January 1

Mid-year conversion

- Not recommended
- Each table must be populated to suggest Banner Financial Aid was running since the beginning of the aid year

Conversion Steps

Steps

1. Document
2. Review and clean current data
3. Determine scope (What will you convert?)
4. Map current data to Banner
5. Write a detail data plan
6. Review plan and get approval
7. Develop procedures and programs
8. Test
9. Create database
10. Perform conversion
11. Verify and correct
12. Test

*You will need your end users!

Financial Aid System Conversions

Conversions

- Persons
- Students
- Award history and transcripts
- Tracking requirements
- Refunds

Refer to Chapters 5, 6, and 8 of the *Financial Aid Technical Reference Manual*.

General Person Records

Records

- Identification table (SPRIDEN) - Person ID (required)
- Address table (SPRADDR)
- Telephone Number table (SPRTELE)
- Person table (SPBPERS) - Basic person
- Applicant Holds table (SPRHOLD) Holds
- Person email table (GOREMAL)

General Student Records

SGBSTDN

Student Base table (SGBSTDN) - (Required)

- Residency
- Academic status
- Educational goal

Other Student data

- Assistantship / Fellowship / Internship
- Educational opportunity
- Student disability
- Veteran status
- Sports information

Tracking Requirements

ROBINST

Financial Aid Institution table (ROBINST)

- Create one record for each aid year that you load
- Enter data manually through the Financial Aid Institution Options Form (ROAINST)

RTVTRST

Tracking Requirements Status Validation table (RTVTRST)

- All tracking requirement status codes must exist before you can load historic records

STVSBGI

Source/Background Institution Code Validation Form (STVSBGI)

- Source background institution codes must exist before you can load historic records with SBGI codes

RTVTREQ

Tracking Requirements Validation table (RTVTREQ)

- All tracking requirements codes must exist before you can load historic records

RTVTGRP

Tracking Group Validation table (RTVTGRP)

- All tracking group codes must exist before you can load historic records

Award History and Transcripts

Functional staff (tables)

- Financial Aid Institution table (ROBINST)
- Fund Base Data table (RFRBASE)
- Satisfactory Academic Progress Validation table (RTVSAPR)
- Award Status Validation table (RTVAWST)
  - Award Status codes must exist here
- Student Term Validation table (STVTERM)
  - Term codes must exist here before you can load historic records

Technical staff (tables)

- Fund Aid Year Specific Data table (RFRASPC)
  - Packaging and Disbursement uses aid year-specific fund codes
- Satisfactory Academic Progress Rules table (RORSAPR)
- Applicant Status Report (RORSTAT)
- Applicant Award table (RPRAWRD)
  - Create award records, one per aid year, per PIDM, per fund code
  - Enter award history via the Pre-Banner Award Summary Form (RHAPBAW)

Prior Institution Transcripts

Tables

Financial Aid Institution table (ROBINST)

Fund Base Data table (RFRBASE)

- If the prior institution award used fund codes, all fund codes must exist here.

Prior Institution Transcript Base table (RHBTRNS)

Prior Institution Transcript Award table (RHRAWRD)

- Prior institution awards/amounts are linked to the prior institution transcript record by PIDM, then by fund code, then by the Prior Institution Transcript Base table (RHBTRNS) sequence number.

Prior Institution Transcript Comment table (RHRPCMT)

- If you want to convert comments from prior institution transcripts, you must link them by PIDM, then by Prior Institution Transcript Base table (RHBTRNS) sequence number.
- Comments are not required.

Source/Background Institution Code Validation Form (STVSBGI)

- Source/background institution codes must exist before you can load historic records with SBGI codes.

Applicant Status table (RORSTAT)

- Create one record per aid year per student.

Prior Institution Transcript Prior College table (RHRPCOL)

- The background institution code is linked to the prior institution transcript record by PIDM, then by the Prior Institution Transcript Base table (RHBTRNS) sequence number.

Prior Institution Transcript Loan table (RHRLOAN)

- Prior institution loan awards/amounts are linked to the prior institution transcript record by PIDM, then by fund code, then by the Prior Institution Transcript Base table (RHBTRNS) sequence number.

Refund Owed and Default Records

Tables

Institution Transcript Information table (RHBTINF)

- This indicates if the student had defaulted on a Perkins or GSL, or if he or she owes a refund on Pell, SEOG or SSIG.

Applicant Status table (RORSTAT)

- You must create one record per aid year per student.

Financial Aid Institution table (ROBINST)

- Create one record for each aid year that you want to load via the Financial Aid Institution Options Form (ROAINST).

Conversion

Financial Aid Hold

- Applicant Status table (RORSTAT)
- Financial Aid Holds table (RORHOLD)
    - Financial Aid holds are linked to the student by PIDM by aid year by hold code
- Hold Type Validation Form (RTVHOLD)
    - Financial Aid hold codes must exist here before you can load historic records

Applicant Comments

- Applicant Status table (RORSTAT)
- Financial Aid Institution table (ROBINST)
- Financial Aid Institution Options Form (ROAINST)
- Applicants Comments (RHRCOMM)
    - Applicant comments are created by PIDM by user ID
    - Comments also require sequence number

Testing

Testing

- Use Banner forms to view inserted data
- Navigate through several forms
- User testing

Database Requirements

Required Tables for Clean instance (CMS-7978 FAQ)

ROBINST (for any aid years referenced in RORDATA, RORPELL, RPRFEDR)
RFRFFID
RNRGLBL
RORDATA
ROBLOGC
RORLOGC
RORDSUP
RORDVAL
RORAPEL
RORMVAL (As of 7.3.1)
RORPELL

RPRCIPC
RPRFEDR

RPRSAHP
RTVCDNT (As of 5.12/6.4)
RTVCDST (As of 5.12/6.4)
RTVICMT
RTVINFC (Code, Desc and Activity Date, only. The rest of the fields should be null, and must be null if constraints are applied.)
RTVPRCD (As of 4.14/5.5)
RTVRJCT
RTVPTYP
RTVDLBT
RTVPHAS
RTVYICD
RURVERS

Additional requirements

Additional Requirements to start up a clean instance (to be able to create the first ROAINST (ROBINST) record):

- RTVAPRD - Keep the 100% record, delete the rest **
- RTVBGRP - Keep the row for DEFALT or DEF, delete the rest **
- RTVPGRP - Keep the row for DEFALT or DEF, delete the rest **
- RTVTGRP - Keep the row for DEFALT or DEF, delete the rest **
- RTVSAPR - Keep the "X" row, delete the rest **
- RTVTRST - Keep the "S" and "E" rows, delete the rest **
- STVTERM - The current term must exist.

** Only one row needs to be kept in the table so that it can be associated with the ROAINST (ROBINST) record. If you choose to keep a different row, that is fine. You can always create additional rows in the tables later, change the values on ROAINST and delete the now obsolete "seed" rows. We have suggested the rows listed above for ease and consistency.

RORDATA & RORDSUP are delivered by SunGard Higher Education and should never be changed. They map the BANNER fields to the fields the central processor recognized for Need Analysis.

Cleaning Database

Cleaning database

Review CMS-7978 FAQ

Migration to Production

Migration to production

- Review the Migration to Production table in the Table Indexes section.
- Non-supported data, client needs to review their database and Banner installation

Troubleshooting

Introduction

This chapter discusses tips and tricks for troubleshooting certain problems while working with Banner Financial Aid.

Objectives

At the conclusion of this chapter, participants will be able to:

- build online group assignment rules using outer joins
- view tips for Financial Aid rules on forms
- improve performance of rules created on RORRULE

Building Online Group Assignment Rules Using Outer Joins

Incorrect rule

The following budgeting assignment rule would not function as intended if you were attempting to select a person into a particular budget group with the following characteristics:

- The person has a BANNER housing code that indicates the person is a dormitory resident, and
- The person is identified as an undergraduate in either the general student record or admission record

```
(RCRAPP1_CURR_REC_IND = 'Y' AND
RCRAPP1_INST_HOUS_CDE = '2' AND
(ROVST08_LEVL_CODE = 'UG' OR
ROVAD08_LEVL_CODE = 'UG'))
```

Explanation

The SunGard Higher Education views to the Student System, the Student View with Effective Date (ROVSTxx) and the Admissions Application View (ROVADxx), were specifically designed for use in Financial Aid rules forms. These views enable you to access data in the Student system without the need to specify term code joins.

The difficulty with the preceding rule is that when you commit and exit from the Financial Aid Selection Rules Form (RORRULE), the Automatic Rule Compilation (ROOGSQL) COBOL process adds the necessary PIDM and aid year joins for you behind the scenes. Therefore, the preceding rule becomes:

```
SELECT DISTINCT(RCRAPP1_PIDM)
FROM FAISMGR.RCRAPP1, BANINST1.ROVST08,
     BANINST1.ROVAD08
 WHERE (RCRAPP1_CURR_REC_IND = 'Y'
   AND RCRAPP1_INST_HOUS_CDE = '2'
   AND (ROVST08_LEVL_CODE = 'UG'
    OR ROVAD08_LEVL_CODE = 'UG'))
   AND ROVST08_PIDM =RCRAPP1_PIDM
   AND ROVAD08_PIDM =RCRAPP1_PIDM
   AND ROVST08_AIDY_CODE = RCRAPP1_AIDY_CODE
   AND ROVAD08_AIDY_CODE = RCRAPP1_AIDY_CODE
   AND RCRAPP1_AIDY_CODE = :AIDY
   AND RCRAPP1_PIDM =:PIDM
```

The additional PIDM and aid year joins require that a record exist in each view, not just one or the other, for this statement to be successful. Students must have records in both views to be selected; this was not the intention of the rule.

Outer join

This statement, however, can be easily modified and made to work using an outer join.

(An outer join enables you to make one table or view optional. If a row exists in that table or view, you use it; if it does not exist, the statement does not fail and continues without it. The "join" column of the optional table or view has the symbol "(+)" added to it to identify it as an outer join.)

In this case, add one more row to the preceding statement:

```
 (RCRAPP1_CURR_REC_IND = 'Y' AND
 RCRAPP1_INST_HOUS_CDE = '2' AND
 (ROVST08_LEVL_CODE = 'UG' OR
 ROVAD08_LEVL_CODE = 'UG')) AND    <--Added AND

ROVST08_PIDM (+) = RCRAPP1_PIDM  <--Additional row
```

Revised statement

The "compiled" statement becomes the following:

```
SELECT DISTINCT(RCRAPP1_PIDM)
  FROM FAISMGR.RCRAPP1, BANINST1.ROVST08,
       BANINST1.ROVAD08
 WHERE (RCRAPP1_CURR_REC_IND = 'Y'
   AND RCRAPP1_INST_HOUS_CDE = '2'
   AND (ROVST08_LEVL_CODE = 'UG'
    OR ROVAD08_LEVL_CODE = 'UG'))
   AND ROVST08_PIDM(+) = RCRAPP1_PIDM
   AND ROVST08_PIDM(+) =RCRAPP1_PIDM
   AND ROVAD08_PIDM =RCRAPP1_PIDM
   AND ROVST08_AIDY_CODE (+) = RCRAPP1_AIDY_CODE
   AND ROVAD08_AIDY_CODE = RCRAPP1_AIDY_CODE
   AND RCRAPP1_AIDY_CODE = :AIDY
   AND RCRAPP1_PIDM =:PIDM
```

Note that wherever ROVST08 is joined to another table or view, the outer join symbol (+) has been included by the ROOGSQL process.

This rule now requires that a record exist in ROVAD08, with or without a level code, and makes the ROVST08 record optional.

In other words, if both a general student record and an admissions record exist for this student, check if either has the level code set to "UG". If a general student record does not exist yet for this student, use the student's level on the admissions record and ignore the general student record references. If there is no admissions record, however, the statement fails and the student is either processed into another group or is assigned to the default group.

Tips for Financial Aid Rules on Forms

Tips

- If you reference any of the RCRAPPx tables in the rules that you write for Financial Aid forms, you do not need to include `RCRAPP1_CURR_REC_IND = 'Y'` on the form; it is automatically generated into the compiled rule by the ROOGSQL process when you commit and exit the form.

- You can only add an outer join to one table or view on the form.

- Use outer joins only on table/view to table/view joins. Therefore, only include the outer join symbol (+) on a single rule line which joins the key column of one table/view to the key column of another. This is considered a PIDM join when grouping rules. You should not use it on every line which accesses the outer joined tables.

- When you use outer joins on the Financial Aid rules forms, you must create the join line with the optional table/view on the left (for example,

  `ROVST08_PIDM (+) = RCRAPP1_PIDM`).

Hints on Improving Performance of Rules Created on RORRULE

Rule

RULE: If you are using RCRAPP2, RCRAPP3, RCRAPP4 or RCRESAR in your RORRULE rules, include the rule RCRAPP1_CURR_REC_IND='Y' as a rule on the form. To make it easier to see, we have placed it as the LAST rule on the form.

Including this rule will ensure that the correct RCRAPP1 index/constraint is used for performance purposes. If RCRAPP1 is the only RCRAPPx table that is used in your rule, you do not need to include it on the form.

The reason for the performance difference is the order in which ROOGSQL creates the "compiled rules" and, as a result, the RCRAPP1 index which is used at run time.

Indexes

RCRAPP1 has 2 indexes, RCRAPP1_CURR_INDEX - a nonunique index, and PK_RCRAPP1 - a unique, primary key constraint. The preferred index/constraint when using multiple RCRAPPx tables is PK_RCRAPP1.

RCRAPP1_ CURR_INDEX

RCRAPP1_CURR_INDEX (nonunique) is made up of:

```
1 RCRAPP1_AIDY_CIDE
2 RCRAPP1_PIDM
3 RCRAPP1_CURR_REC_IND
```

PK_RCRAPP1

PK_RCRAPP1 (primary key constraint) is made up of:

```
1 RCRAPP1_AIDY_CODE
2 RCRAPP1_PIDM
3 RCRAPP1_INFC_CODE
4 RCRAPP1_SEQ_NO
```

Rule created on RORRULE

When the rule is created on RORRULE as:

```
RCRAPP1_VERIFICATION_MSG = '1' AND
RCRAPP2_MODEL_CDE = 'D' AND
ROVST08_STYP_CODE = '9' AND
ROVST08_STST_CODE IN ('AS','SA') AND
ROVST08_LEVL_CODE = 'UG'
```

Compile order

The rule is "compiled" by ROOGSQL in the following order:

```
SELECT DISTINCT(RCRAPP1_PIDM)
  FROM RCRAPP1,RCRAPP2,ROVST08
 WHERE RCRAPP1_VERIFICATION_MSG = '1'
   AND RCRAPP2_MODEL_CDE = 'D'
   AND ROVST08_STYP_CODE = '9'
   AND ROVST08_STST_CODE IN ('AS','SA')
   AND ROVST08_LEVL_CODE = 'UG'
   AND RCRAPP2_PIDM =RCRAPP1_PIDM
   AND ROVST08_PIDM =RCRAPP1_PIDM
   AND RCRAPP2_INFC_CODE = RCRAPP1_INFC_CODE
   AND RCRAPP2_SEQ_NO = RCRAPP1_SEQ_NO
   AND RCRAPP1_CURR_REC_IND = 'Y'
   AND RCRAPP2_AIDY_CODE = RCRAPP1_AIDY_CODE
   AND ROVST08_AIDY_CODE = RCRAPP1_AIDY_CODE
   AND RCRAPP1_AIDY_CODE = :AIDY
   AND RCRAPP1_PIDM =:PIDM
```

RCRAPP1_ CURR_REC_ IND added

When RCRAPP1_CURR_REC_IND is added to the rule on RORRULE as follows:

```
RCRAPP1_VERIFICATION_MSG = '1' AND
RCRAPP2_MODEL_CDE = 'D' AND
ROVST08_STYP_CODE = '9' AND
ROVST08_STST_CODE IN ('AS','SA') AND
ROVST08_LEVL_CODE = 'UG'
RCRAPP1_CURR_REC_IND = 'Y' <====== HERE
```

Compile order

The "compiled" rule is created in this order:

```
SELECT DISTINCT(RCRAPP1_PIDM)
  FROM FAISMGR.RCRAPP1, FAISMGR.RCRAPP2,
       BANINST1.ROVST08
 WHERE RCRAPP1_VERIFICATION_MSG = '1'
   AND RCRAPP2_MODEL_CDE = 'D'
   AND ROVST08_STYP_CODE = '9'
   AND ROVST08_STST_CODE IN ('AS','SA')
   AND ROVST08_LEVL_CODE = 'UG'
   AND RCRAPP1_CURR_REC_IND = 'Y'
   AND RCRAPP2_PIDM =RCRAPP1_PIDM
   AND ROVST08_PIDM =RCRAPP1_PIDM
   AND RCRAPP2_INFC_CODE = RCRAPP1_INFC_CODE
   AND RCRAPP2_SEQ_NO = RCRAPP1_SEQ_NO
   AND RCRAPP2_AIDY_CODE = RCRAPP1_AIDY_CODE
   AND ROVST08_AIDY_CODE = RCRAPP1_AIDY_CODE
   AND RCRAPP1_AIDY_CODE = :AIDY
   AND RCRAPP1_PIDM =:PIDM
```

Notes

Since SQL processes from the bottom up, the rule WITHOUT the manual RCRAPP1_CURR_REC_IND='Y' rule finds the RCRAPP1 columns in the order shown in column 1 and uses the non-unique RCRAPP1_CURR_INDEX.

The column order of the rule WITH the manually added RCRAPP1_CURR_REC_IND is shown in column 2. Based on its order, it uses PK_RCRAPP1:

WITHOUT	**WITH**
1. RCRAPP1_PIDM	1. RCRAPP1_PIDM
2. RCRAPP1_AIDY_CODE	2. RCRAPP1_AIDY_CODE
3. RCRAPP1_CURR_REC_IND	3. RCRAPP1_SEQ_NO
4. RCRAPP1_SEQ_NO	4. RCRAPP1_INFC_CODE
5. RCRAPP1_INFC_CODE	5. RCRAPP1_CURR_REC_IND
(1-3: RCRAPP1_CURR_INDEX)	(1-4: PK_RCRAPP1)

Rule

RULE: If your rule will require PIDM and AIDY_CODE joins between RCRAPP1 and another RCRAPPx table, use an RCRAPPx column as the first column in your rule. This is necessary because the ROOGSQL compile program uses the first rule line's table as the driving table for additional joins.

RCRAPPx

When the rule is created on the form as:

```
RCRAPP2_MODEL_CDE = 'D' AND  <==== First rule line is
                                   RCRAPPx column
RCRAPP1_VERIFICATION_MSG = '1' AND
RCRAPP1 CURR_REC_IND = 'Y' AND
ROVST08_LEVL_CODE = 'UG' AND
ROVST08_STYP_CODE = '9' AND
ROVST08_STST_CODE IN ('AS','SA)
```

Compiled rule

The "compiled" rule INCLUDES an RCRAPP1 to RCRAPP2 PIDM join:

```
SELECT DISTINCT(RCRAPP2_PIDM)
  FROM RCRAPP2,RCRAPP1,ROVST08
 WHERE RCRAPP2_MODEL_CDE = 'D'
   AND RCRAPP1_VERIFICATION_MSG = '1'
   AND RCRAPP1_CURR_REC_IND = 'Y'
   AND ROVST08_LEVL_CODE = 'UG'
   AND ROVST08_STYP_CODE = '9' = 'UG'
   AND ROVST08_STST_CODE IN ('AS','SA')
   AND RCRAPP1_PIDM =RCRAPP2_PIDM    <== RCRAPPx
   AND ROVST08_PIDM =RCRAPP2_PIDM        PIDM join
AND RCRAPP2_INFC_CODE = RCRAPP1_INFC_CODE
AND RCRAPP2_SEQ_NO = RCRAPP1_SEQ_NO
AND RCRAPP1_AIDY_CODE = RCRAPP2_AIDY_CODE
AND ROVST08_AIDY_CODE = RCRAPP2_AIDY_CODE
AND RCRAPP2_AIDY_CODE = :AIDY
AND RCRAPP2_PIDM =:PIDM
```

Non-RCRAPPx

However, when it is created in this order:

```
ROVST08_STYP_CODE = '9' AND <== First rule line is
                               Non-RCRAPPx column
ROVST08_STST_CODE IN ('AS','SA) AND
ROVST08_LEVL_CODE = 'UG' AND
RCRAPP2_MODEL_CDE = 'D' AND
RCRAPP1_VERIFICATION_MSG = '1' AND
RCRAPP1 CURR_REC_IND = 'Y'
```

Compiled code

The resulting "compiled" code DOES NOT INCLUDE either the RCRAPP1 to RCRAPP2 PIDM join or the RCRAPP1 to RCRAPP2 AIDY_CODE join:

```
SELECT DISTINCT(ROVST08_PIDM)
FROM ROVST08,RCRAPP2,RCRAPP1
WHERE ROVST08_STYP_CODE = '9'
AND ROVST08_STST_CODE IN ('AS','SA')
AND ROVST08_LEVL_CODE = 'UG'
AND RCRAPP2_MODEL_CDE = 'D'
AND RCRAPP1_VERIFICATION_MSG = '1'
AND RCRAPP1_CURR_REC_IND = 'Y'
AND RCRAPP2_PIDM =ROVST08_PIDM
AND RCRAPP1_PIDM =ROVST08_PIDM
AND RCRAPP2_INFC_CODE = RCRAPP1_INFC_CODE
AND RCRAPP2_SEQ_NO = RCRAPP1_SEQ_NO
AND RCRAPP2_AIDY_CODE = ROVST08_AIDY_CODE
AND RCRAPP1_AIDY_CODE = ROVST08_AIDY_CODE
AND ROVST08_AIDY_CODE = :AIDY
AND ROVST08_PIDM =:PIDM
```

Banner Help Resources

Introduction

This chapter discusses the discussion lists available as help resources for Banner Financial Aid.

Objectives

At the conclusion of this chapter, participants will be able to

- identify Banner discussion lists
- subscribe to Banner discussion lists.

Banner Discussion Lists

Discussion lists

SunGard Higher Education supports a number of useful discussion lists:

- boracle - Oracle issues
- bfinaid - Financial Aid product issues
- bgeneral - General product issues
- bstudent – Student product issues
- breport - Reporting issues
- btrain - Training issues
- bpost - Electronic distribution announcements

List of lists

For a listing of SunGard Higher Education's discussion lists, send an email message to:
`listserv@sungardhe.com`

- No subject
- Message body:
  - LISTS

For additional discussion lists, try:
`listproc@sungardhe.com`

Subscribing

Send mail to listserv@sungardhe.com (or listproc)

- Leave subject field blank
- In message body:
  - `subscribe listname YourFirstName YourLastName`

You may subscribe to multiple lists in the same email, as long as each subscribe command is on a separate line.

Example:

```
     subscribe boracle John Doe
     subscribe bfinaid John Doe
```

Table Indexes

Introduction

This chapter contains lists of tables and their characteristics that are relevant to Banner Financial Aid.

Objectives

At the conclusion of this chapter, participants will be able to

- view Financial Aid rule and validation tables for migration from PPRD to PROD
- view Applicant Status Table information.

Financial Aid Rule and Validation Tables for Migration from PPRD to PROD (January 2006)

Table Name	Import Order	Rule / Validation Table?	Area	Description	Comments
RTVAPRD	Group 1	X	F	Aid Period Validation Table	
RTVBGRP	Group 1	X	B	Budget Group Validation Table	
RTVPGRP	Group 1	X	A	Packaging Group Validation Table	
RTVSAPR	Group 1	X	SAP	Satisfactory Academic Progress Validation Table	
RTVTGRP	Group 1	X	TRACK	Tracking Group Validation Table	
RTVTREQ	Group 1	X	TRACK	Tracking Requirements Validation Table	
RTVTRST	Group 1	X	TRACK	Tracking Requirements Status Validation Table	
RTVINFC	Group 2	X	EDE	Application Tape ID Validation Table	Code, Desc. & Activity Date
ROBINST	Group 3	X	OVER	Financial Aid Institution Table	
RORCAMP	Group 3	X	OVER	Financial Aid Campus Table	
RORCRHR	Group 3	X	OVER	Credit Hours Table	
RORCLVE	Group 3	X	OVER	Course/Student Level Exclusion Table	
RORUSER	Group 3	X	USER	User-Defined Variable Description Table	
RORWEBA	Group 3	X	OVER	Web Acceptance Table	
RTVARSC	Group 4	X	OR	Financial Aid Resource Codes	
RTVAUST	Group 4	X	SE	Employment Authorization Status Validation Table	
RTVAWST	Group 4	X	A	Award Status Validation Table	
RTVBTYP	Group 4	X	B	Budget Type Validation Table	
RTVCINT	Group 4	X	L	Certification Initials Validation Table	
RTVCOMP	Group 4	X	B	Budget Component Validation Table	
RTVFSRC	Group 4	X	F	Fund Source Validation Table	
RTVFTYP	Group 4	X	F	Fund Type Validation Table	
RTVHOLD	Group 4	X	HOLDS	Financial Aid Hold Type Validation Table	
RTVICMT	Group 4	X	OVER	ISIR Comment Code Text Validation Table	Delivered
RTVLNST	Group 4	X	L	Loan Status Validation Table	
RTVMESG	Group 4	X	MESG	Message Validation Table	
RTVRFST	Group 4	X	SE	Referral Status Validation Table	

Table Name	Import Order	Rule / Validation Table?	Area	Description	Comments
RNRGLBL		X	N	INAS Global Policy Options Table	
ROBLOGC		X	OVER	Logging Control Base Table	
RORDATA		X	OVER	Data Dictionary Table	Delivered
RORAPEL		X	OVER	Pell Grant Alternate Payment Schedule Table	Delivered
RORPELL		X	OTHER	PELL Grant Payment Schedule Table	Delivered
RORLOGC		X	OVER	Logging Control Repeating Table	
RPRFEDR		X	OVER	Federal Rules Table	Delivered
RBRGMSG		X	B	Budget Group Message Rules Table	
RRRTMSG		X	TRACK	Tracking Requirement Message Rules Table	
RFRBCMP		X	F	Fund / Budget Component Table	
RFRBASE		X	F	Fund Base Data Table	
RCRAIDY		X	OVER	Telephone Financial Aid Award Year Table defines the award years that are available for telephone.	
RCRTERM		X	OVER	Telephone Financial Aid Term Table defines the terms that are available for telephone.	
RPRCLSS		X	OVER	Class Code Translation Table	
RPRCLVL		X	OVER	Campus Undergraduate Level Code Rules Table	
RPRDATE		X	OVER	Enrollment Options by Term Table	
RORINID		X	OVER	Financial Aid College Codes Table	
RBRGTYP		X	B	Budget Group/Budget Type Rules Table	
RBRCOMP		X	B	Budget Component Rules Table	
RRRGREQ		X	TRACK	Tracking Group/Requirements Rules Table	
RJBJOBT		X	SE	Job Title Table	
RJBPLBD		X	SE	Placement Base Data Table	
RJBSEDR		X	SE	Student Employment Default Rules Table	
RJRPAYL		X	SE	Payroll Load Rules Table	
RJRPLRL		X	SE	Placement Rules Table	
RORGDAT		X	RULES	Group Selection Rules Base Table	
RORGSQL		X	RULES	Generated SQL Selection Identifier Table	
RORCMPL		X	RULES	Selection Identifier Compiled Statements	

Table Name	Import Order	Rule / Validation Table?	Area	Description	Comments
ROSSLCT		Sequence	RULES	Rules Sequence Generator	Remember to re-set, General TRM, Ch 6 (Ch 5 in Banner 6), Financial Aid - Migration to Production, Sequence Generator Maintenance when moving RORRULE's (RORGDAT, RORGSQL, RORCMPL) Look at finaid/plus/rosslct.sql
RCRINFR		X	EDE	Tape Matching Rules Table	not used in 7.X
RCRTPTR		X	EDE	Financial Aid Tape Translation Table	
RCRNAME		X	EDE	First Name Translation Rule Table	not used in 7.X
RPBLNDR		X	L	Lender Base Data Table	
RPBLNDA		X	L	Lenders and Disbursing Agents Base Table	
RPBLPRD		X	L	Loan Period Base Data Table	
RPBLOPT		X	L	Loan Options Table	Use for DL and FFELP
RORTPRD		X	F	Aid Period/Term Code Table	
RFRDEFA		X	F	Default Fund Award Schedule Rules Table	
RFRDEFD		X	F	Default Fund Disbursement Schedule Rules Table	
RFRDSCH		X	F	Fund Disbursement Schedule Rules Table	
RFRDLCK		X	F	Fund Disbursement Lock Table	
RFRASCH		X	F	Fund Award Schedule Rules Table	
RFRCOMM		X	F	Fund Comments	
RPRLDSB		X	L	Loan Period / Disbursement Schedule Rules Table	
RFRASPC		X	F	Fund Aid Year Specific Data	
RFRMESG		X	F	Fund / Message Assignment Rules Table	May require GTVLETR
RFRTREQ				Fund / Requirements Assignment Rules Table	
RPRLPAP		X	L	Loan Period / Aid Period Rules Table	
RPBOPTS		X	OVER	Packaging Options Table	
RFRDETC		X	F	Fund / Detail Code Rules Table	
RFRPROM		X	F	Promissory Note Requirement Table	
RPRLTRM		X	L	Loan Period / Term Schedule Rules Table	
RPRFSRC		X	A	Maximum Amount Per Fund Source	
RPRDEFE		X	A	Default Equity Packaging Rules Table	

Table Name	Import Order	Rule / Validation Table?	Area	Description	Comments
RPRDEFG		X	A	Default Gap Packaging Rules Table	
RPRGEQY		X	A	Packaging Group Equity Packaging Rules Table	
RPRGSHP		X	A	Packaging Group Self-Help Packaging Rules Table	
RPRGGAP		X	A	Packaging Group Gap Packaging Rules Table	
RPRDEFS		X	A	Default Self-Help Packaging Rules Table	
RPRGFND		X	A	Packaging Group/Fund Rules Table	
RPREXPT		X	A	Exemption Rules Table	
RPRCONT		X	A	Third Party Contract Rules Table	
RPRFUND		X	A	Packaging Fund Total Table	
RORAUDT			OVER	Audit Grading Mode Table	
RORSTRF		X	SAP	SAP Translation Rules Table	
RHBTRNS		X	H&T	Prior Institution Transcript Base Table	Covered with NSLDS
RHRFUND		X	H&T	Transcript Rules Table	Covered with NSLDS
RFRFFID		X	OVER	Federal Fund Id Table	Delivered
RORCOLS		X	OVER	All Financial Aid columns Table	Delivered
RORDSUP		X	OVER	Supplemental Log Data	Delivered
RORDVAL		X	OVER	Audit Log Form/Data Code Table	Delivered
RPRDEFG		X	A	Default Gap Packaging Rules Table	
RPRGEQY		X	A	Packaging Group Equity Packaging Rules Table	
RPRGSHP		X	A	Packaging Group Self-Help Packaging Rules Table	
RPRGGAP		X	A	Packaging Group Gap Packaging Rules Table	
RPRDEFS		X	A	Default Self-Help Packaging Rules Table	
RPRGFND		X	A	Packaging Group/Fund Rules Table	
RPREXPT		X	A	Exemption Rules Table	
RPRCONT		X	A	Third Party Contract Rules Table	
RPRFUND		X	A	Packaging Fund Total Table	

Table Name	Import Order	Rule / Validation Table?	Area	Description	Comments
RPRDEFG		X	A	Default Gap Packaging Rules Table	
RPRGEQY		X	A	Packaging Group Equity Packaging Rules Table	
RPRGSHP		X	A	Packaging Group Self-Help Packaging Rules Table	
RPRGGAP		X	A	Packaging Group Gap Packaging Rules Table	
RPRDEFS		X	A	Default Self-Help Packaging Rules Table	
RPRGFND		X	A	Packaging Group/Fund Rules Table	
RPREXPT		X	A	Exemption Rules Table	
RPRCONT		X	A	Third Party Contract Rules Table	
RPRFUND		X	A	Packaging Fund Total Table	
RORAUDT			OVER	Audit Grading Mode Table	
RORSTRF		X	SAP	SAP Translation Rules Table	
RHBTRNS		X	H&T	Prior Institution Transcript Base Table	Covered with NSLDS
RHRFUND		X	H&T	Transcript Rules Table	Covered with NSLDS
RFRFFID		X	OVER	Federal Fund Id Table	Delivered
RORCOLS		X	OVER	All Financial Aid columns Table	Delivered
RORDSUP		X	OVER	Supplemental Log Data	Delivered
RORDVAL		X	OVER	Audit Log Form/Data Code Table	Delivered
RPRDEFG		X	A	Default Gap Packaging Rules Table	
RPRGEQY		X	A	Packaging Group Equity Packaging Rules Table	
RPRGSHP		X	A	Packaging Group Self-Help Packaging Rules Table	
RPRGGAP		X	A	Packaging Group Gap Packaging Rules Table	
RPRDEFS		X	A	Default Self-Help Packaging Rules Table	
RPRGFND		X	A	Packaging Group/Fund Rules Table	
RPREXPT		X	A	Exemption Rules Table	
RPRCONT		X	A	Third Party Contract Rules Table	
RPRFUND		X	A	Packaging Fund Total Table	

Parameter Sets Population Selections, Letter Generation and Job Submission Parameter Sets

Table Name	Import Order	Rule/ Validation Table?	Area	Description	Comments
GLBSLCT		X	PS	Population Selection Base Table	Application Code is FINAID
GLRSFRM		X	PS	Population Selection Select Table	
GLRSLCT		X	PS	Population Selection Rules Table	

Variables

Table Name	Import Order	Rule/ Validation Table?	Area	Description	Comments
GLBVRBL		X	VAR	Letter Generation Variable Base Table	Application Code is FINAID
GLRVFRM		X	VAR	Letter Generation Variable Select Table	
GLRVRBL		X	VAR	Letter Generation Variable Rules Table	

Letters

Table Name	Import Order	Rule/ Validation Table?	Area	Description	Comments
GTVLETR		X	LTR	Letter Code Validation Table	Begin with FA_
GTVPARA		X	LTR	Paragraph Code Validation Table	
GUBPARA		X	LTR	Letter Paragraph Table	
GUBLETR		X	LTR	Paragraphs within a Letter	

Job Submission parameter sets

Table Name	Import Order	Rule/ Validation Table?	Area	Description	Comments
GJBPDFT		X	JS	Default Parameter Table	Specific User ID
GJRJPRM		X	JS	Job Parameter Set Rule Table	

Applicant Status Table (Base Financial Aid Table - RORSTAT)

Condition

Records must exist in this table for each aid year the student has data converted.

Column	Null OK?	Data Type	Length	Constraint	Comments
RORSTAT_AIDY_CODE	N	VARCHAR2	4	FK1_RORSTAT_ INV_ROBINST_CODE	AID YEAR CODE: The aid year to be associated with information in this record. EXAMPLES: 9091, 9192, 9293, etc.
RORSTAT_PIDM	N	NUMBER	22		PIDM: The internal system student identification number.
RORSTAT_LOCK_IND	N	VARCHAR2	1		LOCK INDICATOR: This field indicates whether the students financial aid application record is locked.
RORSTAT_PGRP_LOCK_IND	N	VARCHAR2	1		PACKAGE LOCK INDICATOR: This field indicates whether a lock exists on the students' award package.
RORSTAT_BGRP_PROC_IND	N	VARCHAR2	1		BUDGET GROUP PROCESSING INDICATOR: Y/N indicates waiting for batch budget group assignment.
RORSTAT_PGRP_PROC_IND	N	VARCHAR2	1		PACKAGING GROUP PROCESSING INDICATOR: Y/N indicates waiting for batch packaging group assignment.
RORSTAT_TGRP_PROC_IND	N	VARCHAR2	1		TRACKING GROUP PROCESSING INDICATOR: Y/N indicates waiting for batch tracking group assignment.
RORSTAT_PCKG_FUND_PROC_IND	N	VARCHAR2	1		PACKAGING PROCESSING INDICATOR: Y/N indicates waiting for batch packaging process.
RORSTAT_PCKG_PROC_IND	N	VARCHAR2	1		No longer used.
RORSTAT_PCKG_STATUS_IND	N	VARCHAR2	1		No longer used.
RORSTAT_AUTH_DISB_AID_IND	N	VARCHAR2	1		AUTHORIZE/DISBURSE AVAILABLE AID INDICATOR: Y/N indicates waiting for batch disbursement process.

Column	Null OK?	Data Type	Length	Constraint	Comments
RORSTAT_VER_COMPLETE	N	VARCHAR2	1		VERIFICATION COMPLETE: This field specifies whether the verification has been completed.
RORSTAT_USER_ID	N	VARCHAR2	30		USER ID: The logon ID of the user who entered or last updated the record in this table.
RORSTAT_ACTIVITY_DATE	N	DATE	7		ACTIVITY DATE: The date that information for this record in the table was entered or last updated
RORSTAT_VER_PAY_IND	Y	VARCHAR2	1		
RORSTAT_BGRP_CODE	Y	VARCHAR2	6	FK1_RORSTAT_INV_RTVBGRP_CODE	BUDGET GROUP: The student's assigned budget group.
RORSTAT_APRD_CODE	Y	VARCHAR2	6	FK1_RORSTAT_INV_RTVAPRD_CODE	AID PERIOD CODE: The Aid Period to which the applicant is currently assigned.
RORSTAT_PGRP_CODE	Y	VARCHAR2	6	FK1_RORSTAT_INV_RTVPGRP_CODE	PACKAGING GROUP CODE: This field indicates the packaging group code.
RORSTAT_TGRP_CODE	Y	VARCHAR2	6	FK1_RORSTAT_INV_RTVTGRP_CODE	TRACKING GROUP CODE: This field maintains the tracking group assigned to the student in the record.

Applicant Status Table (Base Financial Aid Table - RORSTAT)

Column	Null OK ?	Data Type	Length	Constraint	Comments
RORSTAT_RECALC_NA_IND	Y	VARCHAR2	1		RECALCULATE NEED ANALYSIS INDICATOR: This field indicates whether the need analysis needs to be recalculated. If the indicator is set, then the INAS will be run for this applicant in the next batch run.
RORSTAT_PC	Y	NUMBER	22		PARENTAL CONTRIBUTION: Only maintained by the RORCALC process. Will not be maintained in the future.
RORSTAT_SC	Y	NUMBER	22		STUDENT CONTRIBUTION: Only maintained by the RORCALC process. Will not be maintained in the future.
RORSTAT_APPL_RCVD_DATE	Y	DATE	7		APPLICATION RECEIVED DATE: The date on which the applicant's financial aid application was received, processed or created, depending on the institutional option.
RORSTAT_PCKG_FUND_DATE	Y	DATE	7		PACKAGE FUND DATE: The date that the package fund was assigned.
RORSTAT_NEED_DATE	Y	DATE	7		NEED ANALYSIS DATE: The date the last need analysis calculation was performed.
RORSTAT_PCKG_COMP_DATE	Y	DATE	7		PACKAGING COMPLETE DATE: The date that the student's financial aid package was complete.
RORSTAT_DISB_DATE	Y	DATE	7		No longer used.
RORSTAT_ALL_REQ_COMP_DATE	Y	DATE	7		ALL REQUIREMENTS COMPLETE DATE: The date on which all Tracking requirements for this applicant were satisfied.
RORSTAT_MEMO_REQ_COMP_DATE	Y	DATE	7		MEMO REQUIREMENTS COMPLETE DATE: The date that all requirements to allow memo of aid were met for this applicant.

Column	Null OK?	Data Type	Length	Constraint	Comments
RORSTAT_PCKG_REQ_COMP_DATE	Y	DATE	7		PACKAGING REQUIREMENTS COMPLETE DATE: This field indicates the date that all tracking requirements have been satisfied for packaging.
RORSTAT_DISB_REQ_COMP_DATE	Y	DATE	7		DISB REQUIREMENTS COMPLETE DATE: The date that all disbursement requirements were met for the particular applicant.
RORSTAT_AUTH_DISB_AID_DATE	Y	DATE	7		No longer used.
RORSTAT_AUTH_DISB_AID_TERM	Y	VARCHAR2	6		AUTHORIZE DISBURSE AID TERM: The term the last disbursement process was completed to authorize or disburse aid.
RORSTAT_PELL_SCHED_AWD	Y	NUMBER	22		PELL SCHEDULED AWARD: This field indicates the scheduled Pell Grant Award for the aid year.
RORSTAT_PRI_SAR_PGI	Y	NUMBER	22		PRIM SAR PGI: This field indicates the primary SAR pell grant index.
RORSTAT_SEC_SAR_PGI	Y	NUMBER	22		SECONDARY SAR PELL GRANT INDEX: This field maintains the secondary pell grant index from the student SAR.
RORSTAT_SAR_DATE	Y	DATE	7		STUDENT AID REPORT COMPLETED DATE: This field identifies the date that the SAR is complete and ready for payment.
RORSTAT_SAR_TRAN_NO	Y	VARCHAR2	2		STUDENT AID REPORT TRANSACTION NUMBER: This field maintains the SAR transaction number associated with the record.
RORSTAT_SAR_SSN	Y	VARCHAR2	9		STUDENT AID REPORT SSN: This field maintains the original SAR social security number portion of the SAR identification number.
RORSTAT_SAR_INIT	Y	VARCHAR2	2		STUDENT AID REPORT INITIALS: The original SAR initials as a part of the SAR identification number.
RORSTAT_MAX_YRS_ELIG	Y	NUMBER	22		No longer used.
RORSTAT_NO_YRS_USED	Y	NUMBER	22		No longer used.
RORSTAT_CM_SC_LOCK_IND	Y	VARCHAR2	1		CONGRESS. METHOD. STUDENT CONTRIB.LOCK INDICATOR: This field indicates if a lock exists on the congressional methodology student contribution.

Column	Null OK?	Data Type	Length	Constraint	Comments
RORSTAT_CM_PC_LOCK_IND	Y	VARCHAR2	1		CONGRESS.METHOD. PARENTAL CONTRIB.LOCK INDICATOR: This field indicates if a lock exists on the congressional methodology parental contribution.
RORSTAT_CM_TFC_LOCK_IND	Y	VARCHAR2	1		CONGRESS. METHOD. TOTAL FAMILY CONTRIB. LOCK INDICATOR: This field indicates if a lock exists on the congressional methodology total family contribution.
RORSTAT_PGI_LOCK_IND	Y	VARCHAR2	1		PELL GRANT INDEX LOCK INDICATOR: This field indicates if a lock exists on the pell grant index.
RORSTAT_SAR_PGI_LOCK_IND	Y	VARCHAR2	1		SAR PELL GRANT INDEX LOCK INDICATOR: Identifies if a lock exists on the SAR pell grant index.
RORSTAT_INST_SC_LOCK_IND	Y	VARCHAR2	1		INST METHOD. STUDENT CONTRIB. LOCK IND: Indicates if a lock exists on the institutional methodology student contribution.
RORSTAT_INST_PC_LOCK_IND	Y	VARCHAR2	1		INST METHOD PARENT CONTRIB. LOCK IND: Indicates if a lock exists on the institutional methodology parent contribution.
RORSTAT_SLCT_IND	Y	VARCHAR2	1		SELECT INDICATOR:Used by the Batch Group Process. Indicates whether the student has been processed already.
RORSTAT_GRS_NEED	Y	NUMBER	22		GROSS NEED: Only maintained by the RORCALC process. Will not be maintained in the future.
RORSTAT_UNMET_NEED	Y	NUMBER	22		UNMET NEED: Only maintained by the RORCALC process. Will not be maintained in the future.
RORSTAT_INAS_OPTION	Y	VARCHAR2	1		No longer used.
RORSTAT_TFC	Y	NUMBER	22		TOTAL FAMILY CONTRIBUTION: Only maintained by the RORCALC process. Will not be maintained in the future.

Column	Null OK?	Data Type	Length	Constraint	Comments
RORSTAT_AWD_LTR_IND	Y	VARCHAR2	1		AWARD LETTER INDICATOR: This indicates that an award letter should be generated for this student the next time award letters are printed.
RORSTAT_LAST_CYCLE_ELIGIBLE	Y	VARCHAR2	1		LAST CYCLE STUDENT ELIGIBLE FOR PELL: The last cycle that a student was eligible to receive PELL. 1 = 87-88, 2 = 88-89, 3 = 89-90, 4 = 90-91, 5 = 91-92, 6 = 92-93.
RORSTAT_RESOURCE_AMT	Y	NUMBER	22		RESOURCE AMOUNT: Only maintained by the RORCALC process. Will not be maintained in the future.
RORSTAT_RESOURCE_ACT_DATE	Y	DATE	7		RESOURCE ACTIVITY DATE: Only maintained by the RORCALC process. Will not be maintained in the future.
RORSTAT_TRK_LTR_IND	Y	VARCHAR2	1		TRACKING LETTER INDICATOR: Indicates that a tracking letter should be generated for this student the next time tracking letters are printed.
RORSTAT_IM_GRS_NEED	Y	NUMBER	22		IM GROSS NEED: Only maintained by the RORCALC process. Will not be maintained in the future.
RORSTAT_IM_UNMET_NEED	Y	NUMBER	22		IM UNMET NEED: Only maintained by the RORCALC process. Will not be maintained in the future.
RORSTAT_IM_TFC	Y	NUMBER	22		IM TOTAL FAMILY CONTRIBUTION: Only maintained by the RORCALC process. Will not be maintained in the future.
RORSTAT_ADDL_STFD_ELIG_IND	Y	VARCHAR2	1		ADDITIONAL STAFFORD ELIGIBILITY INDICATOR: Indicates if an applicant is eligibile for additional stafford awards.
RORSTAT_IM_LOCK	Y	VARCHAR2	1		IM LOCK INDICATOR: Indicates whether the IM data is locked from any type of update.
RORSTAT_FM_BATCH_LOCK	Y	VARCHAR2	1		FM BATCH LOCK INDICATOR: Indicates whether the FM data is locked from any updating by the batch tape load process.

Column	Null OK?	Data Type	Length	Constraint	Comments
RORSTAT_IM_BATCH_LOCK	Y	VARCHAR2	1		Indicates whether the IM data is locked from any updating by the batch tape load process.
RORSTAT_NSLDS_OVRD_IND	Y	VARCHAR2	1		NSLDS OVERRIDE: Indicates which validation tests Packaging/Disbursment processes should override for a specific Year/Student. L - limit tests (aggregate). D - default tests. O - overpayment tests. A – all tests.
RORSTAT_FORMER_HEAL_IND	Y	VARCHAR2	1		FORMER HEAL INDICATOR: If set to Y, this student qualifies for additional Stafford Unsub loans in replacement of the HEAL Loan program.
RORSTAT_PELL_LT_HALF_COA	Y	NUMBER	22		PELL LT HALF TIME COA: Student's PELL Cost of Attendance if enrolled for less than half time.
RORSTAT_INFO_ACCESS_IND	Y	VARCHAR2	1		INFORMATION ACCESS INDICATOR: A 'Y' or a NULL in this database field (represented by a checked check box) indicates that the award information for this applicant may be accessed via Voice Response, Banner Web and similar methods.
RORSTAT_BORROWER_BASED_CDE	Y	VARCHAR2	1		BORROWER BASED CODE: Indicates whether the student's loans need to be processed using a borrower-based academic year rather than the standard award year. This code is free form without validation.
RORSTAT_PELL_ORIG_IND	Y	VARCHAR2	1		CREATE PELL ORIGINATION INDICATOR: Identifier that indicates whether an updated Pell Origination Record is to be created automatically. Valid values Y, blank.
RORSTAT_PELL_DISB_LOCK_IND	Y	VARCHAR2	1		PELL DISB LOCK IND: If Y, indicates that all Disbursement activity for Pell is prevented due to RFMS Disbursement Reference numbers 91 - 99. Valid values Y, blank.
RORSTAT_POST_BA_PELL_OVRD	Y	VARCHAR2	1		POST BA PELL OVERRIDE INDICATOR: Flag to allow the Pell award and disbursement process to allow students to receive Pell grant funds even though they are Post Bachelor degree students.

Column	Null OK?	Data Type	Length	Constraint	Comments
RORSTAT_INFC_CODE	Y	VARCHAR2	7		INTERFACE DATA CODE: A code from the Interface Tape Table indicating the source and format of the application record.
RORSTAT_ALTERNATE_PELL_IND	Y	VARCHAR2	1		ALTERNATE PELL SCHEDULES INDICATOR: Indicator to identify applicants who are awarded Pell based on the Alternate Pell Schedules calculation. Valid values are Y or blank. Default is Null. Note: Alternate Pell has been disabled for 2007-8 and beyond due to the College Cost Reduction and Access Act.
RORSTAT_PELL_ATTEND_COST	Y	NUMBER	22		PELL COST OF ATTENDANCE: The Pell Cost of Attendance at the time the Pell calculation was last run. This will be used to determine the correct Pell schedules (Regular or Alternate) to use when awarding and disbursing Pell.
RORSTAT_LOW_TUITION_COST	Y	NUMBER	22		LOW TUITION AND FEES COST FOR PELL: The total amount of Tuition, Disability Expenses, and Dependent care costs that are calculated at the time the Pell process was last run.
RORSTAT_SUB_LOAN_EXCL_AMT	Y	NUMBER	22		SUBSIDIZED LOAN AMOUNT TO BE EXCLUDED: The amount of the resource that should be excluded when determining eligibility for a subsidized loan.
RORSTAT_DL_EPNOTE_IND	Y	VARCHAR2	1		DL WEB PROM NOTE INDICATOR: Indicator to identify student has completed an Electronic Promissory Note. Valid values are Y and blank. Default is null
RORSTAT_TGRP_CODE_LOCK_IND	Y	VARCHAR2	1		TRACK GROUP LOCK: Identifies if the Requirements Tracking group code is locked and prevented from further updates. Valid values are Y and N. Default is N.

Column	Null OK?	Data Type	Length	Constraint	Comments
RORSTAT_BGRP_CODE_LOCK_IND	Y	VARCHAR2	1		BUDGET GROUP LOCK: Identifies if the Budgeting group code is locked and prevented from further updates. Valid values are Y and N. Default is N.
RORSTAT_PGRP_CODE_LOCK_IND	Y	VARCHAR2	1		PACKAGE GROUP LOCK: Identifies if the Packaging group code is locked and prevented from further updates. Valid values are Y and N. Default is N.
RORSTAT_TRK_LTR_IND_DATE	Y	DATE	7		TRACKING LETTER FLAG INDICATOR CHANGE DATE: Identifies the date/time stamp of the most recent change to tracking letter flag
RORSTAT_TURN_OFF_PELL_IND	Y	VARCHAR2	1		INDICATOR TO TURN OFF PELL PROCESS: Student level indicator to prevent Pell process from automatically running from forms for this student. Set to Y to prevent.
RORSTAT_VER_USER_ID	Y	VARCHAR2	30		USER ID OF PERSON UPDATING VERIFICATION COMPLETE FLAG : UserID of person updating Verification Complete Flag.
RORSTAT_VER_DATE	Y	DATE	7		DATE VERIFICATION COMPLETE WAS UPDATED: Date Verification Complete Flag was updated.
RORSTAT_DATA_ORIGIN	Y	VARCHAR2	30		DATA ORIGIN: The latest source for the data contained in this record.

APIs

Introduction

This chapter provides a brief overview of APIs used in Banner Financial Aid

Objectives

At the conclusion of this chapter, participants will be able to identify APIs used in Banner Financial Aid.

APIs Used in Banner Financial Aid

Overview

Application Programming Interfaces (APIs) enhance processing and simplify code. The API is actually a program comprised of four different packages:

- a primary package containing query functions and create, update, and delete procedures
- a Rules package containing subprograms that support the primary package
- a Strings package containing error messages
- and an associated DML package that performs the actual database maintenance of the edited data.

In addition to the API packages, there are three supporting package types.

- *Utility* - Provides utility functions and procedures for the Banner APIs.
- *Validation* - Contains standard functions and procedures for a specific validation table.
- *Support* - Contains supporting programs that may be called by other APIs and error messages related to these edits.

The APIs are called by programs within Banner (for example, forms, C processes, self-service packages), and may be called by external systems that need to create, update, or delete information stored in Banner tables.

Various institutions requested that SunGard support school-specific validation logic with the APIs. The Banner APIs include connection points that you can use to interface with your own code to enable your validation logic without having to change baseline code.

For additional information, refer to the API Technical Documentation available on the Customer Support Center.

APIs Used in Banner Financial Aid

The following tables and forms use APIs to process data in Banner Financial Aid. The form listed next to the table in this chart is the representative source used to build the API validation and business rules. The APIs replace the corresponding code in the Banner forms. All APIs packages used in Banner Financial Aid will begin with an 'r' followed by a 'b' and an underscore '_,' followed by a verb/noun describing what the API does. If the 2nd letter is a 'p' instead of a 'b,' then that database package is a Business Process API defining a 'unit of work'.

Table	Form	API Object Name	API Entity Name	Task Performed
REBPAYV	REAORxx	rb_pell_ originations	PELL ORIGINATIONS	A Pell Origination is sent to COD to establish a record for each potential Pell Grant recipient. The origination also serves to verify the student's eligibility for the grant. The returned acknowledgement from COD is input for the purpose of updating the origination.
RERCODD	REICODD	rb_cod_ document	COD DOCUMENT	Use this API to insert a COD document record into the RERCODD table or to update receipt/response status codes and dates on the document ID. You can also update the document type code with this API.
RERSTID	REASTID	rb_cod_ identifier	COD IDENTIFIER	Use the functions and procedures in this API to create and update COD student and borrower identifier records. A record is identified as a student or parent record based on the identifier type. An identifier type of 'S' indicates a student record while a 'P' indicates a borrower record.

APIs Used in Banner Financial Aid, continued

Table	Form	API Object Name	API Entity Name	Task Performed
RPRADSB	RPAAWRD	rb_award_ disbursement	AWARD DISBURSEMENT	Contains the business logic (validation rules and other related processing) surrounding the table associated with Financial Aid Award Disbursement. Internal and external programs can insert/update/delete information into the Banner Applicant Award Disbursement Table (RPRADSB). For example, the Award Maintenance Form (RPAAWRD) calls this API to insert, update and delete entries to the RPRADSB Table. You must create the corresponding award record (RPRAWRD) and award by term record (RPRATRM) before you can create award disbursement information.
RPRRFND	RPALORG	rb_dl_refund	DL REFUND	Use this API to insert a Direct Loan Servicer Refund record into the RFRRFND table, update existing Direct Loan Servicer Refund records, and delete Direct Loan Servicer Refund records from RFRRFND.
RRRAREQ	RRAAREQ	rb_ requirement	APPLICANT REQUIREMENT	Use this API to create, retrieve, change, or remove an Applicant Requirement record. Summary requirement processing and other supporting processing is performed as needed.

There are several additional Financial Aid API's not mentioned here. These additional API's were created for entity processing, meaning they are not intended for use with external or third party systems. For additional information, refer to the API Technical Documentation available on the Customer Support Center.

The most valuable APIs for third party use are the business process APIs: RP_AWARD, RP_AWARD_SCHEDULE and RP_REQUIREMENT. These APIs handle everything necessary for adding and maintaining awards, award schedules and requirements and are used behind the RPAAWRD and RRAAREQ forms.