General Release Guide

SCT Banner

Release 6.2 / April 2004

Confidential Business Information

Prepared For: General Release Guide, Release 6.2

Prepared By: SunGard SCT
4 Country View Road
Malvern, Pennsylvania 19355
United States of America

General

This page intentionally left blank

Overview

This guide documents Release 6.2 of the SCT Banner General System. Release 6.2 includes all of the enhancements and problem resolutions delivered in Releases 6.1.0.1, 6.1.1, 6.1.2, and 6.1.2.1, as well as new enhancements and new problem resolutions.

This is a quick review of the releases described in this document:

Release	*Contents*	*Release Date*
Release 6.1.1	APIs to support e-Procurement	January 2004
Release 6.1.2	SCT Banner Error Messages Form (GUAERRM) Operational Data Store (ODS) Views	January 2004

This document describes the following new enhancements:

- User Interface (UI) changes to make disabled text easier to read.
- Support for Web credit card payment changes.
- An interface to FSA Atlas for SEVIS processing (defect #91900).
- Removal of *Workflow* from the Electronic Documents menus and form descriptions on the title bars.
- Improved menu performance.
- Miscellaneous enhancements.

Appendix A contains instructions on enabling Messaging in Banner. You will need to enable Messaging if you want to use e-Procurement at your institution.

This release also includes problem resolutions.

Prerequisites

Before you install General Release 6.2, you must install General Release 6.1.

In addition, you must perform the following steps:

1. Access the `formsweb.cfg` Forms Service Configuration File located in the `$ORACLE_HOME/forms60/server` directory.

2. Find the `ARCHIVE` parameter.

3. Add *bannerui.jar* to the parameter so it looks like this:

   ```
   ARCHIVE=banicons.jar,bannerui.jar
   ```

4. Save your changes.

Section I User Interface (UI) Enhancements - Functional

Overview

Note: This enhancement pertains only to institutions that use the `OracleLookAndFeel` parameter and `colorScheme` BLAF.

Previously, the labels of disabled fields had been difficult to read due to a feature in Oracle.

For this release, Banner General will allow you to change the color of disabled text to any color you'd like. As delivered, disabled text is black (disabled text will look the same as regular text).

This page intentionally left blank

Section II Support for Credit Card Enhancements - Functional

Overview

For this release, General has been modified to support enhanced credit card processing via the Internet:

- The Credit Card Transaction Audit Table (GORCCAU), has been delivered to help the technical staff at your institution resolve problems. They can use SQL to access the table to see:
  - Sent transactions
  - Returned transactions
  - Transactions that failed
- The Credit Card Merchant ID Form (GOAMERC) was given a new descriptive name, the Credit Card Transaction ID Form.
- The label **Merchant ID** on GOAMERC has been changed to **3rd Party Trans Code**.

Corresponding changes were made in Web General. For details, please refer to the *Web General 6.1 Release Guide.*

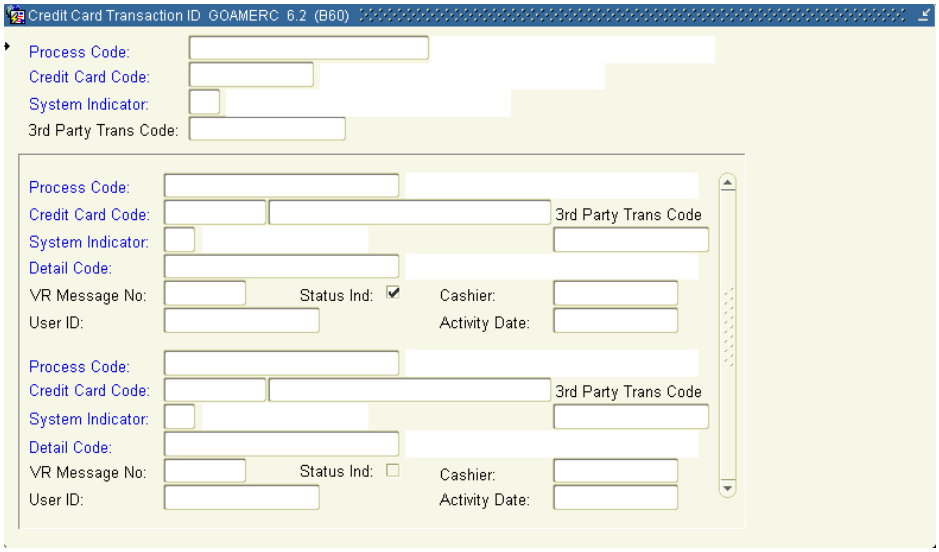
Changed Form

Credit Card Transaction ID Form (GOAMERC)

The text description of the form was changed from Credit Card Merchant ID to Credit Card Transaction ID. It still retains the mnemonic GOAMERC. A script has been provided to make the necessary changes to the General Object Base Table (GUAOBJS).

The **Merchant ID** field has been changed to be **3rd Party Trans Code**, and the status bar help has been changed to indicate that users should not enter the bank merchant number in that field.

These changes were made because some users had confused the Merchant ID with the bank merchant number.


Credit Card Transaction ID GOAMERC 6.2 (B60)
Process Code:
Credit Card Code:
System Indicator:
3rd Party Trans Code:
Process Code:
Credit Card Code:
3rd Party Trans Code
System Indicator:
Detail Code:
VR Message No:
Status Ind:
Cashier:
User ID:
Activity Date:
Process Code:
Credit Card Code:
3rd Party Trans Code
System Indicator:
Detail Code:
VR Message No:
Status Ind:
Cashier:
User ID:
Activity Date:

Section III Interface with SCT fsaATLAS - Functional

Overview

Note: You will not need this feature if your institute does not perform SEVIS processing, or if you use Banner SEVISTA to process SEVIS records.

The SEVIS Batch Export Process (GORSEVE) has been modified to allow you to produce a comma-separated value (CSV) file. This file can be imported by the fsaATLAS Campus DataLink instead of creating the GOASEVR transaction record that would normally be sent to SEVIS via SEVISTA.

You can use the CSV file as you wish, but it was designed for SCT fsaATLAS. SCT fsaATLAS is an application specifically designed for Visa, Case Management, and SEVIS tracking.

Note: If you use fsaATLAS, you will not use the SCT SEVIS Transfer Adaptor (SEVISTA) to transmit your records to the United States Bureau of Citizenship and Immigration Services (BCIS).

This differs from existing SEVIS processing in the following ways:

- No history records are created, so you cannot see the record on the SEVIS Transmittal History Form (GOASEVR).
- You will not use SEVISTA to validate and transmit your data.
- The CSV file will reside in your Job Submission directory and will have the same one-up name as the `GORSEVE.lis` file. You cannot view this file on the Saved Output Review Form (GJIREVO). You can download the file to fsaATLAS using the process in place at your institution.

If you do not specify individual Banner IDs or a population selection, the CSV file will contain an extract of all the records in SEVIS Data Table (GOBSEVS) and the associated views.

Note: If you do not have fsaATLAS, you can continue to use SEVISTA. There has been no change to SEVIS reporting when GORSEVE is run it Audit or Update mode for SEVISTA.

This enhancement resolves defect #91900.

Prerequisites

GORSEVE will not process records correctly unless the correct values have been entered into the Crosswalk Validation Form (GTVSDAX). This has not changed for this release.

For more information, please refer to the *SCT Banner General User Guide*.

Tasks

To create a CSV file of SEVIS data:

1. Specify individual Banner IDs or create a population selection for entry on GORSEVE. If no population selection or individual IDs have been entered on GORSEVE, all the records on the SEVIS Data Table (GOBSEVS) will be selected to be part of the extract/CSV file.

2. Enter the following parameters:

   (a) For parameter 02, enter the SEVIS User ID (Batch Approver).

   (b) For parameter 05, enter *Y* if you use Business Actions to create records. Otherwise enter *N*.

   (c) For parameter 06, enter *Y* if you use Auto-population to create records. Otherwise enter *N*.

   (d) For parameter 07, enter the Term Code if you use Auto-population. Otherwise, do not enter anything for parameter 07.

   (e) For parameter 08, enter the Banner IDs you want to process. Optional.

   (f) For parameters 09 - 12, enter population selection values. Optional.

   (g) For parameter 13, enter *Y* if you want to print records that have had no activity in the `GORSEVE.lis` file. Otherwise enter *N*.

   (h) For parameter 14, enter *E*.

   (i) For parameter 15, enter the Department Owner ID for fsaATLAS. This is optional for creating the CSV file, but it is required by fsaATLAS.

3. SCT fsaATLAS Campus DataLink will insert the Banner ID into the **Admissions ID** field in fsaATLAS. This will help you cross-reference the data in the two systems. **Admissions ID** is not SEVIS-specific information.

4. The CSV file resides in your Job Submission directory, just like the results of any other job. You can view the CSV file in any editor you choose. It will not be available on GJIREVO; it is only written to the Job Submission directory.

   Note: If you do not specify individual Banner IDs or create a population selection, all the records in the SEVIS Data Table (GOBSEVS) will be selected and written to the CSV file. You can run GORSEVE with the extract parameters as many times as you want; all it does is create another CSV file. It will not adversely affect Banner.

You can set up and use Business Rules (Business Actions and Auto-population) with the extract option. For more information, please refer to the *SCT Banner General User Guide*.

Changed Reports and Processes

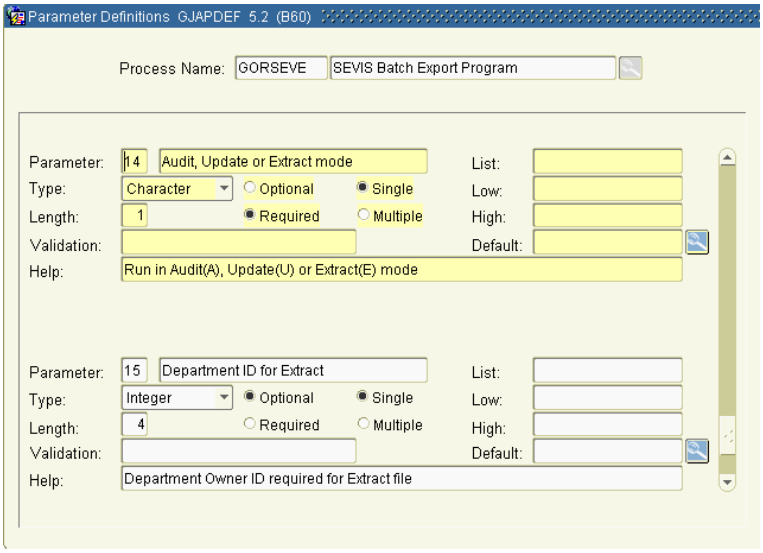
SEVIS Batch Export Process (GORSEVE)

New parameters have been added to GORSEVE to generate the Extract/CSV file.

Parameter 14, Mode in which to process GORSEVE, has an additional valid value. You can enter *E* to specify that you want GORSEVE to create an Extract file.

Note: If you use this new Extract option, GORSEVE will not create a SEVIS transmittal history record in GOASEVR.

Parameter 15, Department Owner ID for Extract, is also new. The only time you need to use it is when you are extracting data for fsaATLAS. In that case it is required. If you are creating a CSV file for other purposes, this parameter is optional. fsaAtlas determines the value you will use for the Department Owner ID.



Caution: The minimum data required by GORSEVE for each record Extracted is a creation reason or termination reason code. Records that do not have one or the other will not be processed by GORSEVE and included in the CSV file, but the records will appear on the .lis file.

Records that have both a creation reason code and a termination reason code will not be processed by GORSEVE, too.

GORSEVE requires the 02 - Batch Approver parameter to execute in Export mode, but it is not required by fsaATLAS. You can enter any value for Batch Approver; it cannot be left blank (be null).

Section IV Changed Descriptions of Electronic Documents Forms - Functional

Overview

Banner contains a module with the designation *Workflow*. This is a misnomer; the forms in this module are General forms, and can be used without the SunGard SCT Workflow product.

For this release *Workflow* has been removed from Banner menus and form descriptions to more accurately reflect the functionality of these forms. The mnemonics have not been changed.

Changed Form Descriptions

The descriptions (`gubobjs_desc`) of the following forms have been changed for this release:

Mnemonic	*Old Description*	*New Description*
GOAWFED	Workflow Electronic Documents Form	Electronic Documents Form
GOAWFIZ	Workflow Electronic Documents Wizard Form	Electronic Documents Wizard Form
GORWFDD	Workflow Electronic Documents Dictionary Form	Electronic Documents Dictionary Form
GORWFDB	Workflow Electronic Documents Builder Form	Electronic Documents Builder Form
GORWFDS	Workflow Electronic Documents Security Form	Electronic Documents Security Form
GTVWFED	Workflow Electronic Documents Validation Form	Electronic Documents Validation Form

Changed Menus

System Functions/Administration Menu (*GENSYS)

The Workflow Electronic Documents Menu (*GENWFED) is now the Electronic Documents Menu.

Workflow Electronic Documents Menu (*GENWFED)

This menu is now called the Electronic Documents Menu. The word Workflow has been removed from each of the forms listed on this menu.

Section V Menu Performance Improvement - Functional

Overview

In previous releases, the menus were built as a hierarchical tree as you logged on. Each time you logged on this tree of menu items was stored in global temporary tables. This could cause performance problems for the Welcome to Banner, Initializing System Form (GUAINIT) and the General Menu Form (GUAGMNU).

In addition, objects that were not displayed on the menu because the person could not access them could be viewed in other places. This would frustrate users because they could see something that they could not use.

For Release 6.2, the hierarchical tree is created differently and security is enforced differently.

Note: These changes apply to the Products menu, not the My Banner menu.

Changed Forms

General Menu Form (GUAGMNU)

This was changed to display the menu from the new GURHMNU table rather than the global temporary tables.

Welcome to Banner, Initializing System Form (GUAINIT)

This was modified to use the hierarchical tree table, decreasing the time needed to display the Banner main menu.

Object Maintenance Form (GUAOBJS)

When an object description is updated, the form will check to see if the object already exists on a menu.

- If it does exist on a menu, the object table (gubobjs) will be updated and the new GURHMNU table will be refreshed.
- If it doesn't exist on a menu, only the object table (gubobjs) will be updated.

When an object is deleted, this form will check to see if the object existed on a menu.

- If it did exist, the new GURHMNU table will be refreshed.
- If it didn't exist on a menu, only the object table (gubobjs) will be updated.

Note: If an object is inserted, the menu is not refreshed because the new object has never been on a menu yet.

In addition, the security policy has been changed. Previously, the main menu would not show any objects that a user could not see. However, the user could see those objects in other areas of Banner (QuickFlows, My Banner menus, etc.).

For Release 6.2, security will be provided via the object table, so the user will not be able to see those object anywhere in Banner, with the exception of the GUAOBJS form. Users who can access that form will be able to see all objects there.

My Banner Maintenance Form (GUAPMNU)

This form was changed prevent you from adding a form to a menu if it already existed at a higher level in the tree.

It was also changed to verify that the user has security access to the menu when adding the menu to the My Banner menu:

- If the user has security access to any object on the menu, GUAPMNU will allow the menu to be added.
- If the user does not have access to at least one object on the menu, the user cannot add the menu.

Menu Maintenance Form (GUTGMNU)

This form was changed to perform a refresh of the hierarchical tree whenever you make a change to something on the Products menu. It was also changed to prevent you from adding a form to a menu if it already existed at a higher level in the tree.

My Banner Maintenance and Copy Form (GUTPMNU)

This form has been changed so you cannot add a form to a menu if it already exists at a higher level in the hierarchical tree.

It was also changed to verify that the user has security access to the menu when adding the menu to the My Banner menu:

- If the user has security access to any object on the menu, GUTPMNU will allow the menu to be added.
- If the user does not have access to at least one object on the menu, the user cannot add the menu.

This page intentionally left blank

Section VI Views for Operational Data Store Processing - Functional

Overview

The delivered composite views collect SCT Banner source data to populate and maintain data stored in the SCT Operational Data Store (ODS) 2.0 product. These views follow conventional Object:Access standards and are designed to meet the reporting demands of the ODS.

Although these views are designed for use with ODS, you can use them without it. They will provide you with information just like any of the other views.

This page intentionally left blank

Section VII Support for e-Procurement - Functional

Overview

SCT has modified the Banner Finance System to integrate with third-party applications. Data from these applications can be stored directly in Banner tables without human intervention.

In addition, the common business logic and rules that are associated with certain basic Banner functions have been consolidated as part of this initiative. The APIs can be called by other programs, reusing the existing code. More APIs will be delivered in future releases.

APIs are database packages that contain the consolidated business logic and rules necessary to manage Banner tables. APIs can be called by Banner and by external systems. The same programming logic is executed no matter how the APIs are called.

For this release, three APIs have been delivered in the Banner General System:

- Identification
- Address
- Telephone

New Form

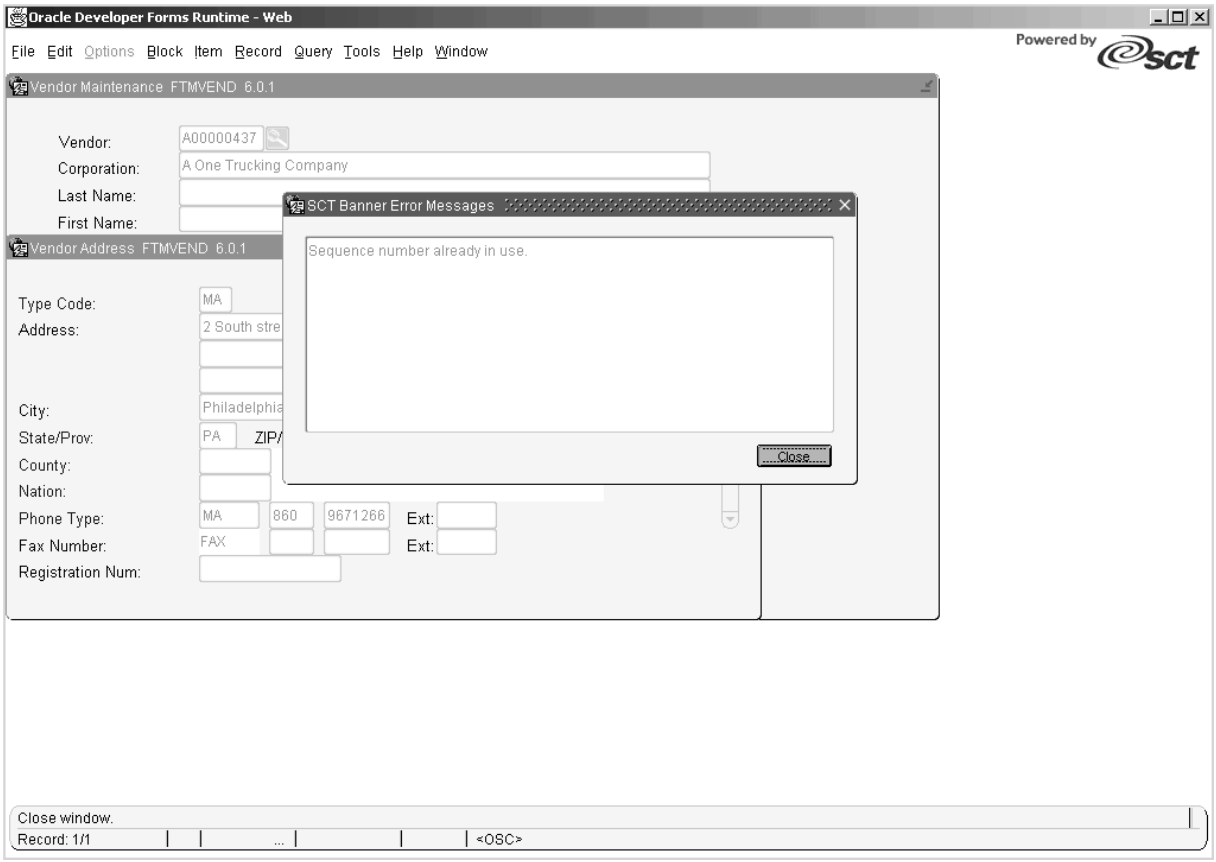
SCT Banner Error Messages Form (GUAERRM)

This form is a modal window, and it displays any error messages that are passed from an API to a form. Error messages are generated if, for example, a record fails to include required information or contains an invalid code upon committing a transaction.

This form is launched by new triggers that have been added to goqrpls.pll to support the use of APIs. The triggers are `ON-LOCK`, `ON-INSERT`, `ON-DELETE`, and `ON-UPDATE`, and an exception handler within the triggers call GUAERRM and displays the error messages.

This form appears as a window in a form, and is display-only. It can display messages up to 512 characters in length. Longer messages are truncated.

Note: This form is not accessible through any Banner menus.



Section VIII UI Enhancements - Technical

Overview

Note: This enhancement pertains only to institutions that use the `OracleLookAndFeel` parameter and `colorScheme` BLAF.

Previously, the labels of disabled fields had been difficult to read due to a feature in Oracle.

For this release, Banner General will allow you to change the color of disabled text to any color you'd like.

As delivered with this release, disabled text is black (disabled text will look the same as regular text). Therefore, the text has the R, G, B code:

- R=0
- G=0
- B=0

You can change these values.

Note: You can look on the Web to find the various R, G, B codes for different colors. SunGard SCT is not providing this information.

Tasks

To change the colors of your disabled text:

1. Create two temporary directories, e.g., `C:\temp\jar\default` and `C:\temp\jar\new`.
2. Place `bannerui.jar` into the `C:\temp\jar\default` directory.
3. Open a command prompt at the `C:\temp\jar\new` directory.

4. Unpack the bannerui.jar file into the `C:\temp\jar\new` directory:

   ```
   jar -xvf c:\temp\jar\default\bannerui.jar
   ```

   (assuming that you placed `bannerui.jar` into the `C:\temp\jar\default` directory)

5. Access the `disabledTextColor.properties` file.

6. Search for the `OracleLookAndFeel` parameter. It will appear in the following heading:

   ```
   ##########################################################################
   #
   # RGB settings to color Disabled Field Text ( OracleLookAndFeel)
   #
   ##########################################################################
   ```

7. The delivered values for R, G, and B are listed in that section. As delivered, they are:

   R=0
   G=0
   B=0

   The delivered values specify that the disabled text is black. Change the values for each as necessary to produce whatever color you'd prefer.

8. Save your changes.

9. Repackage the `bannerui.jar` file in the `C:\temp\jar\new` directory using the jar command:

   ```
   jar -cvf bannerui.jar *.*
   ```

10. Copy the new jar file to the `$ORACLE_HOME/forms60/java` directory on the application server machine for deployment.

New Files

bannerui.jar

This zipped Java file contains the settings for disabled text. It provides a client of Banner applications to select and set up a custom color for disabled text (for institutions that use the `OracleLookAndFeel` parameter and `colorScheme` BLAF).

For Banner to use that feature, the relevant configuration in the `formsweb.cfg` file must ensure that the delivered `bannerui.jar` file is included in the correct archive setting.

This file consists of two other files:

1. `BannerUIManager.class`
2. `disabledTextColor.properties`

disabledTextColor.properties

This file contains the R, G, B code for disabled text. You can modify these values to make the color anything you want.

This page intentionally left blank

Section IX Support for Credit Card Enhancements - Technical

Overview

The new Credit Card Transaction Audit Table (GORCCAU) table has been added for General Release 6.2. You can set it up to automatically be populated with Web credit card transaction information. The technical staff at your institution can use SQL to look at this information to help resolve problems.

You can set up the table to store information in one of three modes:

- Debug - stores information about all transactions that are sent to the merchants and the corresponding responses. Multiple records are stored for each transaction (sent and returned strings).
- Transaction - stores only the responses from the merchants for all transactions. One record is stored for each transaction.
- Failed transaction - stores only the responses for failed transactions. One record is stored for each failed transaction.

You must specify the mode you want on the Crosswalk Validation Form (GTVSDAX). If you do not set up the mode on GTVSDAX, this table will remain empty.

New Crosswalk Validation Form (GTVSDAX) Rule

You will need to set up certain values on the Crosswalk Validation Form (GTVSDAX) to define the mode you want to use. If you do not want to use this table, do not enter these values into GTVSDAX, or use *N* as described in the following table:

External Code	Internal Code	Internal Code Sequence Number	Internal Code Group	Description	Activity Date
D	AUDITMODE		CRDCARDPMT	Store debug information for all transactions in the new GORCCAU table. Can include multiple records per transaction.	Sysdate
T	AUDITMODE		CRDCARDPMT	Store all responses from vendors in the new GORCCAU table. Only one record is stored per transaction.	Sysdate
F	AUDITMODE		CRDCARDPMT	Store only the failed transactions in the new GORCCAU table. Only one record is stored per transaction.	Sysdate
N	AUDITMODE		CRDCARDPMT	Store no transactions in the new GORCCAU table.	Sysdate

New Scripts

Note: All the scripts described below are run as part of your upgrade.

gorccau1.sql

This database script creates the new GORCCAU table.

gorccau2.sql

This database script creates an index on Banner ID for the new GORCCAU table.

gorccau3.sql

This database script creates an index on PIDM for the new GORCCAU table.

gorccau4.sql

This database script adds comments to the columns for the new GORCCAU table.

New Table

Credit Card Transaction Audit Table (GORCCAU)

This table stores information about transactions sent from your institution to various merchants and the response information.

Different information is stored in the table depending on what mode you are in (e.g., all transactions, only failed transactions, etc.). You must set up the mode on the Crosswalk Validation Form (GTVSDAX); if you don't no information will be stored in this table at all.

Column	*Data Type*	*Not Null?*	*Description*
GORCCAU_ID	VARCHAR2(9)	Not Null	ID of the credit card user.
GORCCAU_PIDM	NUMBER(8)		PIDM of the credit card user.
GORCCAU_TERM_CODE	VARCHAR2(6)		Term code selected for the transaction.
GORCCAU_LOCATION	VARCHAR2(120)		Location of the package.procedure combination that is sending the data. **Note:** This is not populated at this time.
GORCCAU_DETAIL	VARCHAR2(20)		Detail code of the transaction.
GORCCAU_STREET_LINE1	VARCHAR2(30)		First line of the credit card user's billing address.
GORCCAU_STREET_LINE2	VARCHAR2(30)		Second line of the credit card user's billing address. **Note:** This is not populated at this time.

Column	*Data Type*	*Not Null?*	*Description*
GORCCAU_STAT_CODE	VARCHAR2(3)		Code that represents the state/province of the credit card user's billing address. **Note:** This is not populated at this time.
GORCCAU_ZIP	VARCHAR2(10)		ZIP/Postal code of the credit card user's billing address.
GORCCAU_NATN_CODE	VARCHAR2(5)		Code that represents the nation of the credit card user's billing address. **Note:** This is not populated at this time.
GORCCAU_CCRD_NUM	VARCHAR2(20)		The credit card number. The number is stored in the table in the format defined on GTVSDAX using the Internal Code *PMTSTORENO*.
GORCCAU_EXP_MO	VARCHAR2(2)		Month in which the credit card expires.
GORCCAU_EXP_YR	VARCHAR2(4)		Year in which the credit card expires.
GORCCAU_AMOUNT	VARCHAR2(16)		Amount of the transaction.
GORCCAU_MERCHANT_ID	VARCHAR2(20)		Identification number of the merchant.
GORCCAU_DEBUG_MSG	VARCHAR2(100)		Messages you can use to help resolve problems with transactions.
GORCCAU_STATUS	VARCHAR2(40)		Status of the transaction.
GORCCAU_SENT_STRING	VARCHAR2(3000)		String sent to the vendor to process the transaction.
GORCCAU_RETURN_STRING	VARCHAR2(3000)		String returned from the vendor with transaction results.
GORCCAU_ACTIVITY_DATE	DATE	Not Null	System date.

Changed Package

GOKFUNC.SQL/GOKFUN1.SQL

This package now contains the requisite insert and update statements so the various credit card packages can use this table.

Section X Interface with SCT fsaATLAS - Technical

Overview

Note: You will not need this feature if your institute does not perform SEVIS processing, or if you use Banner SEVISTA to process SEVIS records.

SCT fsaATLAS is an application that you can use for Visa, Case Management, and SEVIS tracking. If you use fsaATLAS, you will not use the SCT SEVIS Transfer Adaptor (SEVISTA) to transmit your records to the United States Bureau of Citizenship and Immigration Services (BCIS). If you do not have fsaATLAS and submit international data to SEVIS, you can continue to use SEVISTA - there is no change to current processing.

The SEVIS Batch Export Process (GORSEVE) has been modified to allow you to produce a comma-separated value (CSV) file that can be imported by the fsaATLAS Campus DataLink.

You can also use the new parameters to generate a CSV file for your institution's use. The GORSEVE Extract process will not create history records on the SEVIS History Table (GORSEVS).

For this release, GORSEVE.PC has been changed to select rows in the SEVIS Data Table (GOBSEVS) and the associated views:

- SEVIS Student Data View (GOVSVSD)
- SEVIS EV Data View (GOVSVEV)
- SEVIS Dependent Data View (GOVSVDP)

GORSEVE will select these rows and write them to a CSV file.

This process will occur after all Business Rules have been processed, both Business Actions and Auto-populations (if indicated in the program parameters).

After running GORSEVE, the data selected is based on what was entered manually on the SEVIS Information Form (GOASEVS) or what was updated via Dynamic Rules.

GORSEVE.PC will select rows based on any of the following criteria:

- Any entered IDs (required if processing a Business Action)
- Population selection (required if processing a Business Action)
- All records, if no IDs were entered and no population selections were specified.

Note: This enhancement resolves defect #91900.

Banner Columns Included in the Extract

The following Banner columns are included in the CSV file. For more information, please refer to the fsaATLAS documentation.

- GOVSVSD_LAST_NAME/GOVSVEV_LAST_NAME (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_FIRST_NAME/GOVSVEV_FIRST_NAME (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_MI/GOVSVEV_MI (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_NAME_SUFFIX/GOVSVEV_NAME_SUFFIX (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_BIRTH_DATE/GOVSVEV_BIRTH_DATE (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_SSN
- GOVSVSD_TIN
- GOVSVSD_GENDER/GOVSVEV_GENDER (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_BIRTH_NATN_CODE/GOVSVEV_BIRTH_NATN_CODE (depending on whether the person is a student or an exchange visitor)
- GOVSVEV_BIRTH_CITY
- GOVSVSD_LEGAL_NATN_CODE/GOVSVEV_LEGAL_NATN_CODE (depending on whether the person is a student or an exchange visitor)
- GOVSVEV_NATN_PERM_RES
- GOVSVSD_PASSPORT_NUMBER
- GOVSVSD_PASSPORT_EXPIRE_DATE
- GOVSVSD_NATN_CDE_PASSPORT
- GOVSVSD_VISA_NUMBER
- GOVSVSD_SVCP_CODE
- GOVSVSD_DRIVERS_LIC_NUMBER
- GOVSVSD_STAT_CODE_DRIVERS_LIC
- GOVSVSD_ADMISSION_NUMBER
- GOVSVSD_PENT_CODE
- GOVSVSD_PENT_COMMENT
- GOVSVSD_ENTRY_DATE
- GOVSVSD_VTYP_CDE/GOVSVEV_VTYP_CDE (depending on whether the person is a student or an exchange visitor)

- GOVSVSD_PROGRAM_BEGIN_DATE/GOVSVEV_PROGRAM_BEGIN_DATE (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_PROGRAM_END_DATE/GOVSVEV_PROGRAM_END_DATE (depending on whether the person is a student or an exchange visitor)
- GOVSVEV_NEW_PROGRAM_START_DATE
- GOVSVSD_MAJR_CDE
- GOVSVSD_MAJR_CDE_2
- GOVSVSD_MAJR_CDE_MINR
- GOVSVSD_SESSION_START_DATE
- GOVSVSD_SESSION_END_DATE
- GOVSVSD_CRIMINAL_CONVICT_IND
- GOVSVSD_SVEL_CODE
- GOVSVSD_STUDY_LENGTH
- GOVSVSD_ACADEMIC_TERM_MONTHS
- GOVSVSD_ENG_PROF_REQ_IND
- GOVSVSD_ENG_PROF_MET_IND
- GOVSVSD_ENG_PROF_REASON
- GOVSVSD_TUITION_EXPENSE
- GOVSVSD_LIVING_EXPENSES
- GOVSVSD_DEPENDENT_EXPENSES
- GOVSVSD_OTHER_EXPENSES
- GOVSVSD_OTHER_EXP_COMMENT
- GOVSVSD_PERSONAL_FUNDS
- GOVSVSD_SCHOOL_FUNDS
- GOVSVSD_SCHOOL_FUNDS_COMMENT
- GOVSVSD_OTHER_FUNDS
- GOVSVSD_OTHER_FUNDS_COMMENT
- GOVSVSD_EMPLOYMENT_FUNDS
- GOVSVSD_FUNDING_COMMENT/GOVSVEV_FUNDING_COMMENT (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_COMMUTE_IND
- GOVSVEV_SVSC_CODE
- GOVSVEV_SVSC_COMMENT
- GOVSVEV_SVPC_CODE
- GOVSVEV_SVCC_CODE
- GOVSVEV_PROGRAM_SPONSOR_FUNDS
- GOVSVEV_GOVT_ORG_FUNDS
- GOVSVEV_SVGO_CODE
- GOVSVEV_GOVT_ORG_FUNDS_2

- GOVSVEV_SVGO_CODE_2
- GOVSVEV_INTL_ORG_FUNDS
- GOVSVEV_SVIO_CODE
- GOVSVEV_INTL_ORG_FUNDS_2
- GOVSVEV_SVIO_CODE_2
- GOVSVEV_EV_GOVT_FUNDS
- GOVSVEV_BINATION_FUNDS
- GOVSVEV_OTHER_ORG_FUNDS
- GOVSVEV_PERSONAL_FUNDS
- GOVSVSD_USA_STREET_LINE1/GOVSVEV_USA_STREET_LINE1 (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_USA_STREET_LINE2/GOVSVEV_USA_STREET_LINE2 (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_USA_CITY/GOVSVEV_USA_CITY (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_USA_STAT_CODE/GOVSVEV_USA_STAT_CODE (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_USA_ZIP/GOVSVEV_USA_ZIP (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_ZIP_ROUTING_CODE/GOVSVEV_ZIP_ROUTING_CODE (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_NONUSA_STREET_LINE1/GOVSVEV_NONUSA_STREET_LINE1 (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_NONUSA_STREET_LINE2/GOVSVEV_NONUSA_STREET_LINE2 (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_NONUSA_CITY/GOVSVEV_NONUSA_CITY (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_NONUSA_STAT_CODE/GOVSVEV_NONUSA_STAT_CODE (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_NONUSA_ZIP/GOVSVEV_NONUSA_ZIP (depending on whether the person is a student or an exchange visitor)
- GOVSVSD_NONUSA_NATN_CODE/GOVSVEV_NONUSA_NATN_CODE (depending on whether the person is a student or an exchange visitor)
- GOVSVDP_SEQ_NO
- GOVSVDP_DEP_LAST_NAME
- GOVSVDP_DEP_FIRST_NAME
- GOVSVDP_DEP_MIDDLE_NAME
- GOVSVDP_DEP_NAME_SUFFIX
- GOVSVDP_BIRTH_DATE
- GOVSVDP_GENDER
- GOVSVDP_BIRTH_NATN_CODE

- GOVSVDP_DEP_BIRTH_CITY
- GOVSVDP_LEGAL_NATN_CODE
- GOVSVDP_NATN_CODE_PERM_RES
- GOVSVDP_VTYP_CDE
- GOVSVDP_RELT_CODE
- GOVSVDP_DEP_COMMENT

Note: The Department Owner ID is also included in the extract file, but it is a user-entered parameter for GORSEVE that is required by fsaATLAS, not a Banner table column.

No change has been made to the way GORSEVE processes codes. All Banner-only codes (e.g., Nation) will be converted to the corresponding SEVIS code before being included in the CSV file.

This page intentionally left blank

Section XI Changed Descriptions of Electronic Documents Forms - Technical

New Scripts

ugubobjswf62.sql

This script has been provided to change the descriptions (`gubobjs_desc`) of the following forms:

- goawfed.fmb
- gorwfiz.fmb
- gorwfdb.fmb
- gorwfdd.fmb
- gorwfds.fmb
- gtvwfed.fmb

It looks for the word *Workflow* in the program logic comments and removes each occurrence.

ugovwfdb62.sql

This script has been delivered to remove *Workflow* from the description in the GOVWFDB view.

Changed View

govwfdb.sql

This generates the Electronic Documents Builder View. It was previously called the Workflow Electronic Documents Builder View.

This page intentionally left blank

Section XII Menu Performance Improvement - Technical

Overview

In previous releases, the menus were built as a hierarchical tree as you logged on. Each time you logged on this tree of menu items was stored in global temporary tables. This could cause performance problems for the Welcome to Banner, Initializing System Form (GUAINIT) and the General Menu Form (GUAGMNU).

For Release 6.2, the hierarchical tree is created differently and security is enforced differently.

The Hierarchical Tree

The hierarchical tree is built when a menu is changed. It is stored in the new General Hierarchical Menu Table (GURHMNU). When you log on now, the tree is displayed from GURHMNU. The tree's records appear on GUAGMNU in the same order as they do in the table.

When you change a menu using the Menu Maintenance Form (GUTGMNU), the data in the table is refreshed. The table is also refreshed whenever an object description is changed on the Object Maintenance Form (GUAOBJS).

Note: When a user changes a description on the Object Maintenance Form (GUAOBJS), the change will not appear on the hierarchical menu table until the user performs a refresh. Auto Hint information was added on GUAOBJS: *Rebuilding the institution's hierarchical menu table, please wait.*

After making a change to the menu on GUTGMNU, the menu table is refreshed, but you can't see the change until you log out of Banner, then log back in. This is also true for changes you make on GUAOBJS.

Maintenance Via the General Object Base Table (GUBOBJS)

If you delete a submenu from the General Object Base Table (GUBOBJS) that has no objects underneath it, it is automatically deleted on GURMENU and GUAOBJS, and GURHMNU is refreshed.

If you delete a form, it is removed automatically from the menus, QuickFlows, and the Options table (where the form name being deleted exists in `GUROPTM_FORM_TO_BE_CALLED`). The deleted form will not appear on the Options of another form.

If you delete a job via GUAOBJS, it is automatically removed from the menu table, GURMENU, unless there are Job Submission records for that job. If the Job Submission records exist, you must delete them first. You can then remove the job via GUAOBJS. If the job existed on a menu, it is automatically removed from the GURMENU table and the GURHMNU table is automatically refreshed.

Security Changes

In addition, the security policy has been changed. Previously, the main menu would not show any objects that a user could not see. However, the user could see those objects in other areas of Banner (QuickFlows, My Banner menus, etc.).

For Release 6.2, security will be provided via the object table, so the user will not be able to see those object anywhere in Banner, with the exception of the GUAOBJS form. Users who can access that form will be able to see all objects there.

Note: These changes apply to Product Menus, not My Banner menus.

New Scripts

GURHMNU1.SQL

This script creates the GURHMNU table.

GURHMNU2.SQL

This script creates the index for the GURHMNU table, `gurhmnu_key_table`.

GURHMNU3.SQL

This script creates comments for the GURHMNU table.

GURRHMU.SQL

This script refreshes the hierarchical menu table GURHMNU. This script will be migrated to the GENERAL PLUS directory so that subsequent upgrades can refresh GURHMNU when data is updated in the tables GURMENU and GUBOBJS.

GUTMENU3.SQL

This script changes the table comment for the global temporary table GUTMENU to indicate that it is associated with Product menus instead of Personal menus.

Changed Scripts

genutil.sql

A new routine, `product_installed`, was added to evaluate the installed indicators on the Institutional Description Table (GUBINST) to make sure that the menus at your institution display only the Banner modules you have purchased.

genview.sql

The views guvmmu.sql and guvpemnu.sql were removed from this script because they are obsolete with this release.

gspauth.sql

The `g$_check_authorization_fnc` routine was changed to improve performance.

New Table

General Hierarchical Menu Table (GURHMNU)

This table stores the names of objects to be presented on the hierarchical tree menu that appears when you log on to Banner. The General Menu Form (GUAGMNU) uses this information to build the Banner main menu.

Column Name	*Data Type*	*Null?*	*Description*
GURHMNU_LEVEL	NUMBER	Not Null	Value designating the hierarchical level.
GURHMNU_OBJT_CODE	VARCHAR2(10)	Not Null	Type of the object (form, menu, etc.).
GURHMNU_VALUE	VARCHAR2(30)	Not Null	Actual name of the object.
GURHMNU_DESC	VARCHAR2(80)	Not Null	Description of the object.
GURHMNU_SEQ_NO	NUMBER	Not Null	Internal sequence number used to build the hierarchy.
GURHMNU_PRIOR_OBJ	VARCHAR2(30)		Name of the object's parent.

Obsolete Views

GUVMMNU

This view, for the Products Menu, is now obsolete. It was introduced in General Release 6.0.

GUVPEMNU

This view, for the My Banner Menu, is now obsolete. It was introduced in General Release 6.0.

New Package

GSPOSEC.SQL

This script applies the security policy to the object table so users can only see the objects that they have the authority to access.

It uses Oracle's Fine Grained Access feature to remove forms and processes so they are not presented to users who do not have the authority to access them.

Changed Package

gukmenu.sql/gukmen1.sql

A `p_menu_refresh` procedure was added to this package to update the GURHMNU table whenever a change is made to the GUAOBJS form or the GUTGMNU form.

The product menu procedure was changed to use the new hierarchical menu table which contains the product menus in hierarchical format.

Logic was also added to get the object description from GURHMNU for submenu items displayed on the personal menu.

Obsolete Package

GSPMSEC.SQL

This database package applied the security policy to the General Menu Repeating Table (GURMENU). Users would be prevented from seeing objects on the Banner Main Menu that they did not have the authority to access. However, users could see those objects elsewhere in the system. This script has been replaced by GSPOSEC.SQL.

Changed File

GREADME.DOC

This file was changed to include the new GURRHMU.SQL script.

Changed Object

Menu Bar (GUMAPPL.MMB)

When you log on and the menu form is loaded, the default setting for the Menu bar items was enabled. For Release 6.2, the default value for certain Menu bar items are disabled. This eliminates the need for the Welcome to Banner, Initializing System Form (GUAINIT) to disable them, thereby reducing the form's start-up time.

Section XIII Views for Operational Data Store Processing - Technical

Overview

The delivered composite views collect SCT Banner source data necessary to populate and maintain data stored in the SCT Operational Data Store (ODS) 2.0 product. These views follow conventional Object:Access standards and are constructed to meet the reporting demands of the ODS.

If you install the ODS product, the ODS extracts SCT Banner transactional system information using these source views to update reporting information that resides on a separate ODS database. The ODS product provides a data structure for the production of operational and custom reports.

Note: Do not alter the structure of these composite views, or add security to the views, since an altered view may adversely impact SCT Operational Data Store incremental refresh maintenance processes.

Although these views are designed for use with ODS, you can use them without it. They will provide you with information just like any of the other views. If you will not be purchasing the ODS product, you *can* alter and enhance them as you wish.

New Composite Views

The following General composite views are included with this release. Refer to the *General Object:Access Reporting Guide* for a complete description of these views.

Composite View	*Description*
Communication Data (AG_COMMUNICATION)	Provides the ability to access information about communication materials, such as the PIDM, application number, academic year, letter code, and quantity of materials sent.
Event Data (AG_EVENT)	Provides the ability to access event information, such as the type code, agency, contact, campus, and function.

Composite View	*Description*
Geographic Region (AG_GEOGRAPHIC_REGION)	Provides the ability to access geographic information, such as the region and division.

New Common ODS Views

The following ODS views are common and can be used by all Banner products. Information on these views is included in the *General Object:Access Reporting Guide* and the *Student Object:Access Reporting Guide*.

Composite View	*Description*
Address (AS_ADDRESS)	Provides the ability to view data related to a person's address including address type, preferred address, EQI equivalent values, IPEDS codes, and international information.
Hold (AS_HOLD)	Provides the ability to access information related to the types and duration of holds associated with a student.
Institution (AS_INSTITUTION)	Provides the ability to access information related to the source/background details for an institution, including type, address, contact, and EDI information.
Medical Information (AS_MEDICAL_INFORMATION)	Provides the ability to access information related to the medical data for a person.
Organization Entity (AS_ORGANIZATION_ENTITY)	Provides the ability to access information related to the organization or non-person for external communication.
Person (AS_PERSON)	Provides the ability to access detailed information related to the person record, including personal/biographic, veteran/ military, citizen/ethnic, medical/ disability, email, and international/visa information.

Composite View	*Description*
Telephone (AS_TELEPHONE)	Provides the ability to access information related to the person's phone information such as phone number, phone address type, and primary number.
Year Type Definition (AS_YEAR_TYPE_DEFINITION)	Provides the ability to access information related to the year type for financial aid and academic data for an aid period, an academic period, or a specific year with century.

New GTVSDAX Rule

A Crosswalk Validation Form (GTVSDAX) rule is used with the AS_PERSON view to identify which Banner address type is designated as the preferred address. This rule is delivered via a Banner General script with the seed data shown below. The **External Code** value can be changed at your institution to reflect the preferred address type, using a valid address type value from the Address Type Code Validation Form (STVATYP).

External Code	*Internal Code*	*Internal Code Sequence Number*	*Internal Code Group*	*Description*	*Activity Date*
PR	PREFADDR	1	ADDRESS	The preferred address type	Sysdate
AWAC	ACDC	1	TRANS_HISTORY_LOAD	First parameter for ODS extract	Sysdate
E035	AWCS	2	TRANS_HISTORY_LOAD	Second parameter for ODS extract	Sysdate
FT26	E037	3	TRANS_HISTORY_LOAD	Third parameter for ODS extract	Sysdate
FTI1	FTI1	4	TRANS_HISTORY_LOAD	Fourth parameter for ODS extract	Sysdate
YR20	GLAS	5	TRANS_HISTORY_LOAD	Fifth parameter for ODS extract	Sysdate

New Scripts

gevevt0.sql

This script creates the new AG_EVENT view. The view contains data for the scheduled event being tracked in the system. It returns one row per event/function combination.

govrgn0.sql

This script creates the new AG_GEOGRAPHIC_REGION view.

The geographic region table stores the geographic regions that are associated with the addresses of each person.

guvmal0.sql

This script creates the new AG_COMMUNICATION view.

The communication table stores mailing and communication information for each person.

igtvsdax612.sql

This script populates the Crosswalk Validation Table (GTVSDAX) with the AS_PERSON rule. The AS_PERSON view, a common view, will not work properly without this rule.

For more information, please refer to the *SCT Banner General Release 6.1.2 Upgrade Guide.*

sovins0.sql

This script creates the AS_INSTITUTION view. The key for this view is the internal ID generated by Banner (PIDM). The view returns one row per institution.

The institution table contains all data that is common to institutions across all administrative models and products. It contains only the most current data; it does not retain any data as history.

sovorg0.sql

This script creates the AS_ORGANIZATION_ENTITY view. The key for this view is the internal ID generated by Banner (PIDM). The view returns one row per organization.

The organization table contains all data that is common to organizations across all administrative models and products. It contains only the most current data; it does not retain any data as history.

sovyer0.sql

This script creates the AS_YEAR_TYPE_DEFINITION view. The key for this view is the year type definition. The view returns one row for each academic year or aid period per type of year. This view contains the information necessary to provide your institution with the year maps required for reporting.

spvadd0.sql

This script creates the AS_ADDRESS view. The key for this view is the internal ID generated by Banner (PIDM) and the address type. The view returns one row per active address for each type. It contains all address information.

spvhld0.sql

This script creates the AS_HOLD view. The key for this view is the internal ID generated by Banner (PIDM), hold types and dates. The view returns one row for each combination. It contains all the holds for one person.

spvmed0.sql

This script creates the AS_MEDICAL_INFORMATION view. The key for this view is the internal ID generated by Banner (PIDM) and the medical code. The view returns one row for each combination. It contains all the medical and disability information for one person.

spvper0.sql

This script creates the AS_PERSON view. The key for this view is the internal ID generated by Banner (PIDM). The view returns one row for each person.

The person table contains all the demographic information for one person across all administrative models and products. It contains only the most current data; it does not retain any data as history.

spvtel0.sql

This script creates the AS_TELEPHONE view. The view returns information about a telephone entry in the database.

Changed Scripts

comview.sql

This common script was modified to contain the ODS common views:

- sovins0.sql
- sovorg0.sql
- sovyer0.sql
- spvadd0.sql
- spvhld0.sql
- spvmed0.sql
- spvper0.sql
- spvtel0.sql

gendbpr.sql

This script now contains the package body and specification for the General product functions for ODS: gokodsf.sql and gokods1.sql.

genview.sql

This script now contains the new General Object:Access views:

- gevevt0.sql
- govrgn0.sql
- guvmal0.sql

New Package

gokodsf.sql/gokods1.sql

This is the main package that calls all the new ODS views. It contains all the database stored functions used to extract data from Banner General and common tables. These functions are used in the ODS views to identify concepts, perform calculations, and supply ROWID access to the views.

This page intentionally left blank

Section XIV Support for e-Procurement - Technical

SCT is involved in an initiative to facilitate the inter-connectivity of Banner with other applications outside Banner. These applications will be able to create, retrieve, update, and delete information directly in Banner tables.

For this release, three APIs have been provided as part of this initiative. They are:

- Identification
- Address
- Telephone

The process is as follows:

1. The application (e.g., Oracle Forms, C) validates the data as usual.
2. The application calls the API to perform the operation (Create, Update, or Delete).
3. The API performs field-level validations and any other business logic required.
4. If there are no problems, the DML operation is completed (the data is inserted, changed, or deleted). If messaging is enabled, data is prepared for the messaging system so it can generate the XML message. Control then passes back to the application.
5. The application issues a Commit command to the Common API, which performs a database commit of the transaction and signals that the XML message should be sent to any external application that has subscribed to listen for these changes.

If the API detects an error condition, an Exception is raised, no DML operations are performed, and no message data is prepared. An error message is sent back to the application as an Oracle exception, and the application formats and displays the message for the user. The application then rolls back the data and discards any previously-prepared XML message data for the transaction.

New Scripts

igurmesg611.sql

This script populates the GURMESG table with seed data for the APIs. Each row controls whether messaging is enabled for each entity represented by an API.

spraddr1.sql

This script adds the SPRADDR_DATA_ORIGIN column to the SPRADDR table.

spraddr2.sql

This script adds the column comment for the SPRADDR_DATA_ORIGIN column.

spriden3.sql

This script adds the SPRIDEN_DATA_ORIGIN column to the SPRIDEN table.

spriden4.sql

This script adds the column comment for the SPRIDEN_DATA_ORIGIN column.

sprtele1.sql

This script adds the SPRTELE_DATA_ORIGIN and SPRTELE_USER_ID columns to the SPRTELE table.

sprtele2.sql

This script adds the column comments for the SPRTELE_DATA_ORIGIN and SPRTELE_USER_ID columns.

Changed Scripts

bannersupport60101.jar

The Java class file names in this Java archive were changed to make them more descriptive. Also, fail-over support was improved for the publication of synchronized messages.

Note: This Java archive was originally called bannersupport.jar. The name was changed for this release.

gendbpr.sql

This script has been modified to create the API packages for Identification, Address and Telephone.

genform.bat

This script has been modified to generate the new GUAERRM form.

genform.shl

This script has been modified to generate the new GUAERRM form.

gokb_messaging1.sql

The Java class reference `com.sct.messaging.MessagingSupport` was changed to `com.sct.messaging.gateways.MessagingSupport`.

New Packages

gokb_addr_rule0.sql/gokb_addr_rule1.sql

This package body and specification together contain additional rules used in the Address API. The body and specification are part of the API layer.

gokb_addr_strg0.sql

This package contains the text strings used in the Address API. The business logic and rules specify the conditions in which the status messages, error messages, etc., in this package are displayed. The messages are displayed in the new SCT Banner Error Messages Form (GUAERRM). This package is part of the API layer.

Note: The messages refer only to API processing. No edits have been removed from the forms yet, so most of the error messages your users see at this time will be generated by the forms.

gokb_address0.sql/gokb_address1.sql

This package body and specification together contain the common business logic and rules to support the Address API. The body and specification are part of the API layer.

gokb_common0.sql/gokb_common1.sql

This package body and specification together contain the common business logic and rules to support the three APIs. The body and specification are part of the API layer.

gokb_ident_rule0.sql/gokb_ident_rule1.sql

This package body and specification together contain additional rules to support the Identification API. The body and specification are part of the API layer.

gokb_ident_strg0.sql

This package contains the text strings used in the Identification API. The business logic and rules specify the conditions in which the status messages, error messages, etc., in this package are displayed. The messages are displayed in the new SCT Banner Error Messages Form (GUAERRM). This package is part of the API layer.

Note: The messages refer only to API processing. No edits have been removed from the forms yet, so most of the error messages your users see at this time will be generated by the forms.

gokb_ident0.sql/gokb_ident1.sql

This package body and specification together contain the common business logic and rules to support the Identification API. The body and specification are part of the API layer.

gokb_tele_rule0.sql/gokb_tele_rule1.sql

This package body and specification together contain additional rules to support the Telephone API. The body and specification are part of the API layer.

gokb_tele_strg0.sql

This package contains the text strings used in the Telephone API. The business logic and rules specify the conditions in which the status messages, error messages, etc.,

in this package are displayed. The messages are displayed in the new SCT Banner Error Messages Form (GUAERRM). This package is part of the API layer.

Note: The messages refer only to API processing. No edits have been removed from the forms yet, so most of the error messages your users see at this time will be generated by the forms.

gokb_telephone0.sql/gokb_telephone1.sql

This package body and specification together contain the common business logic and rules to support the Telephone API. The body and specification are part of the API layer.

gokd_common0.sql/gokb_common1.sql

This package body and specification together contain the common business logic and rules to support the three APIs. The body and specification are part of the DML layer.

gokd_spraddr0.sql/gokd_spraddr1.sql

This package body and specification together contain the common business logic and rules to support the Address API. The body and specification are part of the DML layer.

gokd_spriden0.sql/gokd_spriden1.sql

This package body and specification together contain the common business logic and rules to support the Identification API. The body and specification are part of the DML layer.

gokd_sprtele0.sql/gokd_sprtele1.sql

This package body and specification together contain the common business logic and rules to support the Telephone API. The body and specification are part of the DML layer.

gokroles0.sql/gokroles1.sql

This package body and specification together contain the programming logic that Banner will use to regulate roles in the three APIs.

gvkb_gtvcurr0.sql/gvkb_gtvcurr1.sql

This package body and specification together contain the common business logic and rules to validate currency information. The body and specification are part of the API layer.

gvkb_gtvntyp0.sql/gvkb_gtvntyp1.sql

This package body and specification together contain the common business logic and rules to validate name type codes. The body and specification are part of the API layer.

gvkb_stvasrc0.sql/gvkb_stvasrc1.sql

This package body and specification together contain the common business logic and rules to validate address source codes. The body and specification are part of the API layer.

gvkb_stvatyp0.sql/gvkb_stvatyp1.sql

This package body and specification together contain the common business logic and rules to validate address type codes. The body and specification are part of the API layer.

gvkb_stvcnty0.sql/gvkb_stvcnty1.sql

This package body and specification together contain the common business logic and rules to validate county codes. The body and specification are part of the API layer.

gvkb_stvnatn0.sql/gvkb_stvnatn1.sql

This package body and specification together contain the common business logic and rules to validate nation codes. The body and specification are part of the API layer.

gvkb_stvstat0.sql/gvkb_stvstat1.sql

This package body and specification together contain the common business logic and rules to validate state/province codes. The body and specification are part of the API layer.

gvkb_stvtele0.sql/gvkb_stvtele1.sql

This package body and specification together contain the common business logic and rules to validate telephone type codes. The body and specification are part of the API layer.

Changed Library

goqrpls.pll

A new routine, `G$_DISPLAY_ERR_MSG`, has been added display error messages from the API in the new SCT Banner Error Messages Form (GUAERRM). When the API routines which need to display error messages call it, they will pass the messages to it in a parameter list. The new routine will call GUAERRM and display the messages in the form. This routine can be used for other messages, too.

This page intentionally left blank

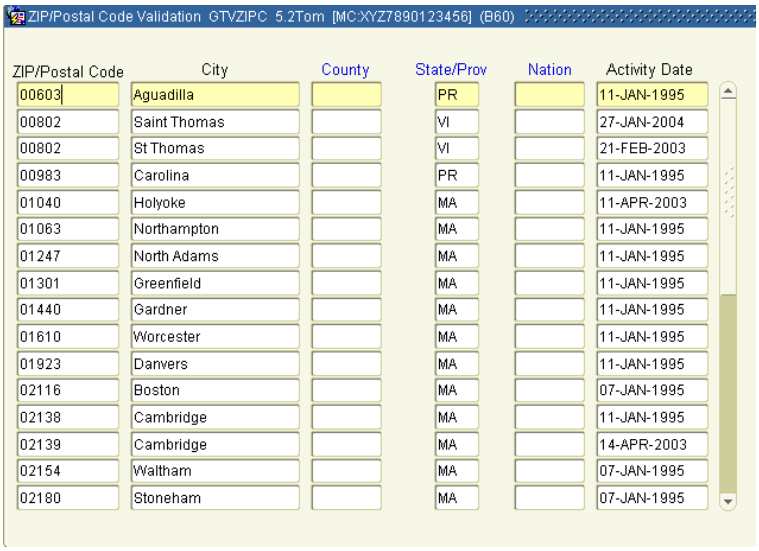
Section XV Miscellaneous Enhancements

Custom Modification Tracking Enhancement

A number of institutions prefer to customize Banner forms. However, this may make it difficult to know if the form being displayed is the SunGard SCT-delivered version or the custom version.

For Release 6.2, Banner will display a modification ID number in brackets on the title bar of the form.

This enhancement does not perform version control of any kind; it merely identifies the form as being custom.



The modification number also appears on the About SCT Banner Form (GUAABOT).

Note: This fulfills RPE #35391.

Tasks

To enable this feature for your custom forms:

1. Access the General User Preferences Maintenance Form (GUAUPRF).
2. Select the **Display Release Number on Title Bar** check box, if it has not been selected already.
3. Save your changes.

   Note: The modification number will not appear on the title bars if you clear the check box on GUAUPRF. However, it will still appear on GUAABOT.

4. In the form in which you want to enable this feature:

   (a) Add a `text_item` to the `FORM_HEADER` block.

   (b) Name this `text_item` *`MOD_ID`*.

   (c) Set the *Enabled Property* of `MOD_ID` to *No*.

   (d) Set the *Max Length* to *16*.

   (e) Provide a value from 1 to 16 alphanumeric characters for the `FORM_HEADER.MOD_ID` in the form-level trigger `LOAD_CURRENT_RELEASE`.

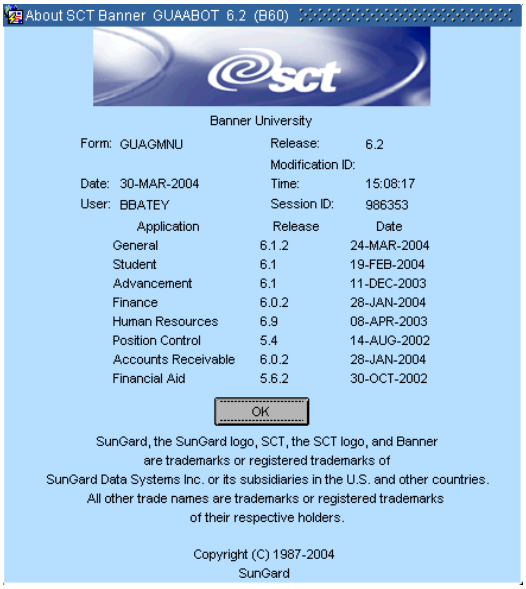
5. Save your changes.

The value you entered will appear in the title bar of that form and on GUAABOT.

Changed Forms

About SCT Banner (GUAABOT)

A new label, **Modification ID**, has been added to this form.



When you select About SCT Banner from Help on the Menu bar, you can see the modification ID in that field, if the form is customized and has been set up correctly (see *Tasks* earlier in this section).

Welcome to Banner, Initializing System Form (GUAINIT)

The programming logic behind this form has been modified to include a new global variable `GLOBAL.MOD_ID`. This global will hold the value of `FORM_HEADER.MOD_ID` so it can be passed from a customized form to the About SCT Banner Form (GUAABOT). The global receives its value when you call GUAABOT from the Help Menu bar while running a customized form.

Changed Library

GOQRPLS

The procedure `G$_SET_INST_PROPERTY` in GOQRPLS.PLL was changed to append the value in `FORM_HEADER.MOD_ID` into the window title bar if `FORM_HEADER.MOD_ID` exists (and is not null), and the title bar has been set up to display the release number (see *Tasks*).

Changed Procedure

G$_SET_INST_PROPERTY

The procedure `G$_SET_INST_PROPERTY` in GOQRPLS.PLL was changed to append the value in `FORM_HEADER.MOD_ID` into the window title bar if `FORM_HEADER.MOD_ID` exists (and is not null), and the title bar has been set up to display the release number (see *Tasks*).

Changed Object

Menu Bar (GUMAPPL.MMB)

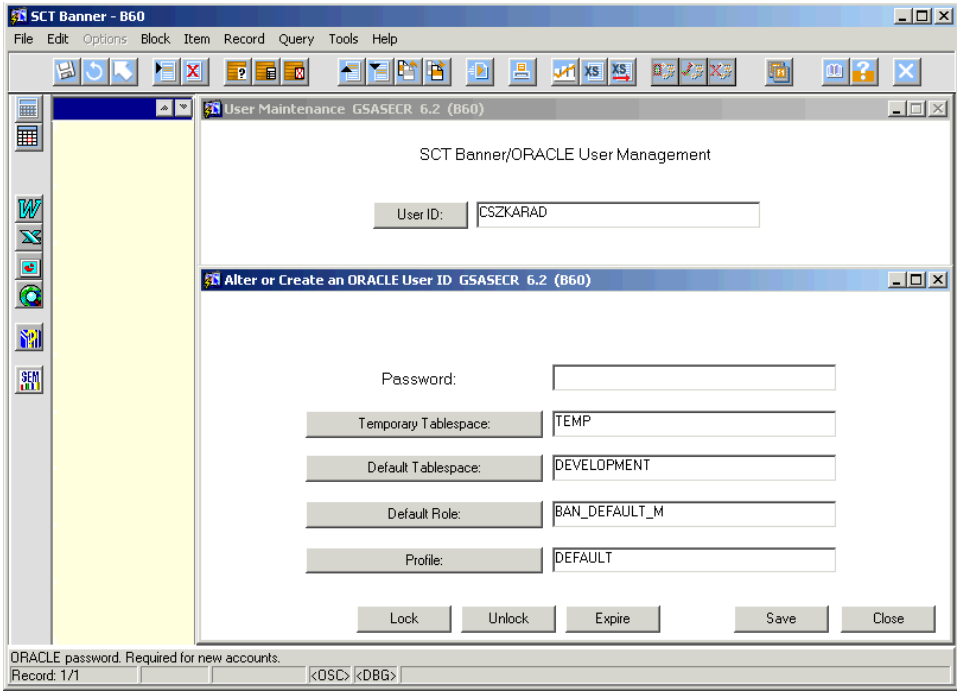
When you call the GUAABOT form from the Help Menu bar item, GUMAPPL.MMB will now populate `GLOBAL.MOD_ID` with the value in `FORM_HEADER.MOD_ID`, if `FORM_HEADER.MOD_ID` exists. If it does not, GUMAPPL.MMB will clear `GLOBAL.MOD_ID`.

When you access GUAABOT, the value of `GLOBAL.MOD_ID` is displayed as the modification ID.

Security Maintenance Form (GSASECR)

For this release, you can now assign an Oracle profile to a user. The Profile button was added to the User Maintenance window of the GSASECR form. If you select it, you receive a list of available user profiles.

These profiles should be set up in Oracle; they are stored in an Oracle Data Dictionary View. For more information, please refer to your Oracle documentation.



In addition, the synchronization logic has been changed. This logic synchronizes classes and class users when, for example, class definitions have changed and a class user may not have all the correct Oracle role grants to run every object in that class. Previously, you had to synchronize one class at a time; now it will synchronize all the classes at once.

Previously, menu classes that had no objects would always trigger a message that they were out-of-synch. For this release, these objects are never out-of-synch.

Support for Human Resources 6.1 Release

Banner General is delivering changes to a package to support the Banner Human Resources 6.1 Release. This package was first delivered in the Banner General 6.1.2 Release. It is related to Operational Data Store (ODS) processing.

Changed Package

gokodsf.sql/gokods1.sql

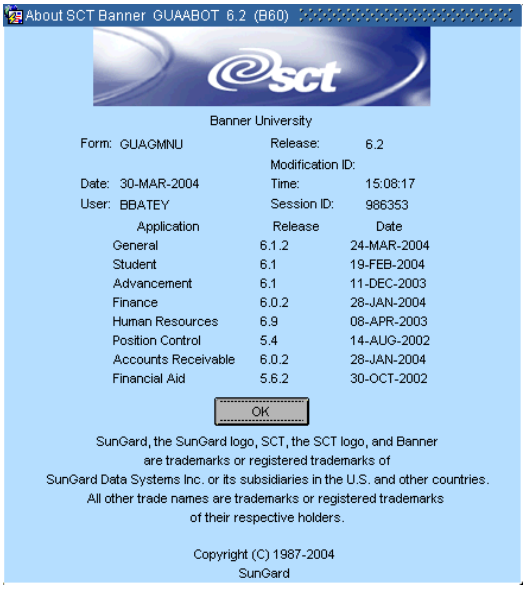
This is the main package that calls the ODS views. It has been changed to use native dynamic SQL. The function `F_GET_DESC` now uses the new variables `SYSTEM_IND` and `SECOND_PARAMETER`. In addition, the package has been changed so multiple rows are not returned in `F_GET_DESC` when a table has more than one primary key.

Updated Trademark Information

Changed Form

About Banner Form (GUAABOT)

This form has been changed to include new information about SunGard SCT.



Event Queue Record Maintenance Form (GOAEQRM)

This form has been changed to display the information about the most recent event at the beginning of the form. Previously, the most recent event appeared at the bottom of the event list and users had to scroll down to see it.

Parameter Group Definitions Form (GOREQPG)

The **Group Code** field could hold up to 30 characters. However, the column in the table, `GOREQPG_EQPM_CODE`, can hold up to 100 characters. For this release the **Group Code** field was changed to hold up to 100 characters.

In addition, a change was made to the Data block. Previously, users could enter a parameter name and save, which would generate an Oracle error. Now the **Seq #** (Sequence Number) field is required, removing the opportunity for the error.

SEVIS Information Form (GOASEVS)

The **Personal Funds** field on the Common Data block has been modified to accept the value *0*.

Also, the autohint on for the **Academic Term** field on the Student Data block has been changed to *Number of months used to calculate expenses.*

Changes to the Extended Query Feature

Previously, Banner allowed you to execute an SQL query from a window in Banner to find the information you want more quickly. You could use the SQL operators to create a more specific query. Because the queries were so powerful, there has been an interest in increasing the security for this feature.

With this release, a new object has been added to the Banner security tables. You can use this object, `EXTENDED_QUERY`, to regulate access to the extended query feature.

Note: If you will be using the new object to restrict access to the extended query feature, you must regenerate all your forms. If you will not be using the new object, you do not need to regenerate them.

Welcome to Banner, Initializing System Login Form (GUAINIT)

Logic was added to the `WHEN-NEW-FORM-INSTANCE` trigger on this form to determine if the user is authorized to perform an extended query. The `WHEN-NEW-FORM-INSTANCE` trigger is used in the `G$_GOQOLIB_KEY_TRIGGER.ON_SELECT` routine, which is executed when an execute query command is fired.

If the user tries to use the extended query feature and does not have the appropriate authority, Banner will display a warning message and it will cancel the query.

EXTENDED_QUERY

This object, a global, has been added to the Banner security tables. As delivered, it is granted to the General class so all users at your institution can continue to use the extended query. If you want to restrict access to that feature, remove the object from

the General class and grant access through a more restrictive class or directly to specific users.

goo_string_nt.sql

This script was modified to include the new grant to the General class and synonym (`STRING_NT`). This is the named type definition for Batch APIs.

goqolib.fmb

An `ON-SELECT` trigger was added to `G$_FORM_CLASS` to perform extended query checking during the execute query processing.

goqrpls.pll

A new routine, `G$_GOQOLIB_KEY_TRIGGER`, has been added to determine if the user has performed an extended query. It evaluates a new global, `EXTENDED_QUERY`, to determine if the user's access should be limited. If `EXTENDED_QUERY` has been set, the user will receive a warning message if they try to use the extended query feature and the query will be cancelled.

Native Dynamic SQL

Security packages have been modified to use Native Dynamic SQL, an Oracle feature. This change simplifies logic and improves performance.

Oracle had implemented Native Dynamic SQL in a previous release of its database system. Previously, the security packages in Banner had been using the older version of Dynamic SQL via the DMBS_SQL package.

genutil.sql

This package has been modified to replace the document commands with comment lines. The audit trail has been moved outside the package creation and the DMBS_SQL commands have been replaced with Native Dynamic SQL commands.

gspauth.sql

This package has been modified to replace the document commands with comment lines. The audit trail has been moved outside the package creation and the DMBS_SQL commands have been replaced with Native Dynamic SQL commands.

gspsecr.sql

The Dynamic SQL in this script has been modified to use Native Dynamic procedures. In addition, logic was added to the `change_password` routine to allow only specified users to change other users' passwords.

Changes for Messaging

goqolib.fmb

An `ON-ROLLBACK` trigger and a `POST-FORMS-COMMIT` trigger were added to `G$_FORM_CLASS` to make the forms that were regenerated messaging-aware. The `ON-ROLLBACK` trigger issues the discard command and the `POST-FORMS-COMMIT` trigger issues publish commands. In addition, `ON_ROLLBACK` and `POST_FORMS_COMMIT_TRG` were added to `G$_FORM_CLASS` for local usage.

`KEY-NXTREC` in `G$_FORM_CLASS` was changed to correct a typographical error.

Support for APIs

Changed Library

goqrpls.pll

A new routine, `G$_DISPLAY_ERR_MSG`, builds a parameter list to hold any error messages, then displays those messages in the new GUAERRM form. The messages are passed as parameters to `G$_DISPLAY_ERR_MSG`. This routine can be used for messages other than those generated by APIs.

Changed Script

guastdf.h

The `POSTORA_API` macro and API error message globals were added to this script. In addition, the `p_commit` and `p_rollback` procedures were added.

This page intentionally left blank

Section XVI Problem Resolutions

This section lists the problem resolutions included in Releases 6.1.1, 6.1.2.1, and 6.2.

Population Selection

gjapctl.fmb

(#66687)
When you ran GLBDATA from the Process Submission Controls Form (GJAPCTL) for a population selection created on the Population Selection Definition Rules Form (GLRSLCT), you could receive a warning message if the application/selection combinations were the same but the Creator IDs were different and there were different dynamic parameters. The message was: **WARNING* Dynamic parms exist at the Application level or Variable level that are not compiled into the selection ID(s). You should recompile the selection ID(s). New parms are: &SAISUSR_NAME.* The message should not have appeared, and the dynamic parameters listed were incorrect.

The cursor `GET_SLCT_PARMS` in `LOAD_GLBDATA_COL_INFO` was changed to select dynamic parameters by application, selection, and Creator ID. Resolved in Release 6.1.1.

Letter Generation

glbparm.shl

(#92535)
This object was modified to remove Linux reserved words. Resolved in Release 6.1.2.1.

glrletr.shl

(#92535)
This object was modified to remove Linux reserved words. Resolved in Release 6.1.2.1.

glolett.pco

(#92206)
When running the Automatic Letter Compilation Process (GLOLETT) either through the Population Selection Definition Rules Form (GLRSLCT) and the Variable Rules Definitions Form (GLRVRBL), or as a batch process, institutions could receive the error message **ERROR* Undefined column* and the process would fail. The COBOL STRING command overlays input values into the receiving field. Because two receiving fields in this process were not explicitly initialized to spaces in error, this caused a problem when the Fujitsu COBOL implicitly initialized them to unpredictable values. Those receiving fields are now explicitly initialized to spaces. Resolved in Release 6.1.2.1.

Note: Although this problem appeared only in the Linux environment, the changed code is valuable for institutions using other platforms.

General International Management

goasevs.fmb

(#89176)
When you entered the employer's name (rather than their ID) and pressed the Enter key, the name disappeared. If you entered the data again and pressed the Enter key, it would disappear again. If you then created a Student record and viewed the results on the SEVIS Transmittal History Form (GOASEVR), you would have seen the employee's name twice. This was because the **Employer Name** field is really an Edit box, like the comment fields. When users pressed Enter to move to the next field, they would actually move to a new line in the same field.

The field has been enlarged to contain two visible lines and a scroll bar. If users press the Enter key while in that field, they can see the cursor move to the next line. Resolved in Release 6.1.1.

goisevs.fmb

(#89698)
This form displayed query results in ascending order by sequence number (**Seq**), but records in Banner should appear in descending order with the most recent record (the highest number) first. GOISEVS now displays records in descending order. Resolved in Release 6.1.1.

goksvi1.sql

(#88346)
Whenever either `GORSEVS_USA_CITY` or `GORSEVS_NONUSA_CITY` contained a city name that was longer than 16 characters and `GORSEVS_SVTS_CODE` contained *C*, an additional record was created even though nothing had been changed. Resolved in Release 6.1.1.

(#89422)
Previously, the `StudentRegistration` structure could only contain *I*, indicating that the student who registered was an *Initial* student. `StudentRegistration` can now contain *C* to indicate *Continued Attendance*. Resolved in Release 6.1.1.

gorseve.pc

(#91900)
Banner had not included an interface to the fsaATLAS product. This is described in more detail previously in this document. Resolved in Release 6.1.2.1.

(#90149)
In the UNIX (Solaris) environment, GORSEVE did not move `AcademicTerm` into the SEVIS Collector Table (GOTSVBT) so it could be processed by the SCT SEVIS Transfer Adaptor (SEVISTA). As a result, `AcademicTerm` was not included in the XML file sent to the Bureau. Resolved in Release 6.1.1.

gorsvsq.fmb (#91020)
The TO_DATE used in the date comparison was coded as *31-DEC_2099*, in error. It has been changed to *31-DEC-2099*. Resolved in Release 6.1.1.

System Functions and Administration

goamedi.fmb (#90193)
This form is designed to determine if the person entered in the Key block is a student, an employee, or both. If the person is an employee, the HR security routines are invoked, based on the rules established on the Installation Rules Form (PTRINST). Previously, the form did not have sufficient Human Resources security access routines to retrieve all the required data. Therefore, it restricted all access to data if the person in the Key block was an employee, even if the user had been granted the appropriate permissions. The trigger KEY_BLOCK.CHECK_KEYS has been changed to call the procedure GET_PEBEMPL_CODE to provide the required data. Resolved in Release 6.1.1.

gtvscod.fmb (#80601)
You had been prohibited from entering rows into the data block. For this release the block and form triggers Key-CreRec, Key-DupRec, and Key-DelRec have been deleted. You can now insert, delete, and duplicate rows. Resolved in Release 6.1.1.

guauprf.fmb (#89877)
Previously, if you changed the information for **Form Name Display Option** in Internet-native Banner, the change was not saved. Now, the procedure Display has been added to the Program Unit IO package body and specification to allow the main body of the code to populate the DISPLAY_OPTIONS. Also, IO.DISPLAY was added to the triggers WHEN-NEW-FORM-INSTANCE, DISPLAY_OPTION.WHEN-CHECKBOX-CHANGED, and BUTTON_CONTROL_BLOCK.DISP_CALL_BTN.WHEN-BUTTON-PRESSED. Resolved in Release 6.1.1.

gukmen1.sql (#90189)
After General Release 6.1, items on personal menus were displayed alphabetical order, in error. Previously, items on personal menus would be listed in the order in which they appeared on the My Banner Maintenance Form (GUAPMNU); they would be sorted by GURMENU_SORT_SEQ. Now the items are again sorted by GURMENU_SORT_SEQ. Resolved in Release 6.1.1.

Event Management

geaattd.fmb (#81176)
When you executed a query in the data block, all the records would be returned, not just the ones that met your criteria. The trigger `GERATTD.Pre-Query` was added, modifying the block WHERE clause.

Note: If your query uses an ID field, and an ID exists that does not meet the criteria but has a previous ID that does match it, the record with the new ID will be displayed (in addition to the one with the old ID), even though it does not match the criteria.

Resolved in Release 6.1.1.

Business Rule Builder

gorrsql.fmb (#91020)
The `TO_DATE` used in the date comparison was coded as *31-DEC_2099*, in error. It has been changed to *31-DEC-2099*. Resolved in Release 6.1.1.

gorsqpa.fmb (#91020)
The `TO_DATE` used in the date comparison was coded as *31-DEC_2099*, in error. It has been changed to *31-DEC-2099*. Resolved in Release 6.1.1.

Miscellaneous Problem Resolutions

banfjsv.pl (#92535)
This object was modified to remove Linux reserved words. Resolved in Release 6.1.2.1.

gjajobs.shl (#92535)
This object was modified to remove Linux reserved words. Resolved in Release 6.1.2.1.

gokb_common1.sql (#92857)
The database package `GOKB_COMMON.SQL` was delivered in General Release 6.1.1. It uses the Oracle sequencers `ID_SEQUENCE` and `PIDM_SEQUENCE` to generate new Banner IDs and PIDMs, respectively, rather than using the existing SOBSEQN table. However, the SOBSEQN table could still be used. There was no programming logic

in place to prevent the same ID or PIDM from being created twice, once by the existing table and again by the new package.

Note: The only forms that could use the new package are the Finance Vendor Form (FTMVEND) and the Finance Personal Information Form (FOAIDEN) that were delivered in Finance Release 6.0.1.

`GOKB_COMMON1.SQL` replaces `GOKB_COMMON.SQL`. It contains the new functions `f_generate_id` and `f_generate_pidm`. These functions check the SOBSEQN table whenever a new ID or PIDM is generated, and return a number that is one higher than the highest number on SOBSEQN. The Oracle sequencers `ID_SEQUENCE` and `PIDM_SEQUENCE` will be used in a future release of Banner. Resolved in Release 6.1.2.1.

goqrpls.pll

(#81303)
Banner screen captures for client/server users always include the date and time in the title bar of the printout. As part of issuing the key-print, the date and time are appended to the runtime form's window title bar, the print built-in is invoked, and the date and time are removed from the title bar. However, some Internet-native Banner environments issued the print built-in before the date and time values were placed in the title bar, so the screen capture did not include them.

To correct this, the procedure `G$_GOQOLIB_KEY_TRIGGER.KEY_PRINT` was modified for Internet-native Banner. The Oracle built-in `dbms_lock.sleep` was added after the print built-in and before the `set_date_time (false)` call. Some Internet-native Banner environments require a sleep time of 5 seconds (after selecting *OK* to acknowledge the printing) to make sure the date-time stamp goes to the printer. Other environments only require the sleep built-in itself. Resolved in Release 6.2.

(#85937)
If you used a query on a Banner form to limit the search results that were displayed, and then you performed a data extract with a key, you would not be able to cancel the query or enter a different query before leaving the form. If you tried to enter and execute another query, all the records that met your criteria would not be displayed, and performing a Rollback or Next Block function would not re-display the records. Instead, the results of the original query would be displayed.

The `DEFAULT_WHERE` block property in the `G$_DATA_EXTRACT` procedure body has been changed from `WHERE_CLAUSE_SAVE` to `WHERE_CLAUSE_ORIG`. Resolved in Release 6.1.1

gsasecr.fmb

(#93382)
If you called this form directly in Internet-native Banner and double-clicked in certain fields, the form might crash. The form has been modified so you cannot double-click in any fields. Resolved in Release 6.2.

(#89857)
If you tried to enter this form directly in stand-alone mode, you would get the error

message *FRM-41067 Cannot find Menu Item: invalid ID*. You could access it, however, if you logged on as BANSECR through the Welcome to Banner, Initializing System Form (GUAINIT), then navigated to GSASECR. This has been corrected. Resolved in Release 6.1.1.

gsavpdi.fmb (#89857)
If you tried to enter this form directly in stand-alone mode, you would get the error message *FRM-41067 Cannot find Menu Item: invalid ID*. You could access it, however, if you logged on as BANSECR through the Welcome to Banner, Initializing System Form (GUAINIT), then navigated to GSAVPDI. This has been corrected. Resolved in Release 6.1.1.

gspsecr.sql (#93036)
Due to security concerns, the `change_password` routine was modified. Now only BANSECR users can modify other users' passwords. Resolved in Release 6.1.1.

guastdf.h (#92535)
This object was modified to remove Linux reserved words. Resolved in Release 6.1.2.1.

guavrfy.shl (#92535)
This object was modified to remove Linux reserved words. Resolved in Release 6.1.2.1.

gurjobs.pc (#92535)
This object was modified to remove Linux reserved words. Resolved in Release 6.1.2.1.

gurrddl.sql (#89906)
Logic was added to this object to support global temporary tables. Resolved in Release 6.2.

sctproc_linux.mk (#92535)
This object was modified to remove Linux reserved words. Resolved in Release 6.1.2.1.

sctprocb_linux.mk (#92535)
This object was modified to remove Linux reserved words. Resolved in Release 6.1.2.1.

smrbcmp.shl (#92535)
This object was modified to remove Linux reserved words. Resolved in Release 6.1.2.1.

Appendix A Implementing Banner Messaging for e-Procurement

Overview

Messaging allows clients to develop real-time data messaging functionality to synchronize data across the corresponding applications on their network. Clients can either plan and manage API-to-API development projects, or they can invest in one or more of the products comprising the Luminis Data Integration Suite, a data-messaging solution for higher education which employs an enterprise application integration (EAI) system.

Banner Messaging Support provides APIs that are used by the Banner Common Business API to make a Banner Business Entity *message-enabled*. If a Business Entity is message-enabled, the data associated with it can be associated with one or more Enterprise Messages.

Banner Messaging Support provides a loosely - coupled architecture that insulates Banner from knowing what Business Entities are associated with which Enterprise Messages.

Prerequisites

To successfully implement Banner Messaging for e-Procurement, the following software must be installed, configured, and running on the messaging host platforms:

1. Luminis Data Integration (LDI)
2. LDI for e-Procurement

Implementation Steps

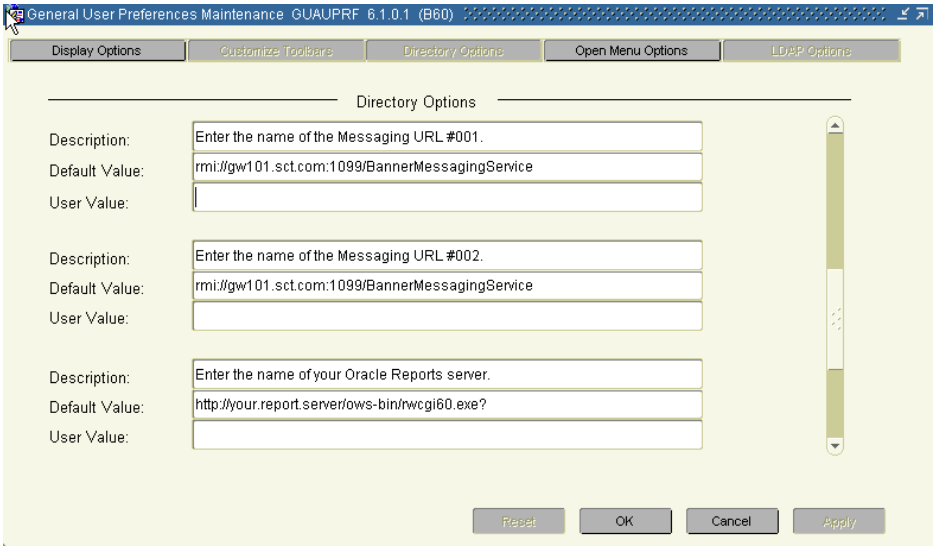
Step 1: Log In as BASELINE

Log in to Banner with the BASELINE user ID.

Step2: Set Up Directory Options on the General User Preferences Maintenance Form (GUAUPRF)

Access the General User Preferences Maintenance Form (GUAUPRF). Select *Directory Options*.

Scroll down until you see the description *Enter the name of the Messaging URL #001*.



Change the URL to reference the primary Banner Gateway Producer application, such as:

```
rmi://messaginghost1.university.edu:1099/BannerMessagingService
```

Find the description *Enter the name of the Messaging URL #002*. Change the URL to reference the secondary Banner Gateway Producer application, such as:

```
rmi://messaginghost2.university.edu:1099/BannerMessagingService
```

The URLs support automatic failover: if Banner Messaging support cannot communicate with the primary Gateway Producer for some reason, it will automatically use the secondary one. If it cannot communicate with the secondary Gateway Producer, Banner Messaging will send an error message to the user.

When you are finished, save your changes.

Technical Note:

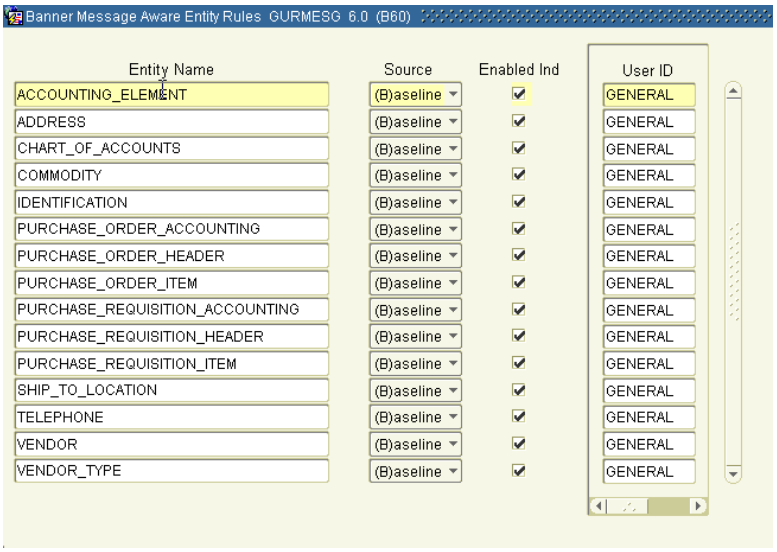
The Banner Gateway Producer application is responsible for generating and publishing Enterprise synchronization messages depending on the changes made to Banner Business Entities.

The information you enter on the GUAUPRF form is stored in the GENERAL.GURUPRF table:

Column Name	*Value*	*Description*
GURUPRF_USER_ID	*BASELINE*	Specifies that this is an SCT baseline row.
GURUPRF_GROUP	*MESSAGING*	Indicates that the record applies to messaging.
GURUPRF_KEY	*RMI_URL*	The key name.
GURUPRF_STRING	*001* or *002*, depending on whether it is the primary or secondary Banner Gateway Producer	Specifies the rank of the URL.
GURUPRF_VALUE	rmi://gw101.sct.com:1099/BannerMessagingService	Specifies the URL to use to locate the Banner Messaging Service.
GURUPRF_SYSTEM_REQ_IND	*Y*	Indicates that Banner requires this record.

Step 3: Enable Business Entities on the Banner Messaging Aware Entity Rules Form (GURMESG)

Access the Banner Messaging Aware Entity Rules Form (GURMESG). It controls whether a Business Entity is messaging-aware or not.



Make sure the Enabled Ind check box is selected for the following:

- *ACCOUNTING_ELEMENT*
- *ADDRESS*
- *CHART_OF_ACCOUNTS*
- *COMMODITY*
- *IDENTIFICATION*
- *PURCHASE_ORDER_ACCOUNTING*
- *PURCHASE_ORDER_HEADER*
- *PURCHASE_ORDER_ITEM*
- *PURCHASE_REQUISITION_ACCOUNTING*
- *PURCHASE_REQUISITION_HEADER*
- *PURCHASE_REQUISITION_ITEM*
- *SHIP_TO_LOCATION*
- *TELEPHONE*
- *VENDOR*
- *VENDOR_TYPE*

These are the Banner Business Entities that are required to support e-Procurement.

Note: In the future, you may have custom Business Entities. If so, they will have the Source *(L)ocal* instead of *(B)aseline*.

When you are finished, save your changes. Now Banner Messaging support will track any changes to these entities as programs call the entities' APIs. During a database commit, any changes will be sent to the Banner Gateway Producer application where Enterprise synchronization messages will be generated and published for each Enterprise message which is associated with the entity.

Because of this synchronization, your institution can manage its information more efficiently and your data administration will be easier.

Technical Note:

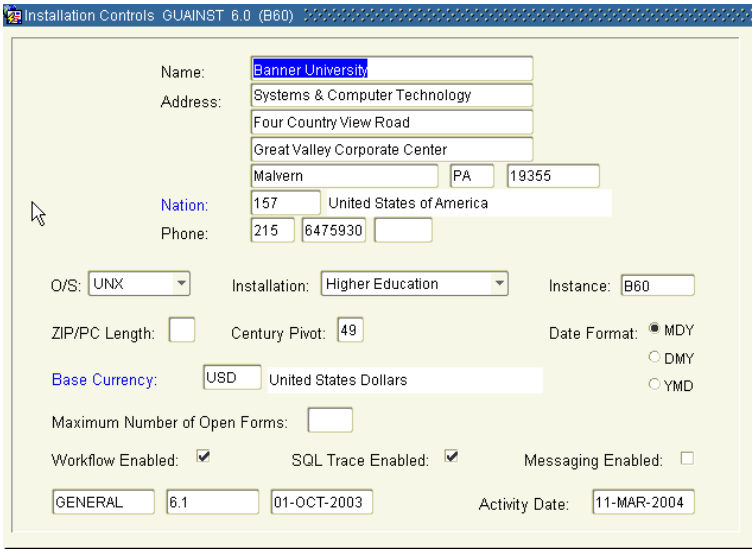
These records are stored in the GENERAL.GURMESG table. Each row defines the Business Entity name, the Source indicator, and the Enabled indicator.

Column Name	*Description*
GURMESG_ENTITY_NAME	Specifies the name of the Business Entity.
GURMESG_SOURCE_IND	Specifies if the Business Entity is SCT-delivered or custom. Valid values are: *B* - Banner *L* - Local, indicates a custom entity

Column Name	*Description*
GURMESG_ENABLED_IND	Indicates that the entity is messaging-aware. Valid values are: *Selected* - Y, the entity is messaging-aware *Cleared* - any other value, including null, the entity is not messaging-aware
GURMESG_USER_ID	The user ID of the person who created or last updated the record.
GURMESG_ACTIVITY_DATE	The date the record was created or last updated.

Step 4: Enable Messaging for All of Banner

Next you must enable messaging for your entire Banner system. Access the Installation Controls Form (GUAINST).



Select the **Messaging Enabled** check box and save your changes. This allows Banner Messaging Support to send business entity changes to the Banner Gateway Producer application. Banner will begin producing Enterprise synchronization messages for the Banner Business Entities that you have defined as messaging-aware.

Technical Note

These records are stored in the GENERAL.GUBISNT table. If the column GUBINST_MESSAGE_ENABLED_IND contains *Y*, messaging is enabled for the entire Banner system. If it contains any other value, including null, messaging is disabled for your system.

Note: The values stored in the GUBINST table override the settings on GURMESG. Clearing the **Messaging Enabled** check box on GUAINST is a quick way to completely turn off messaging in Banner.

Banner Messaging Support will only contact the Banner Gateway Producer if you select the **Messaging Enabled** check box on the GUAINST form and the **Enabled Ind** check box for the individual entities on the GURMESG form (see Step 3). Both conditions must be met for messaging to work.

Step 5: Test

Test the connectivity between your Banner system and the Banner Gateway Producer.

The following will produce a HelloWorld-Create-Sync message:

1. Execute the Oracle SQLPlus application and log in with the BANINST1 user ID.

2. Execute the following statements:

```
set serverout on;
call dbms_java.set_output(20000);
begin
gb_messaging.p_register_entity('$CREATE_HELLO_WORLD',
gb_messaging.CREATE_OPERATION);
gb_messaging.p_add_parameter('$MESSAGE', 'Hello Banner World');
gb_messaging.p_publish;
end;
/
```

The SQLPlus application should display *PL/SQL procedure successfully completed* if no errors were encountered. If any errors are encountered they will appear in the SQLPlus application window.