SUNGARD®
SCT • HIGHER EDUCATION

Banner Finance Conversion

And Interfaces Guide

Detailed Version

CONFIDENTIAL BUSINESS INFORMATION

****

****

Prepared For: BANNER Finance System Release 6.0
Prepared By: SunGard SCT
4 Country View Road
Malvern, PA 19355
Date Prepared: March 2004

Table of Contents

Banner Finance Conversions and Interfaces Guide

1. Introduction

This document is to serve as a guide to outline the key areas within Banner Finance that will need conversion and possible interfaces that may be necessary if Banner Finance is installed and implemented prior to the other products within Banner. This document is to be used as a supplement to the Banner Finance Technical Reference Manual Conversion Notes.

Included in this document is an outline of a general methodology for conversions, each area that is typically converted for Finance, and discussion about the typical interfaces. When looking at conversions, try to keep in mind the cost involved in performing the conversion programmatically versus entering the data directly into Banner Finance. Where will you get the most value for your time? Will it be more cost effective to write programs to convert the data, or more cost effective to directly enter the data into Banner Finance?

You will find below a list of the areas that are typically converted within Banner Finance and time estimates. These estimates do not include the time necessary to extract the data from legacy, only the time necessary to load the data from an extracted flat file or excel spread sheet and write the code (SQL, PL/SQL, SQL@Loader).

- Vendors (including Vendor 1099) – typically takes anywhere from 2-5 weeks, depending on the complexity of the conversion.
- Commodity Codes – 1 to 2 days
- Budget and Fund Balance (Balance Forward) – typically 1-2 weeks
- Revenue and Expense (if going live mid-fiscal year) – 5 to 10 days
- Fixed Assets – 1 to 2 weeks depending on the complexity of the conversion
- Legacy Student Balances to Banner Accounts Receivable (if bringing Student up at the same time). 3 – 10 days.
- Endowments – table and balances 5 to 10 days
- Grant inception to date summary balances – 5 to 10 days
- Grants themselves – 1 to 3 weeks depending on the complexity of the conversion
- Grant Billing – 280 hrs + weeks depending on complexity of accounting
- Chart Conversion Load (many times optional depending on size of data) – 1 week

Legacy areas that can be interfaced into Banner Finance if Finance goes live prior to the other Banner Products.

- Alumni Interface into Banner Finance as Journal Entries
- Payroll Interface into Banner Finance as Journal Entries
- Student Payment and Charges into Banner Finance as Journal Entries
- Student and/or General Refunds into Banner Finance as Invoices.

2. Finance Conversions

Again, please keep in mind the tradeoffs for each conversion. Will it be more cost effective to write a program to load the data, or will it be more cost effective to manually enter the data online.

2.1 Vendors

This process assumes that the Vendor ID does not duplicate any Person/Non-person ID's. If the ID were not unique then checks would have to be made on last name, first name, middle initial, maybe birth date, to further qualify that the person/corporation does not exist.

Vendors typically need to be in place by the go-live date of the Institution. A vendor load takes between 2-4 weeks depending on the cleanliness of the legacy data and the complexity of the load itself. One address, one telephone number keeps it simple. Multiple addresses and multiple phone numbers complicate the load.

2.1.1 Questions To Ask Yourself for a Vendor Conversion:

1. Will you only convert corporations? Or are you also bringing in persons as vendors.
2. How clean is your vendor data.
3. Should you not convert and spend the first couple months of going live just entering Vendors as you need them. Many clients do this as they have thousands of obsolete vendors and do not want to spend the time cleaning the data.
4. Will you have the ID of the Vendor generated, or bring in some pre-defined ID? Or will you have the legacy ID as alternate ID's in Banner?
5. How many addresses and or telephone numbers will you want to load? A pre-defined number or unlimited number. Many clients usually load just 1-2 addresses with associated phone numbers.
6. What are the pre-defined address types to use for vendors if not converting/crosswalking address types?
7. Will you be using Taxes?
8. Will you want to load 1099 rpt ids with the Vendor ID itself? What will be the default Income Type Sequence Number if no legacy value is supplied?
9. Will this vendor be classified as E-Procurement Vendor?
10. Will you be using currency conversion?
11. Will you want tax codes to default from the Vendor (Non-US Clients)?
12. Will you want discounts to default from the Vendor?
13. Will this be an in-state or out-of-state vendor?
14. What will you default address type code be for purchasing? Accounts Payable?
15. Will you want to load any special data into text for the vendor?

Generating a Vendor in Banner Finance will at the minimum create the following 4 records.

- SPRIDEN
- SPRADDR
- SPRTELE
- FTVVEND

Optional additional tables may be:

- GOREMAL: to load vendor email addresses
- GXRDIRD: to load the vendor Direct Deposit setup data
- SPBPERS: to load name prefixes or suffixes
- FTVVENT: to load vendor classification or categorization options
- FOBTEXT: to load text information

- FTRVTCL: to load the vendor tax codes

Generating a Vendor in Banner Finance requires that you have the following tables already populated:

- STVSTAT - State Table is already present and fully loaded
- STVCNTY - County Table is already present and fully loaded. (Optional)
- STVNATN - Nation Table is already present and fully loaded.
- STVATYP - Address Types are already present and fully loaded.
- STVTELE - Phone Types are already present and fully loaded.
- SOBSEQN Counters for PIDM and ID generation appropriately set before and after conversion.
- FTVITYP - Seed Data exists.
- GTVEMAL - Email Address Types if loading email addresses in GOREMAL (Optional)
- FTVVTYP - Vendor Type setup table (Optional)
- FTVTRAT - Tax Rate setup table (Optional)
- GXVDIRD - Bank Routing Numbers (Optional)

NOTES: General Person Module Tables

"Person" records include both real people and non-persons, which may be vendors, third party entities (contracts), etc. One must load at least one SPRIDEN per "Person"/"Entity" you will load. You will also want to create address and telephone records for each person/entity.

SPRIDEN contains both previous and current names and ID's. Be sure you understand the use of these data elements before proceeding. For the Vendor load, we can load prior name and ID information as alternate ID's or historical data.

System-generated PIDM's and ID's are controlled using the Sequence Number Table (SOBSEQN). The counters need to be set appropriately before and after conversion. ID's most likely are generated in the Vendor Conversion load where the legacy ID's are set as alternate ID's.

Telephone data is stored in a separate table from addresses, but may be linked to a specific address type and sequence number. FTMVEND form in Banner will automatically generate telephone records in SPRTELE even if there were no data entered in the telephone fields in the form.

- **SPRIDEN** - Person Identification/Name Repeating Table: Required for all "Persons" (people and non-persons). Unique PIDM must be created when loading this table. **All** records in the system are matched to the person using this PIDM. At least one SPRIDEN record, representing the current name and Id, must be loaded. Be sure to load all person-related data in all tables using the same PIDM.
- **SPRADDR** - Address Repeating Table: Not required, but most installations will convert at least some address data for their persons. Requires match on PIDM established when SPRIDEN's were loaded. Use legacy id's to crosswalk to PIDM's generated in SPRIDEN. The following is a definition of what fields are required in loading SPRADDR table:
  - Type of Address is required.
  - STREET LINE 1 and CITY are always required.
  - STAT_CODE and ZIP are required.
  - Nation should not be entered when it is "United States". Conversely, when the nation code is not

- "United States", it should always be loaded and crosswalked.

- **SPRTELE** - Telephone Table: Not required, but most installations will convert telephone data for each address. Requires match on PIDM established when SPRIDEN's were loaded. If telephone records are to be linked to an address, this requires additional link based on address type and sequence number. Use legacy id's to crosswalk to PIDMs generated in SPRIDEN. In most cases, you will want to associate a set of telephone data with each address. If telephones are to be linked to addresses, load one SPRTELE per SPRADDR unless you require loading two: one for the phone and the other for the fax. It is recommended that you always load at least 1 phone number for each address. One that matches the address type linked to the address type. Even if the phone number is not available so it can be used as a record "holder" for the phone number. This simulates what the Vendor Form does online whenever an address is created. Use the following definition of fields:

  Loading one record in SPRTELE per SPRADDR:
  - SPRTELE_CODE = SPRADDR_TELE_CODE
  - SPRTELE_ATYP_CODE = SPRADDR_ATYP_CODE
  - SPRTELE_ADDR_SEQNO = SPRADDR_SEQNO
  - SPRTELE_PRIMARY_IND = 'Y'

  Loading two records in SPRTELE per SPRADDR:
  Record (1):
  - SPRTELE_CODE = SPRADDR_TELE_CODE (Phone Record Type)
  - SPRTELE_ATYP_CODE = SPRADDR_ATYP_CODE
  - SPRTELE_ADDR_SEQNO = SPRADDR_SEQNO
  - SPRTELE_PRIMARY_IND = 'Y'

  Record (2):
  - SPRTELE_CODE = SPRADDR_TELE_CODE (FAX Record Type)
  - SPRTELE_ATYP_CODE = SPRADDR_ATYP_CODE
  - SPRTELE_ADDR_SEQNO = SPRADDR_SEQNO
  - SPRTELE_PRIMARY_IND = NULL

- **FTVVEND** - Vendor table where either persons or entities are defined. The person or entity is first defined in SPRIDEN with a generated PIDM. For this conversion, we are generating 1 SPRIDEN and 1 FTVVEND record for each Vendor defined in the legacy system. The Vendor ID MUST BE UNIQUE for all Vendors.

2.2 Vendor 1099 Data

Many clients that go live in July do not immediately load their 1099 data into Banner Finance by that date. You have 6 months.

2.2.1 Questions To Ask Yourself for a 1099 Conversion:

1. Will you be cutting your 1099's from your legacy system or from Banner Finance? If you want to cut your 1099's from Banner Finance, then you will need to ensure that all 1099 vendors have been loaded into Banner Finance from your legacy system along with its 1099 data.
2. Will you be loading detail legacy 1099 data or summary? It is preferred (and sometimes required) to load detail legacy 1099 data to allow users to verify, modify and correct reporting data before reporting 1099 data to the Federal Government.

Generating 1099 data means that the FAB1099 table is to be populated for each vendor that has 1099 related data.
Generating 1099 data in Banner Finance means that the FAB1099 table is populated along with key fields in the FTVVEND table.

- FTVVEND – update 1099 data
- FAB1099

2.3 Commodity Codes

Commodity Codes are sometimes converted. If you intend to convert commodity codes, minimum information is needed to populate the FTVCOMM and the optional FTRCOMM table. The following are the minimum required fields to populate FTVCOMM table:
- Commodity Code
- Commodity Description
- Unit of Measure to order in

2.3.1 Questions To Ask Yourself for a Commodity Conversion:

- Will some of the converted commodity codes to be used for Fixed Assets (as an asset commodity) or Stores Inventory (as stock item)?
- Will you want a tax group code default?
- Will you want to set a useful life default?
- If loading for Stores Inventory, what is the cost of the inventory?

2.4 Fund Balance (Balance Forward) Load

The Fund Balance Load Flat file is generated by the Institution either as an extract from their legacy system or usually entered by hand into an Excel Spreadsheet. The information needed is the final fund balance information as of year-end close from the legacy system. Some clients actually enter this information in by hand as Journal Vouchers into the system. The rule class code to use unless instructed otherwise from the client is:

JE05 Loads beginning balances into period 00 of the general ledger.
Requires FUND, ACCT, account must be a G/L account of either internal type of '10' for assets, '20' for liabilities, or '40' for fund balance. DO NOT LOAD CONTROL ACCOUNTS!

Better to use is a copy of the JE05 rule class code and call it JE6, with a balance indicator by (F)und instead of by (T)otal.

Everything is hard coded except for the Banner Fund, Banner Account, and Amount. The preference is to have the amount represented as the true balance. The rule code JE6 requires that the funds total by zero so that the Debits must equal the credits. The amount has to be an absolute value amount column with a "D" or "C" in the Debit/Credit column.

All funds should be loaded balanced. What does that mean. Assets must equal liabilities + fund balance. So if you add up all amounts for the fund, they should also total to zero. Scripts can be run up front to verify. Furthermore, Claim on Cash should total to zero for ALL FUNDS. Scripts can be run up front to verify.

To make it easier for your accountants, one can modify the rule class code to actually post automatically to the Claim on Cash account for each transaction line. The rule class code would have to be changed to (S)elf-balancing in this case. Thus it would look somewhat like this from a process standpoint.
I061 N
G011 N
G011 R with posting account modifier for Interfund Cash

Or another method to make it easier for your accountants, one can modify the rule class code to actually post automatically to the Claim on Cash account for each transaction. Just a different method to perform the same thing above. The rule class code would have to be changed to still be (S)elf balancing just like in the paragraph above. Thus it would look somewhat like this from a process standpoint.
I061 N
G011 N
G020 N

2.4.1 Questions To Ask Yourself for a Fund Balance Conversion:

There really are no questions other than when are you going to do this? The fund balance load does not have to be in place by go-live in Production. In actuality you won't be able to have it in place at that time as you will be finishing up year-end processing in your legacy system. However, you must plan a test load into TEST to ensure that scripts will run correctly. And you must load your fund balance into Production prior to running any month end reports. Thus you have approximately one

month after go-live to ensure that you fund balance is converted. You can wait later, however you will not get meaningful reports out of the system until this is done.

When loading fund balance, the GURFEED table is populated with the appropriate information. Many times a temporary GURFEED table is created and used to manipulate the data before populating the permanent GURFEED table.

2.5 Budget Load

The Budget Load Flat file is generated by the Institution either as an extract from their legacy system or usually entered by hand into an Excel Spreadsheet. The information needed is the budget for the new fiscal year. Some clients actually enter this information in by hand as Journal Vouchers into the system. The rule class code to use unless instructed otherwise from the client is:

BD01 Loads Adopted Budget into the operating ledger. Requires FUND, ORGN, ACCT, PROG, ACTV (optional), LOCN (Optional). The account code must be an Operating account of either internal type of '50' for revenue, '60' for labor, '70' for expenses, or '80' for transfers.

Everything is hard coded except for the Banner Fund, Banner Account, Banner Organization, Banner Program and Amount. The preference is to have the amount represented as the true balance. You can strip out the minus sign and create the Debit Credit column as a '+' or '-' for the load. You will also need to create an absolute value amount column that will be loaded.

Note: Do not forget to properly fill the BUDGET PERIOD column with the appropriate budget period ('01' for annual budget and so on).

2.5.1 Questions To Ask Yourself for a Budget Load:

There really are no questions other than when or are you going to do this? Will you be using Non-Sufficient Funds Processing with the Error Severity set to 'E'? If so, then you will want to have the budget definitely loaded by your go-live date. If not, do you still wish to have budget loaded into Production by go-live. Again, the budget load may affect your reports by month end so the same terms apply here as they do for the Fund Balance Load.

When loading budgets, the GURFEED table is populated with the appropriate information. Many times a temporary GURFEED table is created and used to manipulate the data before populating the permanent GURFEED table.

Many times clients will have their legacy chart information in the spreadsheet instead of the Banner Chart data. The legacy chart as the load can be accommodated by creating a crosswalk between Banner Finance Chart and Legacy Chart by loading the data into a table called FTVEELT. This table is then used to convert the legacy chart data into Banner Finance Chart data. It can also be viewed online by the end user in form FTVEELT. Another option is to use the Converter Tool's crosswalk table if you have purchased the converter tool. Still another option may be to look at using the Reporting Attributes function as a crosswalk.

2.6 Revenue and Expense

If the implementation date is not the beginning of a fiscal year, current year fiscal activity that has occurred in the legacy system must also be brought into the Banner Finance Ledgers. There are a number of options for bringing in current year transactions. Please review the options and work with your consultant to develop your site-specific conversion plan.

Rule Class to use on transactions
There are two baseline rule classes that will work for this processing, or a cloned rule can be developed that includes features from both. For either of the rule classes listed below, the actual cash activity in the cash accounts/bank fund will need to be explicitly entered.

JE15 – General Journal Entry (Intra-Fund)
This rule class requires that the sum of the debits and credits be equal at the fund level. It will post input lines for general ledger or operating ledger accounts, but will not perform any claim on cash or bank fund processing.

JE16 – General Journal Entry (Inter-Fund)
This rule class requires that the sum of the debits and credits be equal at the chart level. It will post input lines for general ledger or operating ledger accounts, and will also post an offset to the input fund's claim on cash account. It will not perform any bank fund or cash account processing.

A cloned rule could enforce balancing at the fund level and process the input fund's claim on cash activity. A cloned rule would also help visually identify this transaction as one that had been brought forward from the legacy system.

2.6.1 Questions To Ask Yourself for a Mid-Year Conversion:

Bring transactions forward in total or by separate accounting periods?
The institution will need to determine the importance of being able to identify which accounting period the transaction originally occurred. If this identification is important then each period should be brought in on a separate Journal Voucher, each having a transaction date that corresponds to the appropriate accounting period, typically the last day of the month. If the separate identification is not important, then a single Journal Voucher can be created using a transaction date of the last day of the last accounting period on the legacy system.

Use of Clearing Accounts for subsidiary ledger balance forwards
Certain subsidiary ledgers, such as Student Accounts Receivable, will require conversion of their outstanding information into the corresponding Banner module. In this case the method used to bring those transactions forward can affect how the General Ledger asset or liability account is handled during the Finance conversion. If transactions will be brought forward from both the accounting system and a subsidiary ledger, then use of a Clearing Account is recommended. If the transactions from the accounting ledgers equal the transactions from the subsidiary ledgers, then the Clearing Account will be zero. A non-zero balance will identify a problem that needs to be investigated and resolved.

Again the GURFEED table is populated to generate journal entries into Banner Finance.

2.7 Endowments

If a client has hundreds of endowments, it may be easier to load the endowment data into the FNVENDW table from an excel spreadsheet. Other supporting tables that need to be in place or loaded from spreadsheets are FNVPOOL (Pool codes identified), and FTVINCL (Income classifications identified. Additionally Endowment Types are set up in FTMSDAT to support new endowment types other than those delivered with SEED Data.

Before loading balances, make sure that your pool has the 6/30 unit value set up in form FNMUCTR – the value as of June 30th (or the day prior to your go-live date) is really the only one that matters.

At a minimum, you will need the following fields in your spreadsheet:

FNVENDW_COAS_CODE – Pooled Chart
FNVENDW_POOL_CODE – Pool Code
FNVENDW_FUND_CODE – Principal Endowment Fund Code (must already exist in FTVFUND)
FNVENDW_COAS_SPEND_CODE – Spendable Chart
FNVENDW_FUND_SPEND_CODE – Spendable Fund
FNVENDW_ENDW_TYPE – defined on FTVSDAT, original (QU)asi, Term and True endowments.
FNVENDW_REST_TYPE – FASB restriction indicator (U)nrestricted or (R)estricted
FNVENDW_INCL_CODE – income classification code from FTVINCL

Additional fields that could be loaded
FNVENDW_ORGN_CODE_SPEND – spendable organization
FNVENDW_ACCT_CODE_SPEND – spendable account
FNVENDW_PROG_CODE_SPEND – spendable program
FNVENDW_EST_DATE – date fund was established
FNVENDW_LONG_FUND_TITLE – Funds Long title
FNVENDW_INC_DIST_SPEC_INST – special instructions
FNVENDW_ALPHA_TITLE – Alpha title or short title

Effective date hard coded to your chart go-live date.
Next Change date hard coded to '31-DEC-2099'

NOTE: Unrestricted funds do not require income classification.

2.7.1 Questions To Ask Yourself Loading Endowments:

1. None other than to determine which funds are to be restricted and which funds unrestricted.

2.7.2 Endowment Summary Units and Balances

Endowment summary balances and units must be entered as a journal voucher in Banner Finance in order to populate the Endowment tables with summary data. The units and the amount both on the same transaction record. Clients typically enter Endowment Funds' summary revenue into the prior fiscal year prior to the go-live fiscal year. When doing this, one CANNOT utilize the options of loading Grant Inception to date balances AND performing a Fiscal Year End Roll. You can load Grant Inception to date balances, just No Year End Roll can take place if you are loading Endowment balances. The data is typically entered in with a June 30th date, or the last day of the

fiscal year prior to go-live fiscal year. The purpose is to record summary revenue and summary units into the endowment units tables to have the summary balances and units represented prior to the go-live date.

Prior to loading these journal entries in GURFEED, you must ensure that the endowment Fund Code itself has been set up in table FNVENDW table via the FNMENDW form. In order to populate GURFEED, read about populating GURFEED in the JV Interfaces section.

NOTE: Special consideration must be made to get the units into the journal voucher form and subsequently into the endowments module. The gurfeed table does not contain a gift units field. So this is one method to accomplish this function.

1. Make sure that you create a new system id in FTMSDAT to load endowments.
2. Populate gurfeed with the gift units in the gurfeed_doc_ref_num field.
3. Use the rule code CR05 with the transactions.
4. Set the balance to zero in the header record to force the document to error out and to be corrected in the journal voucher form. This gives you a chance to run an update script to move the units stored in fgbjvcd_doc_ref_num field into the fgbjvcd_emc_units field with a to_number function after you have run FGRTRNI.
5. Remove the data from fgbjvcd_doc_ref_num, otherwise it will get loaded as a Gift Number in the Endowments module.
6. When the fund code is defined as an Endowment fund, the Gift Date in fgbjvcd_gift_date is populated automatically with the transaction date by the fgrtrni process.
7. After updating fgbjvcd, re-complete the document from online by correcting the document amount and pressing the complete button.
8. Post the document. Then run the unitization process FNPUNTZ with the same date you loaded the transactions with.
9. Review the data in FNIEBAL to ensure that it reflects the data that was converted.
10. Create opening Fund Balances for Endowment to load with a July 1st date. These balances should be equal to the revenue balances you loaded with the June 30th date or that last day of your prior fiscal year. The opening balances probably need to be split into two components. Permanently Restricted F/B (Corpus) and Unrestricted F/B (undistributed gains and losses). You will normally use multiple F/B processing (By ACCT) with specific accounts going into the Corpus.

2.8 Fixed Assets

Fixed Assets consists of populating two tables, FFBMAST, the asset master table and FFRMASA the Capitalization/Depreciation Table. There are several key values that must be set in both of the tables. Look at the attached spreadsheet sent with this document to gain a good understanding of what is needed and what is not needed. Many times, the majority of the values can be hard coded. At a minimum, you will need in the spreadsheet:
Asset description
Asset Type
Amount
Date of Acquisition
Useful Life in Years

Each of the following values can either be hard coded if they are the same across all assets, or must be in the spreadsheet if the value could change among the assets.
Bank Code
Asset Account Code
Capitalization Fund Code
Equity Account Code
Accumulated Depreciation Account Code
Depreciation Expense Account code
Depreciation Expense Organization Code
Depreciation Expense Program Code
Depreciation Method Code
Depreciation Frequency
Depreciation Post Code
System Status Code (usually set to 'N' for non-procurement)

The origination tag (system tag), is generated by using the FOBSEQN table. The PTAG (permanent tag) can also be generated if desired or loaded from your legacy system, or generated in whatever fashion desired.

DEPRECIATION: One of the things you will need to ask yourself; are you going to load fixed assets with depreciation or load without depreciation and have the system perform catch-up depreciation? (No longer used for current clients as everyone should be using Depreciation.) The simplest method is to load fixed assets without depreciation and let the system calculate it. If you are loading Fixed Assets with depreciation several additional fields must be populated to accommodate proper accounting adjustments after the assets are loaded. I.E. If these additional fields are not populated, then you could Reverse the Capitalization of a PTAG even though depreciated, with an out of balance entry. The additional fields to be populated are listed as follows, and can be viewed in the Depreciation Data Flow diagram.

FFBMAST

Ffbmast_Last_Depr_Date
Ffbmast_Old_Last_Depr_Date
Ffbmast_Old_Rem_Life_Qty
Ffbmast_Adj_Est_Life_Qty
Ffbmast_Old_Adj_Est_Life_Qty

FFRMASA
Ffrmasa_Old_Net_Depr_Amt
Ffamasa_New_Net_Depr_Amt

2.8.1 Questions To Ask Yourself for a Fixed Assets Conversion:

1. Will the acquisition/purchase date represent your PTAG date? OTAG date? Capitalization Date? Depreciation Start Date.
2. Will all assets being loaded be represented as capitalized? What is your capitalization amount limit?
3. Will all assets loaded need to be depreciated?
4. Will you load assets with depreciation, or will you need to run catch-up depreciation. (Catch up depreciation should no longer be used.) Loading with depreciation is much more complicated than loading without depreciation and then running catch-up depreciation with the FFPDEPR process?
5. Will you want to load responsible organization and location codes?
6. Will you want to load information that is not represented in the FFBMAST table?

The minimum amount of data to be loaded will be in the following two tables:
FFBMAST – Asset Master
FFRMASA – Asset Capitalization/Depreciation Accounting Table

NOTE: Many times people mistakenly believe that the FFRMASF and FFRDEPR tables must be populated. This is not true. FFPDEPR represents depreciation controls that have changed. Thus all depreciation information is originally loaded into FFBMAST and FFRMASA. FFRMASF represents the Banner Funding Source information. Since these assets were not generated from within Banner Finance, you will not want to populate this table. If you try to do so, you may end up with your End User getting "stuck" in the form and not being able to get out. If you do load FFRMASA, then you must populate the both the FOAP and the amount together.

2.9 Legacy Student Balances to Banner Accounts Receivable

This process assumes that the students have been set up in the SPRIDEN and SPBPERS tables and have been assigned a PIDM. In other words, Student is going live at the same time Finance is, or Student has gone live prior to Finance.

After data is loaded into table TBRACCD, and the appropriate cashiering sessions have been loaded into TBBCSHR, application of payments is run (TGRAPPL), followed by TGRFEED that feeds the A/R transactions to the general table GURFEED. When FURFEED is run the A/R transactions are moved from GURFEED to the FGBTRNI Finance table where after error checking they will be posted to the Finance ledgers.

2.9.1 Questions To Ask Yourself Loading Student Balances to Banner Accounts Receivable:

1. How do you want to load the data; summary balances or detail? Usually, most clients summarize the balance on the student's account and load just the balance as a Conversion Charge. This kind of a conversion takes only a couple of days. However if you decide to load detail information and want the system to act as if billing has already occurred, and enforce some kind of aging, the conversion becomes much more complicated and takes a much longer time. But it can be done.
2. When loading the balances, are these balances already represented in Banner Finance? If the AR balances are in Banner Finance, then you will want to mark the TBRACCD records as already fed to Finance. If the balances are not represented in Banner Finance, then you will want to run the interface processes to feed the transactions into Banner Finance.

The key table populated in this conversion is TBRACCD.

2.10 Grants

2.10.1 Grants Conversion

If a client has hundreds of grants, it may be easier to load the grant data into the FRBGRNT table from an excel spreadsheet.

2.10.1.1 Questions To Ask Yourself Loading Grant Codes:

2. Will you want your fund code to be exactly the same as your grant code? Many clients do this. By using this method, one can then use SQL to update the grant code on the fund code table along with the termination date and expenditure end date from the grant code itself.
3. What will your standard date be for termination date or expenditure end date? Many times the expenditure end date can be calculated as 6 months after the termination date. Or the Expenditure end date is not entered at all.

At a minimum, the table to populate is FRBGRNT, the grant table. Many other tables can be populated to support the grant, but are not discussed at this time. In addition, a grants billing conversion may also be performed and entails much more complexity than just loading the grant codes themselves.

When populating FRBGRNT, one would also load the supporting table FRRGRPI to represent the grant personnel if a Principal Investigator was loaded into FRBGRNT, and table FRVGRST to represent grant status. One record needs to exist for the primary investigator identified on FRBGRNT. Also one record is loaded into FRVGRST to represent the current Grant's status in FRBGRNT.

2.10.2 Grant Inception To Date Summary Balances

Clients typically enter grant expenses and revenue into the prior fiscal year prior to the go-live fiscal year. At that point a decision can be made. They do nothing more and load their fund balances in total which includes a Year End Roll of their grant expenses and revenue from legacy OR, they can perform a Year End Roll in Banner Finance to roll grant expenses into fund balance. The data is entered in with a June 30th date, or the last day of the fiscal year prior to go-live fiscal year. The purpose is to record inception to-date expenses and revenue into the grant ledger and to have the summary balance represented prior to the go-live date. The data would be loaded similarly as done for a mid-year conversion for Expenses and revenue.

The two methods to loading Grant Inception to date summary balances and their advantages/disadvantages:
METHOD 1: Without Year End Roll –most clients choose this one
You can either load the summary balances into GURFEED or hand enter them into FGAJVCQ. However, please note that you must use a new rule class code by copying the JE15 rule class code (Call it ITG for Inception To Date Grants) and changing the O030 process code to an O033 process code. The O033 process code prevents the system from calculating indirect costs and cost share. Therefore you will also need to generate entries for indirect costs, cost share and revenue.
ADVANTAGE: When fund balance load is performed, you do not have to exclude the grant funds' balances.
DISADVANTAGE: There are tables used to track the amount of costs used to date, to compare against any maximum values you may have set in your Basis code. If you are using maximum cap

amounts, the FRRGCSA, FRRGCST, FRRGICA, and FRRGICT tables will also need to be populated by a script to maintain maximum cost checks. It will also cause the billing tables to NOT be populated.

METHOD 2: With Year End Roll
You can either load the summary balances into GURFEED or hand enter them into FGAJVCQ using again a copy of the rule class code JE15 (ITG). Then a Year End Roll for that year would be performed to bring in the Grant Fund Balances into your go-live Year.
ADVANTAGE: You do not have to load fund balances for your grant funds.
DISADVANTAGE: When performing the Fund Balance load, you must exclude the fund balances for your grant funds. Also, the same disadvantage applies from method 1.

NOTE: It is recommended that you still use the new rule code no matter what you do and not the JE15 rule code as the calculation of the costs will probably be different than what you have incurred over the prior years. Thus you would still want to load your indirect costs.

Prior to loading these journal entries in GURFEED, you must ensure that the grant code itself has been set up in FRBGRNT table via the FRAGRNT form, and then associated to a fund code on the FTMFUND form to populate FTVFUND table.

Additionally, if you intend using the Deferred Grant process and or Grant Billing Process, you need to ensure that you have set up the correct information in FRVFUND table.

If the inception to date balances will be used to work with grant billing, these additional steps should be considered for a NEW Implementation.
Convert inception to date revenues and expenses from legacy system to Banner FOAPAL ‘s for funds that require tracking of inception to date activity across fiscal years. Prepare Excel files to create separate Journal Vouchers for entry in the fiscal year prior to your going live with Banner Finance for the following: The below entries (b – d) are for Grants using the Grant Billing process. For other multi-year inception to-date activity

a. All Multi-Year activity: Budget Transactions – Rule Code = BD01 – Make sure Budget Period is defaulting correctly from Rule Code

b. Grants: Revenue = Expense Transactions where Expense has not been billed- Rule Code = ‘ITG’. Transactions entered by Fund Org Acct Prog, where expense and revenue are operating accounts (FRS object codes). Document Code Prefix should be set = ‘JGU’. The ‘JGU’ is to identify documents that need to be fed as billing details. U meaning unbilled.

c. Grants: Revenue = Expense Transactions where Expense has been billed Rule Code = ‘ITG’. Use a different Document Code than in (b) above. Document Code Prefix = ‘JGB’. B meaning Billed.

d. DR Unbilled A/R = CR Cash for total amount of Expense (b and c), other GL activity is entered normally. Rule Code = ‘ITG’. Document Code Prefix = ‘JGG’. G meaning General Ledger activity.

e. For Non Grant Funds where Inception to Date Activity must be tracked create an entry for Inception To Date Revenue and Expense and offset the difference to fund balance. In the year- end close process step (FGRGLRL), the fund balance entry will be cleared.

f. If maximum limits for grants are being used, then scripts need to be set up to populate the tables FRRGICA and FRRGICT for the indirect costs taken to date.

2.10.3 Grant Billing

Grant Billing is a very complex conversion specific to each client dependent upon what implementation stage of Banner Finance Grants they are in and how they have been performing their accounting over the years for the grant billing. (I.E. are they a new client just going live or have they been using Banner Finance for awhile and want to implement grants or using Banner Finance for awhile with Grants and want to just implement Grants Billing.) This entails a week of onsite discussion with the client to determine how best to convert the data with the importance being more upon how the accounting has been managed these past few years.

Additional time is also needed to write the scripts to insert billing detail to insert history for those grants already billed and payments already made against those grants. The billing detail will need to be entered into the FRRBDET table based upon information populated in the FGBTRND table. Additionally, information will need to be inserted into TRRACCD based upon the payments already made against these Grant Billing charges.

2.10.3.1 Questions To Ask Yourself For Grant Billing:

1. Is this a new Banner Implementation or a Post Banner Implementation?
2. Does the aging report need to be accurate as of Grant Billing implementation date or will you be satisfied if open receivables all start with common date knowing aging report will eventually be accurate?
3. What data do you have available from legacy system?
4. New implementation: Have you billed all grants through year-end? Post implementation have you billed all grants through date of conversion?Do you use revenue recognition?If this is a post implementation then

2.11 Chart Conversion Load

As a client, you have a choice to either hand enter your Chart of Accounts tables or have them loaded for you via excel spreadsheets. There is an advantage and a disadvantage to both.

If you have thousands or hundreds of chart elements to be loaded, you may want to have them loaded via an excel spreadsheet. Sometimes, it is difficult to determine if you have exceeded the number of hierarchy limits that each chart has. Then can easily be found when you run the hierarchy reports after the chart has been loaded.

The four chart tables typically loaded through an excel spread sheet are FTVFUND, FTVORGN, FTVACCT, and FTVPROG. Sometimes, you might want to load FTVFTYP, FTVATYP and perhaps FTVACTL. Usually the effective start date for each code on each table is hard coded as one year prior to go-live. The status code is hard coded to (A)ctive and the chart code is hard coded to the Institution's Chart.

The fields necessary to populate each table are as follows:
FTVFUND – it is assumed that the FTVFTYP fund type table has already been loaded
Fund Code
Fund Predecessor Code
Fund Type Code
Fund Title
Ftvfund_Data_Entry_Ind - Data Entry Indicator
Ftvfund_pool_ind - Pool Indicator (Default to N)
Ftvfund_close_proj_ind - Close Proj Indicator (Default to N)
Ftvfund_Src_Cap_Same_Ind - Source Cap Same Indicator (Default to N)
Ftvfund_Bank_Code - Bank Code (optional)
Ftvfund_Rest_Ind - Restriction Indicator (Default to U) if FBAL indicator set, other valid values are (T)emporary or (P)ermanent
Ftvfund_Fbal_Ind – if set values are 'FBAL' or 'FUND' or 'ACCT'

FTVORGN
Organization Code
Organization Predecessor Code
Organization Title
Data Entry Indicator

FTVACCT – it is assumed that the FTVATYP account type table has already been loaded
Account Code
Account Predecessor Code
Account Type
Account Title
Data Entry Indicator
Normal Balance Indicator

FTVPROG
Program Code
Program Predecessor Code

Program Title
Data Entry Indicator

FTVFTYP
Fund Type Code
Fund Type Predecessor Code
Fund Type Title
Internal Fund Type Code
Default Override Indicator (Choose either O for Organization or F for fund.)

FTVATYP
Account Type Code
Account Type Predecessor Code
Account Title
Internal Account Type Code
Normal Balance Indicator

FTVACTL
Account Type Code
Operating Ledger Field Sequence Number
Control Account Code
Offset Account Code
Prior Year Control Account Code
Prior Year Offset Account Code

2.11.1 Questions To Ask Yourself for a Chart Conversion:

There are no questions other than determining if you want to load more than the minimum required data. I.E. load financial managers for the fund or organization code. If you decide to load Financial Managers you have two options: you can first load FTVFMGR table with the Financial Managers PIDM's and then load FTVFUND and FTVORGN. Or you can load FTVFUND and FTVORGN first, then extract the PIDM's for the Financial Managers from both tables (unique PIDM's) and load them into FTVFMGR table.

3. Finance Interfaces

All currently defined interfaces for Banner Finance journal vouchers and student refunds are fed through either the GURFEED (Finance Transaction Input Table) or GURAPAY (Accounting Feed Accounts Payable Table). GURFEED represents journal entries, GURAPAY represent a direct pay invoice. Thus Alumni journal entries, Payroll journal entries/encumbrances, and student payments and charges journal entries are fed into the GURFEED table. The hardest part about populating GURFEED will be the determination of the rule class code to use and cross-walking the legacy accounting chart elements into Banner Finance Chart elements. A/P Check reconciliation is done via the FABBKTP table.

3.1 A/P Bank Reconciliation

Bank reconciliation is performed by loading the bank data into the Bank tape table FABBKTP. After loading the data in the FABBKTP table, the reconciliation process FARBREC can be run to reconcile A/P checks. The data requirement for the FABBKTP table is very simple. Please remember that the Internal Document Number in the table must be filled with the same value as the Check/Deposit Number value.

3.1.1 Questions To Ask Yourself for Bank Tape Load

1. Does the bank data represent data for one account or multiple accounts?
2. Does the bank data only represent checks or both checks and deposits. This determines the setting of the check_deposit_ind. This will be 'C' for checks, 'D' for deposits.
3. Is the check number that comes from the bank the same as the one you have in Banner as the document number in the check table FABCHKS? Some clients utilize 6 characters check numbers while filling the first two characters with some identifiers such as "C0" for Banner 8 characters document number requirement. However, when the data comes from the bank, the will have to concatenate "C0" to the check numbers given by the bank to reconcile to the correct check number in Banner.
4. Will the bank send you an identifier to represent checks vs. deposits or cashed checks vs. voided checks?
5. Will the bank send you the check date?
6. Will the bank send you amounts with decimal point or not?
7. How will you receive the information from the bank? It is common for banks to allow users to download monthly checks reconciliation data via on-line or receive them via e-mail from the bank. You must then extract the data from the provided data files into an ASCII text file if the downloaded/received file is not in flat format type. Furthermore, the Bank will provide users with a document that determine the positions for the fields that will be loaded into the Bank tape table FABBKTP.

3.2 JV Interfaces via GURFEED

GURFEED can be used via external systems such as Travel System and Library system to feed JV transactions into Banner ledgers directly. Before approaching the Banner side of this process, you will need to determine the feed data format coming from the external systems into GURFEED.

3.2.1 Questions To Ask Yourself for Finance Journal Entry Interfaces:

1. Where will the crosswalk of legacy codes need to be maintained? There are several choices. Either in the legacy system, which is recommended if this area is to never become a part of Banner Finance, in a separate Oracle table, in the Entity Translation Table FTVEELT for Banner

Finance, or in the Reporting Attribute Tables. This FTVEELT table was created to specifically crosswalk legacy or outside entity codes to a Banner finance Chart code.
2. What rule class codes are to be used for each type of transaction?
3. You will need to set the debit credit indicator based upon the rule class code.
4. Will this interface be temporary until the legacy system is brought into Banner, or will it be permanent?
5. Will you use the system-ids already defined in FTVSDAT or need to create your own?
6. Will you be feeding non-payroll encumbrance generation via GURFEED? This will only apply to Banner 6.x with limitation that might not allow you to use this option at this time.
7. Will you be feeding encumbrance liquidation via GURFEED?

When preparing for "go-live" with Banner Finance, the GURFEED table is instrumental during the conversion process or in designing any external feeds into Finance. What follows is an explanation of how to populate the GURFEED Table and the steps necessary to bring this population into Banner Finance.

HINTS:
When populating the GURFEED table, think of it as writing a journal voucher in Banner Finance. A transaction in the journal voucher is represented by a journal voucher header record FGBJVCH (equivalent to a record type 1 in GURFEED) and journal voucher detail records FGBJVCD (equivalent to a record type 2 in the GURFEED table). Test your perceived population by directly entering the transaction into the Finance TEST environment on the FGAJVCD form. This way you can assure yourself that the population of the table will work without trying to guess at it.

POPULATING THE GURFEED TABLE:
Following are the required fields for all transactions into Finance:. There may be other fields you will need to use, such as GURFEED_LOCN_CODE or GURFEED_BANK_CODE for specific types of transactions. Most of these fields are self-explanatory; since the interface process creates a journal voucher. Field names in GURFEED correspond to field names in FGBJVCD.

You must provide a Header record for each set of transactions being batched and sent into Finance. The GURFEED Table represents two types of records, a header record and detail records all within the same structure. To denote the difference to the system, the record type is set to a 1 to represent Header records and a 2 to represent Detail records.

The header record is used as a control for all attached detail records. The key structure must be the same between the header and the associated detail records. This key structure is represented by:
GURFEED_SYSTEM_ID
GURFEED_SYSTEM_TIME_STAMP
GURFEED_DOC_CODE
GURFEED_USER_ID
GURFEED_TRANS_DATE

All records with these same matching keys are processed as a single journal voucher in the system. You need to create **ONE** header record for the journal voucher. And as many detail records necessary to support the transaction. The transaction amount on the header records represents the **HASH** total of the transaction amounts on the associated detail records regardless of the debit credit sign. This is a system control check and can be used to manually force the journal vouchers to

suspend by setting the amount on the header record to zero. This allows for intervention and manual manipulation of the journal voucher on the FGAJVCD form once it has been fed into Finance**. IE. $50.00 D plus -$50.00 D gives you a document total of $0.00. A $50.00 D plus a $50.00 C gives you a document total of $100.00.**

THINGS TO DO BEFORE POPULATING AND FEEDING GURFEED INTO FINANCE

1. Decide which rule class codes you are utilizing in the transaction and ensure that they exist in the Rule Class Code table with the proper process codes. (FTVRUCL, FTVEDIT, FTVRULP)
2. Make sure that the Chart of Accounts being used has been set up in their respective tables (FTVCOAS, FTVFUND, FTVORGN, FTVACCT, FTVPROG. FTVFSYR) with the correct effective dates.
3. Decide upon a system ID for the feed and enter the system ID into the form FTMSDAT. This is where you control how the transaction is being fed, either in (S)ummary mode, (D)etail mode, and if there are any errors, to either automatically (S)uspend the transaction as a journal voucher or to (R)eject the transaction and not populate the Finance tables. To do this:

a. Enter into the FTMSDAT form the following values:

Screen Literal	Column	Value
Entity/Usage Code	FTVSDAT_Sdat_Code_Entity	'FGBTRNI'
Attribute Code	FTVSDAT_Sdat_Code_Attr	'SYSTEM_ID'
Optional Code #1	FTVSDAT_Sdat_Code_Opt_1	'*XFEED*' user defined
Title	FTVSDAT_Title	*'External Feed Process'* user defined
Data	FTVSDAT_Data	'*DS*' user defined

Valid Data field combinations are:
SR = Summary/Reject
SS = Summary/Suspense
DR = Detail/Reject
DS = Detail/Suspense

Where 1st column: D= Detail, output records are written as input
S= Summary, output records are summarized from input

Where 2nd column: R= Reject, does not write a Journal voucher record if an error is found
S= Suspense, writes a journal voucher record in suspense if an error is found

If the summary option is chosen, transactions are summarized by like coas/fund/orgn/acct/prog/actv/locn/rule/document reference number/transaction description and debit credit indicator within a document into a single transaction to reduce the number of postings into the ledgers.

Populating GURFEED Document Header Records (Required Fields)

GURFEED_SYSTEM_ID Defined in FTMSDAT as defined above. This must also be the same value on the associated detail records.

GURFEED_SYSTEM_TIME_STAMP
Format TO_CHAR(sysdate,'YYYYMMDDHH24MISS'); for example, March 5, 1998 would be 19980305000000. This must also be the same value on the associated detail records.

GURFEED_DOC_CODE The document identifier for the Journal Voucher. This is the code used to view detail data after the document has posted to the ledgers. This must also be the same value on the associated detail records. This is typically controlled by the one-up number from FOBSEQN where fobseqn_seqno_type ='F'. This document number cannot already exist in the ledgers. That is why we prefer to use the FOBSEQN table in assigning the document numbers for external feed control purposes. In Finance release 5.4 you may want to consider using the new FOBFSEQ table that allows for a two-digit prefix and one up number. This allows one to separate each type of feed by a two-digit prefix.

GURFEED_REC_TYPE Set = "1" for header records

GURFEED_SEQ_NUM Set = "0" for header records. This field cannot be null!

GURFEED_ACTIVITY_DATE
Format DD-MMM-YYYY (05-MAR-1998). Typically the System Date or the date the GURFEED record is populated.

GURFEED_USER_ID Typically 8 bytes in length (FIMSMGR) with a maximum of up to 30 characters. The Oracle User ID of the person creating this transaction. Must also be the same value on the associated detail records.

GURFEED_TRANS_DATE Format DD-MMM-YYYY (05-MAR-1998). This is the date used to determine which Fiscal Year and Fiscal Period the transaction posts into the ledgers. Must also be the same value on the associated detail records.

GURFEED_TRANS_AMT The total of all the values of each transaction in document detail records. I.E 1 record has debit of 100 dollars, record 2 has a credit of 100 dollars, the header transaction amount is represented as 200 dollars.

Populating GURFEED Document Detail Records (Required Fields):

GURFEED_SYSTEM_ID Matches Header

GURFEED_SYSTEM_TIME_STAMP
Matches Header

GURFEED_DOC_CODE Matches Header

GURFEED_REC_TYPE Set = "2" for detail records

GURFEED_SEQ_NUM A “One-Up” line number for each detail record unique within the document. Begin with the number 1.

GURFEED_ACTIVITY_DATE
Format DD-MMM-YYYY (05-MAR-1998). Typically the System Date or the date the GURFEED record is populated

GURFEED_USER_ID Matches Header

GURFEED_RUCL_CODE The rule class code defined in FTVRUCL that tells posting how to post this transaction into the ledgers.

GURFEED_TRANS_DATE Matches Header

GURFEED_TRANS_AMT The dollar amount of the transaction.

GURFEED_TRANS_DESC Brief description of the journal voucher record being created.

GURFEED_DR_CR_IND The debit credit indicator. Set to 'D' or 'C' for general ledger type transactions. '+' or '-' for operation ledger type transactions. For most Payroll transactions, you will use a 'D' or 'C' as the initiator routines as Payroll Rule Class codes are I061.

GURFEED_COAS_CODE Chart of Accounts code defined in FTVCOAS.

GURFEED_FUND_CODE Fund code defined in FTVFUND. Field should be Left Justified with no blanks in it.

GURFEED_ORGN_CODE Organization code populated only if an operating account is used. Defined in FTVORGN. Field should be Left Justified with no blanks in it.

GURFEED_ACCT_CODE Account code defined in FTVACCT. Operating ledger accounts are defined where the "INTERNAL" account type of the account is equal to either 50, 60, 70, 80. Field should be Left Justified with no blanks in it.

GURFEED_PROG_CODE Program code populated only if an operating account is used. Defined in FTVPROG. Field should be Left Justified with no blanks in it.

Populating GURFEED Document Detail Records (Optional Fields):

GURFEED_BANK_CODE The bank code is required for certain transactions that require a bank to post the transaction. Predefined in GXVBANK.

GURFEED_DOC_REF_NUM
Document reference number is user defined and provides another reference mechanism to the posted transaction. Field should be Left Justified with no blanks in it.

GURFEED_BUDGET_PERIOD
Budget Period must be populated when budget type transactions are performed. I.E. using rule classes similar to BD01, BD02, BD03, and BD04. This is the numeric period (01 for example to represent period one in the fiscal year) that the budget is to be posted into and may be different than the fiscal period set by the transaction date.

FOR ENCUMBRANCE LIQUIDATION OR GENERATION TRANSACTIONS

Some transactions are written to liquidate existing encumbrances in the system. In order to do this, you must populate the following fields.

GURFEED_ENCD_NUM — The encumbrance number to be liquidated or generated. Must already exist in the encumbrance table as fgbench_num if this is an encumbrance liquidation. Field should be Left Justified with no blanks in it.

GURFEED_ENCD_ITEM — The encumbrance item to be liquidated or generated. It is mostly hard coded with '0'. Must already exist in the encumbrance table as fgbencd_item if this is an encumbrance liquidation.

GURFEED_ENCD_SEQ — The encumbrance sequence to be liquidated or generated. Must already exist in the encumbrance table as fgbencd_seq_num if this is an encumbrance liquidation.

GURFEED_ENCD_ACTION_IND — The action to perform upon the encumbrance. 'T' = Total, 'P' = Partial, 'A' = Adjustment for liquidation or leave NULL for generation.

GURFEED_ENCB_TYPE — The type of encumbrance (R)equisition, (P)urchase order, General (E)ncumbrance or (L)abor Encumbrance.

NOTES (1):

Edit Codes on Rule Classes can be used to force defaulting from Account Indexes or defaulting from Fund or Organization. The edits for index is 3800, 3801, 3802, 3803. Fund defaults are 4000; Organization defaults are 4201. The FGRTRNI process does the defaulting. The edits should be bypassed by FGRTRNI if not applicable.

NOTES (2):

Banner 6.0 Release includes an RPE to allow the posting process FGRACTG to post feed documents that causes encumbrance generation with rule process code E117. This allows feed documents that contain encumbrance generation data for non-labor encumbrance to be posted by the posting processes. You must populate the gurfeed_encd_num, gurfeed_encd_item, and gurfeed_encd_seq with the encumbrance information to be generated.. Set the gurfeed_encb_type = 'E'. However, there are no changes in the FGAJVCD or FGAJVCQ forms to allow the same RPE. Thus, a document that contains encumbrance generation data CANNOT be suspended in the JV tables. Or in other words, if an encumbrance line to be generated contained an error, it can never be corrected online in the Journal Voucher forms. However, if you remove the line in error, while on the journal voucher form, and do NOT touch the other lines in the document, you can complete the document and it will post, generating encumbrances for the lines not in error.

3.3 Student and General Refunds via GURAPAY

GURAPAY can be used via external systems such as Travel System and legacy Student A/R to feed refund check transactions into Banner via Banner Accounts Payable module. Before approaching the Banner side of this process, you will need to determine the feed data format coming from the external systems into GURAPAY. You will also need to analyze the requirement that all refunds require the presence of the PIDM (and thus the Banner general person record) in Banner. Please

remember that the FURAPAY process will only generate a direct pay invoice for a given PIDM and CAN 'T generate a One-Time-Vendor Direct Pay Invoices or Encumbrance Invoices.

NOTES:

The FURAPAY process requires the presence of the PIDM in Banner to process the refund records found in GURAPAY. There is a modified (by a client) FURAPAY process that will allow you to feed a One-Time-Vendor Direct Pay Invoices. This will eliminate the need for your institution to convert all person data from all legacy systems into Banner by Finance go-live date. However, please note that One-Time-Vendor Invoices are not a recommended practice in Banner Finance.

3.3.1 Questions To Ask Yourself for Finance General and Student Refund Interface:

1. Where will the crosswalk of legacy codes need to be maintained? There are several choices. Either in the legacy system, which is recommended if this area is to never become a part of Banner Finance, in a separate Oracle table, in the Entity Translation Table FTVEELT for Banner Finance, or in the Reporting Attribute Tables. This FTVEELT table was created to specifically crosswalk legacy or outside entity codes to a Banner finance Chart code.
2. Will this interface be temporary until the legacy system is brought into Banner, or will it be permanent (i.e. for external travel systems)?
3. Since the student must already exist in SPRIDEN in order for the invoice to be created, a methodology must be constructed whereby those students are entered into Banner through the FOAIDEN or FTMVEND form. You will need to make a decision, do we convert and load all students into Finance up front, or do we only load those students in, one at a time that will be obtaining refunds. Furthermore, you will need to determine how to cross maintain the data in both systems (synchronization of person data in both systems). Options are via manual maintenance (multiple data entry in different systems) or automated (via synchronization automation scripts).

To create an accounts payable invoice from outside the Banner Finance system, use the following field definition found in the GURAPAY table:

GURAPAY_SYSTEM_ID (maximum 8 characters) is MANDATORY

Defined in FTMSDAT as defined in GURFEED section above. A system ID that must be defined on FTMSDAT form with the following values:

Entity/Usage Code: FGBTRNI
Attribute Code: SYSTEM_ID
Optional Code #1: The SYSTEM_ID you intend to use
Data: DS

GURAPAY_SYSTEM_TIME_STAMP (maximum 14 characters) is MANDATORY

Use the formatting to character function in Oracle (TO_CHAR(sysdate,'YYYYMMDDHH24MISS')). The format mask must be "YYYYMMDDHHMMSS"; for example, July 23, 1992 would be 19920723000000

GURAPAY_DOC_CODE (maximum 8 characters) is MANDATORY

This will represent the invoice number in the invoice tables. This code must be unique and may not pre-exist on the FGBTRNH table. This is typically controlled by the one-up number from FOBSEQN where fobseqn_seqno_type = 'S'. In Finance release 5.4 you may want to consider

using the new FOBFSEQ table that allows for a two-character prefix and one up number. This allows one to separate each type of feed by a two-character prefix

GURAPAY_USER_ID (maximum 30 characters) is MANDATORY

This should be a user ID defined in Banner.

GURAPAY_PIDM (maximum 8 numbers) is MANDATORY

This must be an existing PIDM on the SPRIDEN table. All name and ID information are drawn from the PIDM.

GURAPAY_ID (maximum 9 characters) is MANDATORY

However, what is entered here is irrelevant, since the PIDM will retrieve the correct ID from SPRIDEN

GURAPAY_TRAN_NUMBER (maximum 4 characters) is MANDATORY

In Banner Accounts Receivable, this is the transaction number on the customer record. It does not affect invoice processing. Hard code this value.

GURAPAY_DETAIL_CODE (maximum 4 characters) is MANDATORY

What is entered here is irrelevant and not used by the invoices.

GURAPAY_DESC (maximum 30 characters) is MANDATORY

What is entered here will be shown in the item level description of the invoice. FURAPAY process, which reads this table, inserts "Student Refund" or “Refund” as the commodity description in the error table if there were errors generated by processing the records. Thus, the error report will show “Student Refund” or “Refund” in the item description.

GURAPAY_TERM_CODE (maximum 6 characters) is MANDATORY

This is also irrelevant to the invoice creation process.

GURAPAY_ACCOUNT (maximum 60 characters) is OPTIONAL

This field is used for Banner Accounts Receivable feeds to external finance systems and should be left blank if feeding Banner Finance.

GURAPAY_DR_CR_IND (maximum 1 character) is MANDATORY

This should be + for a refund or – for a credit memo.

GURAPAY_SRCE_CODE (maximum 1 character) is MANDATORY

This is irrelevant to the invoice creation process.

GURAPAY_LAST_NAME (maximum 25 characters) is MANDATORY

What is entered here is irrelevant, since the PIDM will retrieve the correct last name from SPRIDEN.

GURAPAY_FIRST_NAME (maximum 15 characters) is OPTIONAL

What is entered here is irrelevant, since the PIDM will retrieve the correct first name from SPRIDEN

GURAPAY_MI (maximum 1 character) is OPTIONAL

What is entered here is irrelevant, since the PIDM will retrieve the correct middle initial from SPRIDEN

GURAPAY_STREET_LINE1 (maximum 30 characters) is MANDATORY

What is entered here is irrelevant, since the combination of PIDM, Address Type Code, and Address Sequence will retrieve the proper address.

GURAPAY_STREET_LINE2 (maximum 30 characters) is OPTIONAL

What is entered here is irrelevant, since the combination of PIDM, Address Type Code, and Address Sequence will retrieve the proper address.

GURAPAY_STREET_LINE3 (maximum 30 characters) is OPTIONAL

What is entered here is irrelevant, since the combination of PIDM, Address Type Code, and Address Sequence will retrieve the proper address.

GURAPAY_CITY (maximum 20 characters) is MANDATORY
What is entered here is irrelevant, since the combination of PIDM, Address Type Code, and Address Sequence will retrieve the proper address.

GURAPAY_STAT_CODE (maximum 2 characters) is OPTIONAL
What is entered here is irrelevant, since the combination of PIDM, Address Type Code, and Address Sequence will retrieve the proper address.

GURAPAY_ZIP (maximum 10 characters) is OPTIONAL
What is entered here is irrelevant, since the combination of PIDM, Address Type Code, and Address Sequence will retrieve the proper address.

GURAPAY_NATN_CODE (maximum 5 characters) is OPTIONAL
What is entered here is irrelevant, since the combination of PIDM, Address Type Code, and Address Sequence will retrieve the proper address.

GURAPAY_ATYP_CODE (maximum 2 characters) is MANDATORY
This is the Address Type Code required to retrieve the address data from SPRADDR

GURAPAY_ATYP_SEQNO (maximum 2 numbers) is MANDATORY
This is the Address Type Sequence Number.
The address type and sequence number must be defined for the entered PIDM on the SPRADDR table

GURAPAY_TRANS_DATE (oracle date format) is MANDATORY
This is the date which the invoice you are creating will affect your accounting system.

GURAPAY_TRANS_AMT (maximum 12 numbers with 2 decimals) is MANDATORY
This is the amount of the invoice you will be creating.

GURAPAY_COAS_CODE (1 character) is MANDATORY
This should be your chart of accounts code

GURAPAY_ACCI_CODE (maximum 6 characters) is OPTIONAL
Enter an account index if desired.

GURAPAY_FUND_CODE (maximum 6 characters) is MANDATORY
Fund code must be entered unless it is defaulting from the account index.

GURAPAY_ORGN_CODE (maximum 6 characters) is OPTIONAL
Organization code is only entered if you are affecting an operating ledger account.

GURAPAY_ACCT_CODE (maximum 6 characters) is MANDATORY
Account code must be entered unless it is defaulting from the account index. This will be the account debited upon posting of the invoice.

GURAPAY_PROG_CODE (maximum 6 characters) is OPTIONAL
Program code is only entered if you are affecting an operating ledger account.

GURAPAY_ACTV_CODE (maximum 6 characters) is OPTIONAL
Activity code is optional and only used in the operating ledger

GURAPAY_LOCN_CODE (maximum 6 characters) is OPTIONAL
Location code is optional and only used in the operating ledger

GURAPAY_ACTIVITY_DATE (oracle date format) is MANDATORY
This should be set to today's date.

4. Conversion To Banner

4.1 Description

The Conversion Process

The section outlines the generic conversion of data into Banner. This methodology can be used for any Banner Product.

The generic conversion to BANNER process consists of 3 steps:

Step 1. Map legacy data to Banner table(s) and fields.

Each Conversion table in Banner is defined and its fields mapped to legacy data fields. The mapping documents should contain all columns that could possibly be populated during the conversion. Record types define the data that may be stored in individual Banner tables. Most if not all conversion data fields will be created as a character record types initially. This allows the client to populate different portions of the table at different times should their data not be stored in common files. In addition, only the portions of the table identified by the record types will be populated. This makes the conversion program less complex and easier to test. And since VARCHAR2 is defined for character data, no disk space is consumed when data fields within the table are null.

Mapping will include crosswalk information where legacy data needs to be converted to BANNER data that is defined with different coding structure and/or code definitions.

This step may define one or more tables depending on the conversion process. For example, in the Vendor conversion process, there are two conversions. The first populates the general person tables SPRIDEN, SPRADDR and SPRTELE, the second then populates the vendor table FTVVEND.

Defining multiple tables and process provided not only a logical sequence to the conversions, but also simplifies the process and reduces the time in testing the individual processes.

Step 2. Extract Data from Legacy system

In this step, the client will implement and run extract programs, scripts, tools … etc. to extract the data from the legacy system into flat data files. The data files output will be dependent on the method the client will be invoking their conversion. A simple conversion methodology is to extract each set for data for a Banner table in a separate file (e.g. extracting data for the SPRIDEN table in a separate file while extracting data for the SPRADDR table another separate file). The more complex way of conversion is to extract all the data sets in one SUPER TABLE format data files (e.g. extracting all the data for SPRIDEN, SPRADDR, SPRTELE and SPBPERS in one file). This method is simpler to implement (as to extract the data) but more complex to convert into Banner.

Step 3. Populate the conversion temporary table(s) (e.g. SCT Converter Tool Tables)

In this step, scripts (e.g. via SQL Loader control files) to populate the temporary tables are developed based on the mapping document prepared in step 1 (to determine the fields to be load in each table). Thereafter, the data is loaded directly into the temporary conversion tables.

All editing of the legacy to BANNER data **should** be done to the temporary tables either during the load or after as data conversions require. When the data is clean and all data visual testing has been completed, this table will then be used to populate the BANNER tables.

Step 4. Execute conversion scripts and populate the appropriate BANNER tables

This step can be done via in-house written conversion scripts or SCT provided tools (e.g. SCT Converter Tool generated scripts or other SCT scripts). Some of these tools read each row of the temporary table and then test, convert and insert the data into the appropriate Banner tables.

SCT also provides conversion tools and services that help clients convert data into Banner different modules. Examples are SCT Converter Tool that help clients generate the required code to convert into any Banner/Oracle table. Other tools that can be used to convert into Banner tables are Oracle Warehouse Builder, which works, in similar nature to SCT Converter tool. Both tools generate codes (PL/SQL scripts) move data into an Oracle tables with built-in conversion rules.

Conversion Tools and SCT Technical Services

SCT also provides conversion tools and services that help clients convert data into Banner all modules. Examples are SCT Converter Tool that helps clients generate the required code to convert into any Banner/Oracle table. Other tools that can be used to convert into Banner tables are Oracle Warehouse Builder that works in similar nature to SCT Converter tool. Both tools generate codes (PL/SQL scripts) move data into an Oracle tables with built-in conversion rules.

The mapping of legacy data to Banner described in step 1 above can be contracted as a provided service by SCT Technical Consultant who understands the requirements of the Banner tables with the help of the client team members who understand the data housed within the legacy system. The SCT Functional consultant can also be involved with the technical consultant on the more complicated type of conversions.

While the conversion scripts that load the temporary tables in step 2 and step 3 are the responsibility of the client (either manually or via one of the tools described above), SCT Technical Services can be contracted to provide conversion services upon request. Typically, the client will have the responsibility of extracting and delivering the flat data files to SCT Consultants who in their terms convert the data into the client database Banner tables (based on the mapping document agreed upon).