Banner Enterprise Data Warehouse
Administration Guide

Release 8.4.3

February 2013

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

Prepared by: Ellucian
4375 Fair Lakes Court
Fairfax, Virginia 22033
United States of America

Revision History

Publication Date	Summary
February 2013	New version that supports Banner Enterprise Data Warehouse 8.4.3 software.

Banner Enterprise Data Warehouse 8.4.3
Administration Guide

Contents

1 Overview

This Administration Guide includes information about the warehouse architecture and instructions on how to set up and maintain the warehouse using the Administrative User Interface.

Prior to the EDW 8.4.3 release, there was an EDW Handbook that supported the EDW product. That guide has been divided into two guides:

- *EDW Administration Guide* (this guide) - the Administration Guide includes information intended for use by technical administrators who will set up and maintain the warehouse.
- *EDW User Guide* - the User Guide includes information intended for use by functional end users to work with the warehouse data.

BPRA Products

The complete set of Banner Performance Reporting and Analytics (BPRA) products give your institution the ability to take full advantage of the data stored in your source system by turning it into applied knowledge in the warehouse. The BPRA solution set includes the following products:

- Banner Operational Data Store (Banner ODS)
- Banner Enterprise Data Warehouse (Banner EDW)
- Banner Recruiting and Admissions Performance
- Advancement Analytics for Cognos

Your institution may license some or all of these products. If you do license products in addition to the Banner EDW, it is important that you understand the relationship among all of the products as you use them. You can use the products together to help you make informed decisions, guide strategic institutional planning and forecasting based on analysis of historical trends, and enhance institutional performance.

Banner Operational Data Store

The Banner ODS enables you to extract information from your source systems, reorganize the information into a simplified format, and store the information in the Banner ODS database. End users can then create and deploy operational and ad hoc reports.

Banner ODS provides an extensive and flexible data store and business-organized reporting views with fewer columns and improved performance. You can use these views alone, or in combination with other views.

Banner Enterprise Data Warehouse

The Banner EDW is a multi-dimensional database that gives you a complete picture of your institution's current and past business conditions. The Banner EDW offers comprehensive reporting and analysis capabilities by providing the following data objects:

- Operational stars that you can refresh with current data on a regular basis
- Snapshot stars that offer a historical snapshot of the data at a point-in-time

This combination of current and historical data allows you to do comparative reporting and analysis.

Banner EDW includes Advancement specific components intended for use with the Advancement Analytics for Cognos. It is possible to source these packages from both the Advance and Banner Advancement systems.

Advancement Analytics for Cognos

You can use the Cognos analytics components together with the warehouse to meet your institution's advancement data analysis needs. You can also use other reporting or business intelligence applications with the warehouse to analyze the data.

The Advancement Analytics for Cognos product includes the following types of objects built using the Cognos Business Intelligence application:

- Reports - display trends of outcomes, summaries of current outcomes, and detailed information about constituents
- Dashboards - display several graphical performance charts for a specific business area on a single screen that you can review at a glance
- Scorecards - display institutional goals and objectives including Key Performance Indicators (KPIs) that monitor progress toward your goals and objectives and a set of strategic initiatives that are needed to produce desired outcomes

The data for these objects is stored in the Banner EDW. These objects are intended to illustrate the kind of analysis you can perform on the warehouse data. You can use the reports, dashboards, and scorecards as delivered or you can modify them to reflect the specific information you need to analyze and monitor your institution's progress.

Banner Recruiting and Admissions Performance

Banner Recruiting and Admissions Performance is the reporting analytics and performance portion of the Banner Enrollment Management Suite. You can use Banner Recruiting and Admissions Performance to report, analyze, monitor and track performance; define goals and objectives; and measure progress against key performance indicators. Banner Recruiting and Admissions Performance also lets you easily access recruitment, admissions, and selected financial aid information and use it to create reports.

Banner Recruiting and Admissions Performance solution includes a package of operational reports. These reports display trends of outcomes, summaries of current outcomes and detailed lists of recruits and applicants. The solution also includes capabilities to configure organizational scorecards to reflect your organization's specific recruitment goals, objectives, and Key Performance Indicators (KPIs).

Banner Student Retention Performance

The Banner Student Retention Performance product provides a package of reports and analytic components that support common student retention performance goals and objectives. You can use Banner Student Retention Performance to monitor student retention and student success to satisfy institution goals and objectives; extend and modify performance monitoring capabilities; and create operational reports and ad-hoc queries that meet the specific needs of your institution.

Related documentation

You can refer to the following documents for more details about the Banner EDW components and the related Cognos components if you license them.

EDW User Guide

The *Banner Enterprise Data Warehouse User Guide* includes information about how to set up EDW Extract Parameters and Parameter Maps that define aspects of how data will be loaded into the warehouse and how it will display in reports. The User Guide also includes information about the Banner EDW business concepts and Cognos cubes delivered with the warehouse.

EDW Stars Reference Guide

The *Banner Enterprise Data Warehouse Stars Reference* guide is a manual that supplements information about the Banner EDW. The Stars Reference guide includes summary information about each of the stars in the warehouse with hypertext links to each of the star diagrams. The guide is delivered as zip file that is compilation of PDF files including the *Banner EDW Stars Reference* guide and all of the warehouse star diagrams. You can download the zip file from the Documentation and Download area of the Customer Support Center under the “Cross Product - Enterprise Data Warehouse” product.

EDW Installation and Upgrade Guides

The *Banner Enterprise Data Warehouse Installation and Upgrade Guides* include instructions for installing or upgrading the Banner EDW product.

Cognos components (Handbook available if you license that product)

If you license the Performance products that use the warehouse as the basis for reporting, you will have access to the following guides. (Each Handbook is available if you license that product.)

- *Advancement Analytics for Cognos Handbook*
- *Banner Recruiting and Admissions Performance Handbook*
- *Banner Student Retention Performance Handbook*

Banner Performance Reporting & Analytics Resource Guidelines

The BPRA Resource Guidelines document includes hardware and software recommendations for all of the BPPA products. This document also includes compatibility information describing which Banner, Oracle, and Cognos product releases are compatible with each of the BPRA products.

2 Architecture

The Banner Operational Data Store (Banner ODS) and the Banner Enterprise Data Warehouse (Banner EDW) are the data warehouse components of the Banner Performance Reporting and Analytics Business Intelligence platform. The following sections describe the architecture of this platform and the roles and integration of ODS and EDW with the other components.

BPRA product architecture

The complete suite of BPRA products provides comprehensive content across areas such as student, financial aid, finance, accounts receivables, human resources and advancement giving your institution the ability to take full advantage of the data stored in your source system by turning it into applied knowledge in the warehouse. You can use the BPRA products together to help you make informed decisions, to guide strategic institutional planning and forecasting based on analysis of historical trends, and to enhance institutional performance.

The BPRA solution set includes the following products:

- Banner Operational Data Store (Banner ODS)
- Banner Enterprise Data Warehouse (Banner EDW)
- Advancement Performance (AP)
- Banner Recruiting and Admissions Performance (Banner RAP)
- Banner Student Retention Performance (Banner SRP)

Note

Your institution may license some or all of the BPRA products. If you do license multiple BPRA products, it is important that you understand the relationship among all of the products as you use them.

The following figure illustrates the components of the BPRA suite of products.

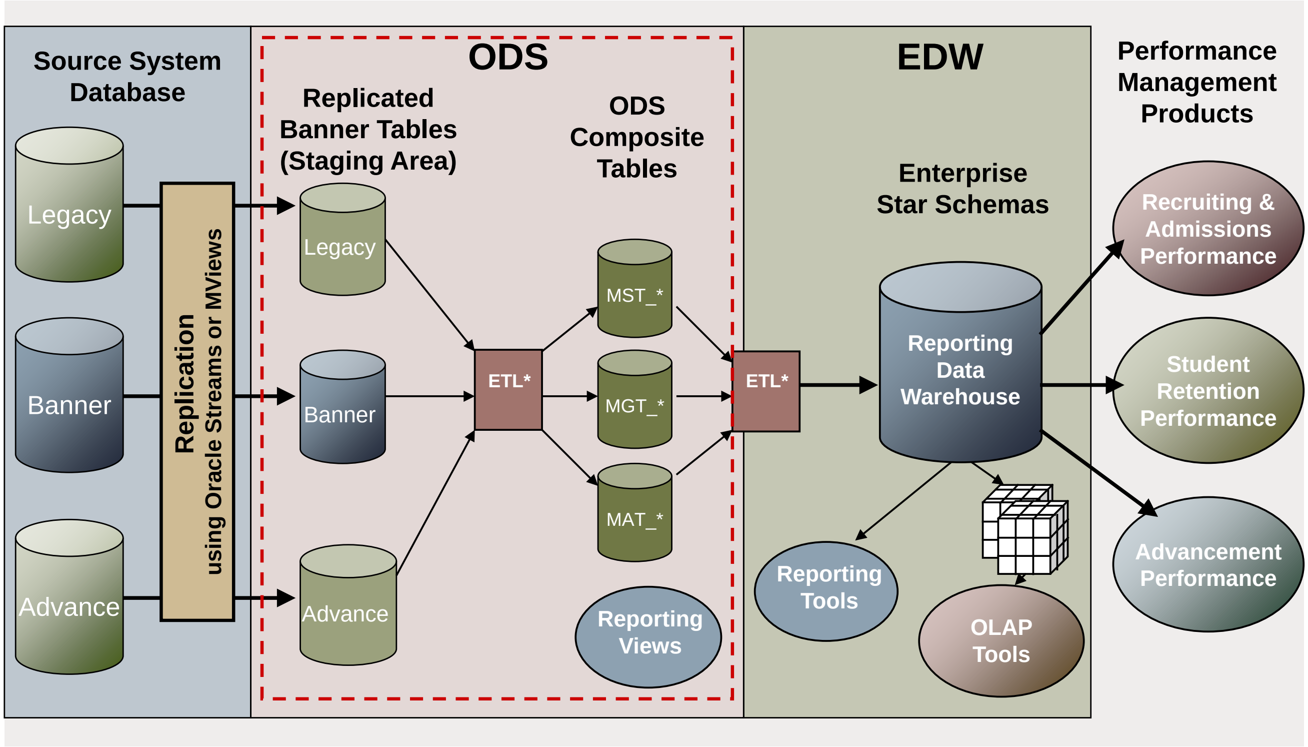


Figure 6: BPRA product architecture

Source system database

The starting point for any performance or reporting analysis solution is your source system data. The information stored in the source transactional database is ultimately the information that you want to analyze.

The BPRA products are specifically designed to accept information from the Banner and Advance products. However, the BPRA products use an open design and can accept information from other sources as well. References to the “source” database refer to whichever source product you use, typically Banner or Advance.

Target database

The “target” database refers to the database where you load information from the source database. Depending on the way you license your BPRA products, this may be the Banner ODS or Banner EDW database or both.

Banner Operational Data Store

The Banner ODS enables you to extract information from your source administrative systems and reorganize the information into a simplified set of tables in the Banner ODS database. End users can then create and deploy operational and ad hoc reports.

Banner ODS provides an extensive and flexible data store and business-organized reporting views with fewer columns and improved performance. You can use these views alone, or in combination with other views. Banner ODS also uses the supported third party reporting metadata layers to deliver an enterprise business area with many prejoined conditions to enhance operational and ad hoc reporting.

In the Banner ODS information from complex and normalized source tables are grouped into more simplified, denormalized tables that are grouped by concept. The following picture illustrates how data from Banner tables of person-related information are combined into one Banner ODS table named AS_PERSON.

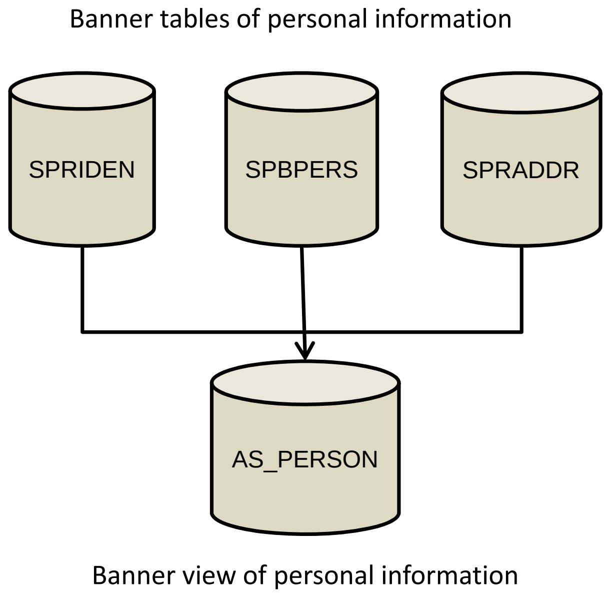

Banner tables of personal information
SPRIDEN
SPBPERS
SPRADDR
AS_PERSON
Banner view of personal information

Figure 7: Banner to Banner ODS table consolidation

In Banner, to properly access the data, you need to understand the rules used to store the data in each table and the rules used to properly join the tables. Using the Banner ODS, you can access replicated Banner data in the ODS without the need to understand the complexities of the data structure because you can retrieve the data from the view.

Banner Enterprise Data Warehouse

The Banner EDW is a multi-dimensional database that gives you a complete picture of your institution’s current and past business conditions. The Banner EDW offers comprehensive reporting and analysis capabilities by providing the following data objects:

- Operational/Aggregate stars that you can refresh with current data on a daily basis at both a summary and detail level
- Snapshot stars that offer a historical snapshots of the data at institution-specific points-in-time at a summary level

This combination of current and historical data allows you to do comparative reporting and analysis. Banner EDW includes prebuilt metadata integration with the IBM Cognos BI software to enable fast deployment of reports and analytics.

Performance Management applications

The Performance Management products are a subset of BPRA products that you can license and use in conjunction with the Banner ODS and/or Banner EDW to monitor and manage your institutions business objectives and analyze outcomes. The following picture illustrates the Performance Management products and high-level features.

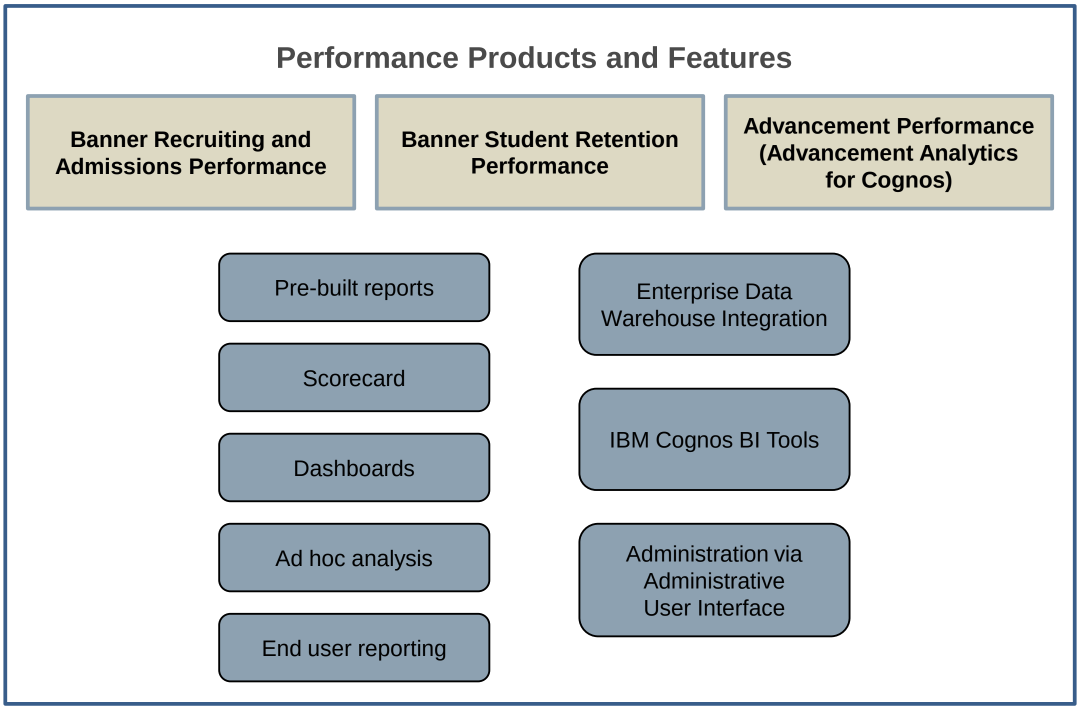

Performance Products and Features
Banner Recruiting and Admissions Performance
Banner Student Retention Performance
Advancement Performance (Advancement Analytics for Cognos)
Pre-built reports
Scorecard
Dashboards
Ad hoc analysis
End user reporting
Enterprise Data Warehouse Integration
IBM Cognos BI Tools
Administration via Administrative User Interface

Figure 8: Performance Management products and features

Each Performance Management product includes the following types of objects built using the IBM Cognos Business Intelligence application:

- Business Concept Packages - reporting metadata layer
- Cubes - predefined reporting structures for quick analysis of summary measures by many attributes

- Reports - display trends of outcomes, summaries of current outcomes, and detailed information about students, applicants, recruits, or constituents (depending on the product)
- Dashboards - display several graphical performance charts for a specific business area on a single screen that you can review at a glance
- Scorecards - display institutional goals and objectives including Key Performance Indicators (KPIs) that monitor progress toward your goals and objectives and a set of strategic initiatives that are needed to produce desired outcomes

The data for these objects is stored in the Banner EDW. These objects are intended to illustrate the kind of analysis you can perform on the warehouse data. You can use the reports, dashboards, and scorecards as delivered or you can modify them to reflect the specific information you need to analyze and monitor your institution's progress.

Advancement Performance

The Advancement Performance solution provides Advancement organizations (Banner Advancement and Advance users) with innovative ways to manage prospects and campaigns, drive fundraising, engage alumni and other constituents, and more. The Advancement Performance solution is comprised of the following products:

- Advancement Analytics for Cognos
- Enterprise Data Warehouse (Advancement data)

The Advancement Analytics for Cognos product provides the performance application content and tools and uses the Banner EDW multi-dimensional database that gives you a complete picture of your institution’s current and past business conditions. This permits your institution to report both current and historical data for summary, trend and detail reporting and analysis

Banner Recruiting and Admissions Performance

Banner Recruiting and Admissions Performance is the reporting analytics and performance portion of the Banner Relationship Management Suite that lets you easily access recruitment, admissions, and selected financial aid information and use it to create reports.

Banner Recruiting and Admissions Performance uses the Banner EDW multi-dimensional database that gives you a complete picture of your institution's current and past business conditions. This permits your institution to report both current and historical data for summary, trend and detail reporting and analysis.

Banner Student Retention Performance

You can use Banner Student Retention Performance to monitor student retention, student success (performance and progress) and student engagement to satisfy institution goals and objectives; extend and modify performance monitoring capabilities; and create operational reports and ad hoc queries that meet the specific needs of your institution.

Banner Student Retention Performance uses the Banner EDW multi-dimensional database that gives you a complete picture of your institution's current and past business conditions. This permits your institution to report both current and historical data for summary, trend and detail reporting and analysis.

Data replication

The replication of data between the source and target databases is key to the usefulness of the warehouse solution and in turn the reports built off the target database. Data replication is referred to as the “staging” process, which simply means to copy tables in the source database into the operational staging area of the target database as illustrated by the following picture.

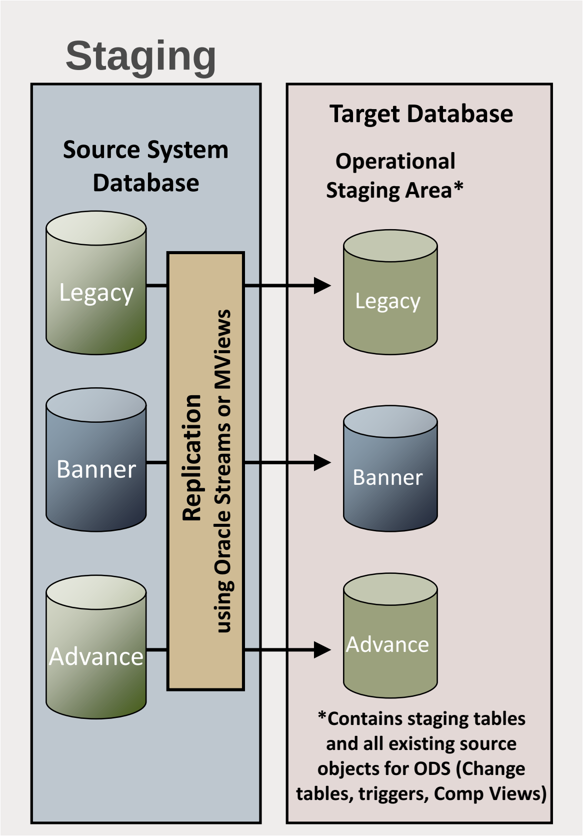

Staging
Source System Database
Legacy
Banner
Advance
Replication
using Oracle Streams or MViews
Target Database
Operational Staging Area*
Legacy
Banner
Advance
*Contains staging tables and all existing source objects for ODS (Change tables, triggers, Comp Views)

Figure 9: Data replication - staging data

You have two options for staging data in the target database. You can use Oracle Streams or Oracle Materialized Views as the framework for staging data. During the initial installation or upgrade process, your institution chose which staging approach to implement.

Refer to one of the following sections to learn more about managing the staging environment in the framework used by your institution.

- "Oracle Streams framework" on page 2-10
- "Oracle Materialized Views framework" on page 2-12

You can also refer to the *Oracle Streams Operations Supplement* or the *Materialized Views Operations Supplement* for additional information about setting up, configuring, and administering one of the frameworks.

Schemas and users

Each schema in the source system needs to have an identical schema in the target database to successfully synchronize data between the two. The following sections list the components owned by the various schemas for each of the BPRA products.

Banner ODS schemas

The following schemas exist in the Banner ODS.

Schema	Owns
ODSSRC	• All composite views • Database packages that contain business logic used in the composite views and trigger logic. **Note:** These are all the ETL objects that used to live in the ODSMGR schema of Banner prior to ODS 8.2.
ODSMGR	• All composite tables, and the reporting views that sit on top of the composite tables • Database packages that contain business logic used in the reporting views • OWB mapping packages
IA_ADMIN	• Metadata tables • Any objects used or associated with the Administrative User Interface like the parameter table, the data display rules table, and the security tables.

Schema	Owns
ODSSTG	Depending on the architecture framework you use, the ODSSTG schema owns one of the following: • Oracle Streams objects *Or* • Materialized Views objects (packages) This user is created in Banner and the ODS.
ODSEUL	• Discoverer End User Layer tables **Note:** ODSEUL is the default name of this schema; however, you can rename the schema.
ODSLOV	• List Of Value views that are created from the MGT_VALIDATION table These views are used as part of the Cognos, Discoverer and Self-Service Reporting (SSR) tool metadata layers to build List of Values that can be used for reporting.
SSRMGR	• All objects related to building the SSR application

If you source the target database from a Banner source database, the following additional schemas may exist depending on which Banner products you license and stage in your target database.

- ALUMNI
- FAISMGR
- FIMSMGR
- FTAEMGR
- GENERAL
- PAYROLL
- POSNCTL
- SATURN
- TAISMGR

These schemas would house the staging tables (materialized views), change tables, and triggers.

ODSSTG password management

You must pay special attention when changing the password to the ODSSTG database user account on either the source or target database because the ODSSTG account in the Banner ODS has an Oracle DB Link back to the source account. If you change any user account passwords for schemas on the source database, for example, in Banner ODSSTG, SATURN, GENERAL, you must also update the DB link in the Banner ODS database to match the password for the related Banner account schema.

Refer to FAQ 1-AXRVD8, which describes the process and steps to alter passwords for any of the Banner ODS related database accounts.

Banner EDW schemas

The following schemas exist in the Banner EDW.

Schema	Owns
EDWMGR	• All fact, dimension, aggregate tables • OWB mappings to load those tables
EDWSTG	• EDW stage/input & clean tables • OWB mappings to load those tables • Table function packages to load input tables

Banner RAP schemas

The RAP product includes the schemas listed for Banner EDW as well as the following additional schema.

Schema	Owns
RELATEMGR	• Staging tables that house the Relationship Management information • Triggers and change tables associated with the staging tables

Banner SRP schemas

The SRP product includes the schemas listed for Banner EDW as well as the following additional schema.

Schema	Owns
RELATEMGR	• Staging tables that house the Relationship Management information • Triggers and change tables associated with the staging tables

Oracle Streams framework

Note

Refer to the *BPRA Oracle Streams Operations Supplement* for more information about maintaining the Oracle Streams framework.

The Oracle Streams framework uses Oracle Streams functionality to replicate data from the source to target database. Any insert, update, or delete actions performed on the source database tables are also performed on the tables in the staging area of the target database to synchronize the source and destination databases. The existing table triggers, change tables, and packages to create change records for the target database reside in that database.

The following picture shows the components used to replicate data between the source database and the target database.

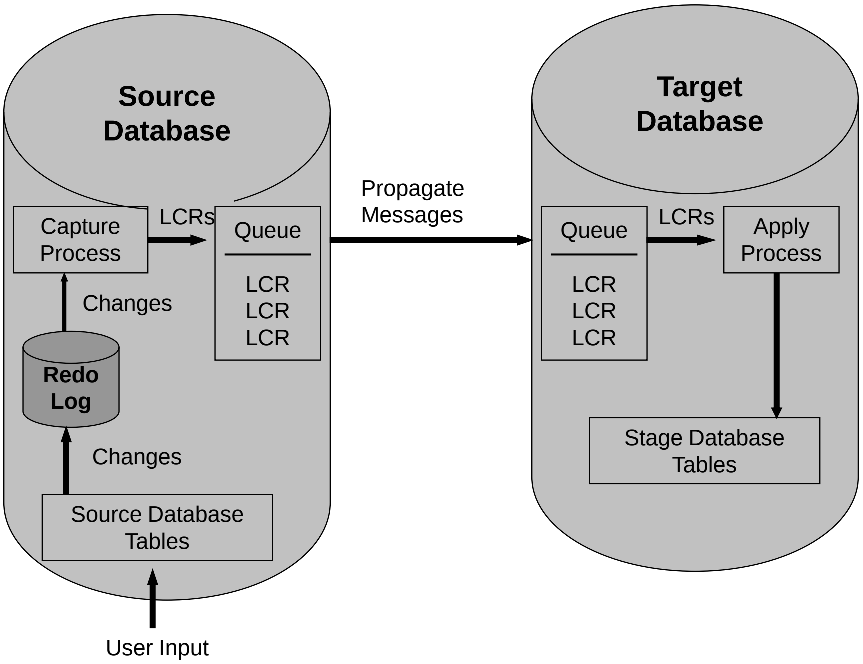


Figure 10: Oracle Streams data replication between source and target databases

The replication process uses the following Oracle Streams components.

Oracle Streams Component	What it does...
Capture Process	The Streams capture process mines changes from the redo log on the Banner database whenever possible and from the archived logs if it falls behind the generated redo logs. Changes in the redo log that match specified rules are converted into messages called logical change records (LCRs), which are placed in a queue associated with the capture process.
Propagation Schedule	The Streams propagation schedule moves the messages from the source database queue to a queue on the Banner ODS. Each message remains in the source queue until the destination confirms that it has received it. This confirmation guarantees messages are never lost during the propagation stage.
Apply Process	The Streams apply process in the Banner ODS removes the messages from the queue and applies them directly to the destination tables in the Banner ODS. Any errors encountered while applying the messages are placed into an error queue. The messages in the error queue can be reprocessed once any issues have been resolved.

During the installation or upgrade process, the system creates all the components required to set up the Oracle Streams framework. This includes setting up the Streams queues, propagation schedule, capture and apply processes, and populating the staging tables in the warehouse with data from the source Banner tables.

Refer to the following Oracle documents for more information about maintaining and troubleshooting the Oracle Streams environment.

- “Oracle Streams Concepts and Administration Guide”
- “Oracle Streams Replication Administrator's Guide”
- “Streams Complete Reference FAQ” (MetaLink Document ID: 752871.1)

DDL Handler

A DDL handler is assigned to each Streams apply handler to replicate DDL statements from the source database to the staged tables in the target database. DDL statements run against a source table are replicated to warehouse unless a statement includes table dependencies. This means DDL statements executed on the source database that create, alter, or drop columns, non-foreign key constraints, and indexes are replicated to the target database; however, using the DDL handler allows the replication process to ignore the same types of statements for table triggers and foreign keys. These changes will not be replicated in the target database.

The user who executes a DDL statement on the source database must also exist in the target database for the change to be replicated successfully in the target.

When a DDL command is executed on the source database and the object schema is not specified, then the DDL command will only be applied successfully on the target database if the source user who executed the command also exists in the target database.

Warning

Be aware that source objects with system-generated names will have different names in the target database. This means that DDL statements involving these objects will not be replicated successfully, and will result in DBA_APPLY_ERROR records being created. The majority of the system-generated names in the source system that may be affected are NOT NULL constraints on table columns. ■

Oracle Materialized Views framework

Note

Refer to the *BPRA Materialized Views Operations Supplement* for more information about maintaining the Oracle Materialized Views framework.■

The Materialized Views architecture uses Oracle materialized views (mviews) to initially stage data in the Banner ODS database and keep that data synchronized with the source database. In the Materialized Views architecture the staging tables in Banner ODS are actually materialized views that have the same names as the tables in the source database and get their data from the source database tables over a DB link. These materialized views are implemented as physical database tables that include triggers.

The Banner ODS upgrade process creates materialized views for all of the tables in your source database that are associated with Banner ODS. The upgrade process also creates an mview log for each table in the source database that doesn't already have an mview log associated with it. These mview logs track changes that are made to the source tables. The processes that refresh the Banner ODS database read the changes recorded in the mviews logs and update the materialized views (staging tables) in the Banner ODS staging area accordingly. This keeps the staging area synchronized with the source database.

The Operational Staging Area refers to those schemas in the Banner ODS/EDW database where the staging tables (replicated copies of the source tables) reside. In the mviews framework, the staging tables are implemented as read-only materialized views. This reduces the chance of a conflict between the source tables and the stage tables (materialized views) because the materialized views cannot be updated. The only way that a materialized view in the Banner ODS would be out of sync with its master table in the source system is if the source table has had a change that hasn't yet been applied to the Banner ODS via the refresh process.

There is an ODSSTG schema in the Banner ODS that houses the staging infrastructure and an ODSSRC schema that houses the ODS ETL components, for example, composite views. Refer to the "Multi-Entity Processing" section later in this document for information about schema relationships and how they relate between the source and target databases.

The following picture illustrates where materialized views fit into the Banner ODS architecture. For every source table that is used to create a materialized view, there is an associated materialized view log where changes are tracked for refresh purposes.

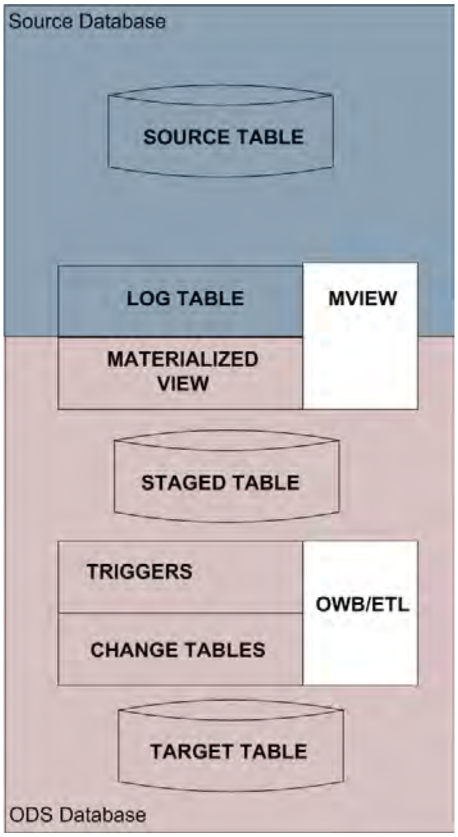


Figure 11: Materialized Views in Banner ODS architecture

Database links

The Materialized Views framework uses database links (db links) to connect to the source system and create or refresh the materialized views in the Banner ODS. To support the use of Oracle Fine Grained Access Control (FGAC) in the source system the Oracle user associated with the DB Links (the ODSSTG user in the source system) must be defined as FGA policy exempt.

There are two options for creating the required database links between the Banner ODS and the source system: a single public database link or multiple private database links. The choice between public and private db links has potential database security implications. Your institution should consider both options, and then decide which one is the best choice for your institution and internal policy requirements.

Public database link

You can create the db link as a public database link. You create this link as the ODSSTG user in the BPRA instance and connect to the source database as ODSSTG. This option simplifies the administration of the Materialized Views environment since it requires only a single link. However, a public db link may pose a security concern because it is “public”.

Private database link

You can create the db links as private database links. This requires creating a private db link for each of the source system schema owners. Each of these database links will be owned by the individual schema owners and will connect to the ODSSTG user in the source system. Using private db links provides somewhat tighter security because the links are private, but it also adds to the potential administrative overhead because of the number of links that you need to maintain. Additionally, if private db links are used there will also be a private db link owned by ODSSTG connecting to the ODSSTG user in the source system.

Each schema requires only one db link from Banner ODS to the source when you choose to use private database links. The following figure illustrates some of the schemas that would be in place for a Banner ODS environment with Banner as the source system.

Note

This illustration does not include all schemas.

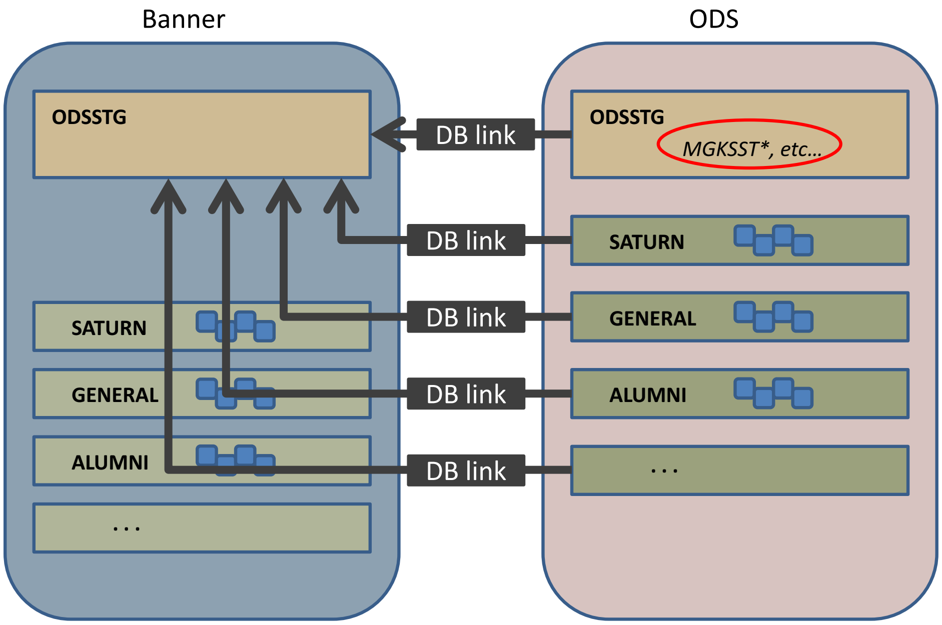


Figure 12: Source and Banner ODS schemas

Source to target data flow

The following picture illustrates the detailed data flow of information from the source to target database using Materialized Views. This example identifies a Banner table and its components related to the materialized view framework. An Advance source system would share a similar data flow replacing the Banner-specific tables and codes with comparable Advance tables and codes.

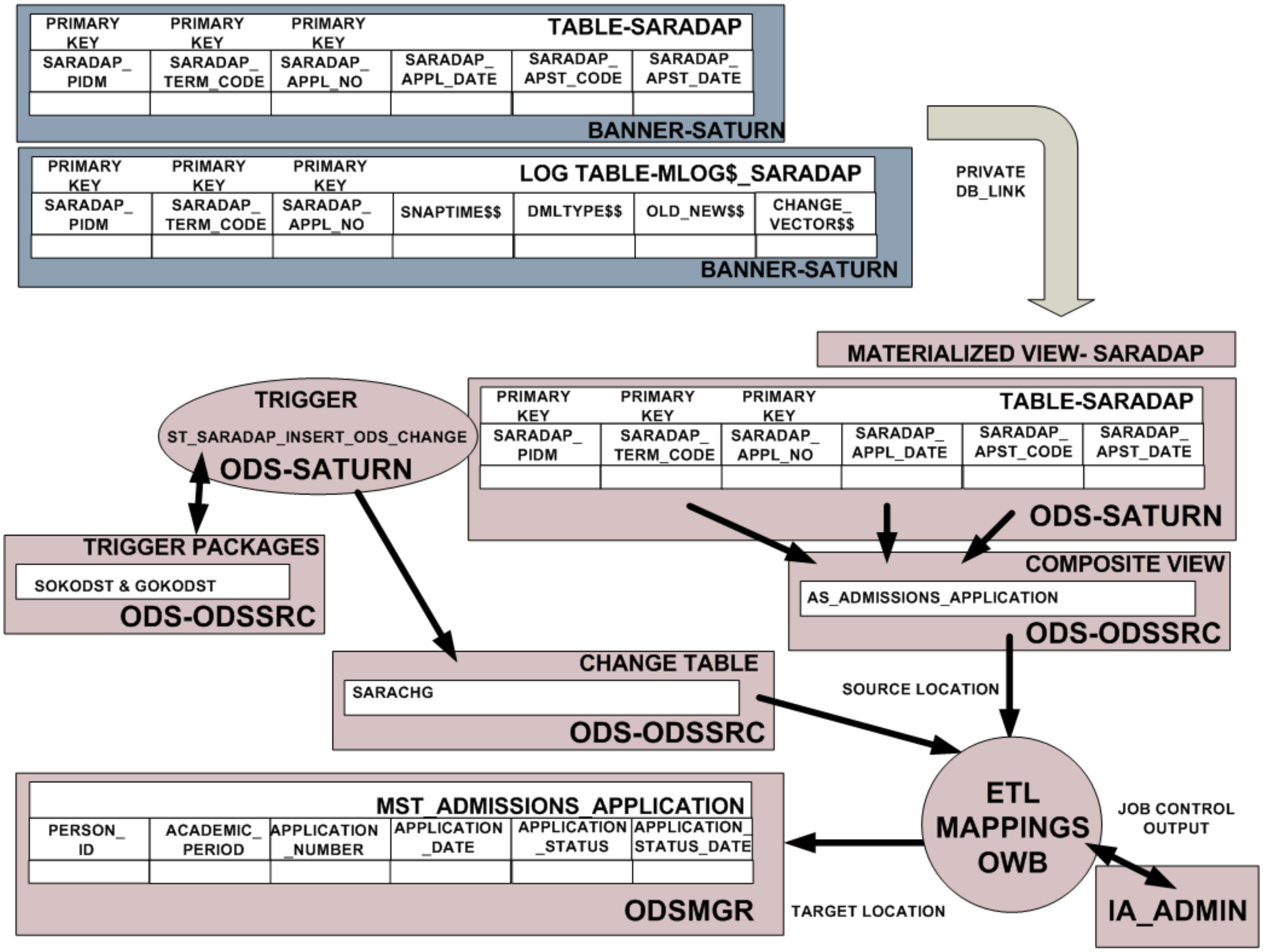


Figure 13: Materialized Views with Banner and ODS data flow

This example uses the Banner table SARADAP, which resides in the SATURN schema in the Banner database. The log table created for this Banner table is called MLOG$_SARADAP and also resides in the SATURN schema in the Banner database. This log is used to track INSERT, UPDATE, and DELETE actions occurring in the SARADAP table to be used by the FAST refresh of the materialized view. In this case, the SARADAP table in Banner has a Primary Key on it which is used for the FAST refresh. The MLOG$_SARADAP table will log records in it which contain the Primary Key columns. These columns will be used during the FAST refresh to compare data in the log to the data in the materialized view.

Records will continue to accumulate in the log tables until a materialized refresh is performed for SARADAP. The materialized view refresh process uses the DBLINK back to the Banner database to pull the records from the log and refresh the materialized view SARADAP in the ODS.

From this point forward, the ETL architecture will perform as it has previously using triggers and change tables to refresh the Banner ODS. The difference being that all of this logic has been taken out of the Banner database and now lives in the ODS database. The materialized view SARADAP will have an ODS trigger on it (ST_SARADAP_INSERT_ODS_CHANGE) which then populates the change table SARACHG. This change table, in conjunction with the AS_ADMISSIONS_APPLICATION Composite View, is used as part of the ODS refresh of the Composite Table MST_ADMISSIONS_APPLICATION.

Staging infrastructure

As part of setting up and maintaining the staging area of the warehouse, you will create, load, and stage materialized views in the target database. In addition, you may need to remove and restage materialized views.

The initial installation of or upgrade to the Materialized Views framework performs the initial creation of materialized views and the database links that are needed to support the delivered Banner ODS features. These materialized views are the staging tables located in the staging area of the target database.

Note

Refer to the "Maintain Materialized Views framework" section in Chapter 4, "Administrative User Interface" for more information about staging and maintaining the Materialized Views architecture. ■

Materialized views staging objects

The system uses several objects (tables, packages, function, and procedures) to manage and let you monitor the Materialized Views framework. The following figure illustrates the staging objects used with the Materialized Views framework.

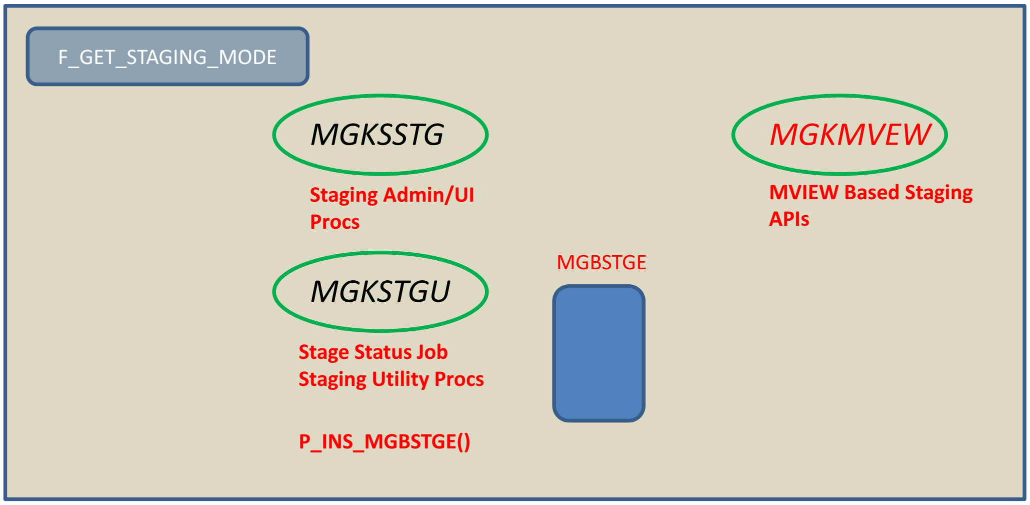


Figure 14: Staging packages for Materialized Views

The F_GET_STAGING_MODE function returns a value showing whether tables have been staged in the Banner ODS. The MGBSTGE table contains a record for each Banner table that is needed to support the delivered Banner ODS processes. The MGKMVEW contains the procedures that are used to stage tables as materialized views. The packages MGKSSTG and MGKSTGU provide the functionality to maintain and monitor the materialized views from the BPRA Administrative User Interface (UI).

The Administrative UI tool offers you a user GUI interface to perform tasks that maintain and update the Materialized Views framework. Using the Administrative UI, you can create additional materialized views or refresh the materialized views. The procedures P_STAGE_MVIEW and P_UNSTAGE_MVIEW located in the MGKMVEW package are the actual components used to carry out the staging (data refresh) tasks.

P_STAGE_MVIEW procedure

The system uses the P_STAGE_MVIEW procedure located in the MGKMVEW package to create, reload, or restage the materialized views that replicate source tables in the Banner ODS. All indexes on the source system master table are created on the materialized view in the Banner ODS. This procedure also creates the necessary synonyms in the ODS to support the ETL processes.

```
PROCEDURE P_STAGE_MVIEW(
            SRC_ALIAS_IN           VARCHAR2,
            {OWNER   VARCHAR2,
            TABLE_IN   VARCHAR2
            |  TABLES_IN              ODSSTG.STAGING_TABTYPE,
}
            OVERWRITE_IN           VARCHAR2 DEFAULT NULL,
            MVLOG_TABLESPACE_IN    VARCHAR2 DEFAULT NULL,
            MVIEW_TABLESPACE_IN    VARCHAR2 DEFAULT NULL,
            MVINDX_TABLESPACE_IN   VARCHAR2 DEFAULT NULL,
            DIRECTORY_IN           VARCHAR2 DEFAULT NULL);
```

P_STAGE_MVIEW procedure arguments

Following are descriptions of the arguments for the P_STAGE_MVIEW procedure.

SRC_ALIAS_IN

A logical name assigned to the database link owned by ODSSTG that points to the master site.

OWNER_IN, TABLE_IN | *TABLES_IN*

The owner and tables to be staged in the ODS as materialized views. OWNER_IN must be a single owner, while TABLE_IN supports wildcards in order to stage a single or multiple tables.

The procedure is overloaded to alternatively accept a TABLES_IN parameter. TABLES_IN is a collection of objects defined as fields of OWNER and TABLE_NAME, where both fields are of a VARCHAR2(30) data type.

OVERWRITE_IN

An optional string specifying what to do when a materialized view is already staged in the ODS. Valid values are:

- 'N' - No overwrite. Do nothing if the materialized view already exists. (Default value)
- 'S' - Synchronize (or reload) the materialized view. This option disables any table triggers on the materialized view, purges the materialized view from the materialized view log, performs a complete refresh of the staged data, and enables the table triggers.
- 'Y' - Overwrite (or restage) the materialized view. This option saves all table triggers for the materialized view, safely drops the materialized view log, drops the materialized view, recreates the materialized view log, recreates the materialized view, and restores all table triggers on the materialized view.

MVLOG_TABLESPACE_IN

An optional string specifying in which tablespace at the master site the materialized view log should be created. If no value is specified, the log will be created in the default tablespace for the owner of the master table.

MVIEW_TABLESPACE_IN

An optional string specifying in which tablespace at the materialized view site the materialized view should be created. If no value is specified, the materialized view will be created in the default tablespace for the owner of the materialized view.

MVINDX_TABLESPACE_IN

An optional string specifying in which tablespace at the materialized view site the materialized view indexes should be created. If no value is specified, the indexes will be created in the default tablespace for the owner of the materialized view.

DIRECTORY_IN

This procedure provides the ability to either create the materialized views for you, or generates scripts that can be run later. This is an optional parameter specifying where to generate the scripts. This must be a valid directory object name as can be viewed in the ALL_DIRECTORIES database view.

Each call to the P_STAGE_MVIEW procedure also generates the following driver scripts needed to create and drop the materialized views:

- mvlogs_create.sql - run at the master site to create the materialized view logs
- mviews_create.sql - run at the materialized view site to create the materialized views and synonyms
- mvlogs_drop.sql - run at the master site to drop the materialized view logs
- mviews_drop.sql - run at the materialized view site to drop the materialized views and synonyms

Example: Stage ALUMNI table

The following command stages the ALUMNI.AABDUES table in the ODS. The materialized view log will be created in the ALUMNI user's default tablespace at the master site. The materialized view and indexes will be created in the ALUMNI user's default tablespace at the materialized view site.

```
exec mgkmvew.p_stage_mview('BPRA_BANNER', 'ALUMNI', 'AABDUES');
```

Example: Restage tables

The following commands will restage both the SATURN.SPRADDR and GENERAL.GURMAIL tables. All materialized view logs will be created in the MVLOG tablespace at the master site. All materialized views and indexes will be created in the MVIEW and INDX tablespaces at the materialized view site, respectively.

```
Declare
      Tab   odsstg.staging_tabtype;
Begin
      Tab(1) := 'SATURN.SPRADDR';
      Tab(2) := 'GENERAL.GURMAIL';
      Mgkmvew.p_stage_mview(
      Src_alias_in => 'BPRA_BANNER',
      Tables_in => TAB,
      Overwrite_in =>  'Y',
      Mvlog_tablespace_in => 'MVLOG',
      Mview_tablespace_in => 'MVIEW',
      Mvindx_tablespace_in => 'INDX');
End;
/
```

Example: Generate materialized view scripts

The following commands will generate materialized view scripts for all Banner Finance materialized views. The scripts will be generated in the database server's directory associated with the DATA_PUMP_DIR directory object. The script will create all objects in their users' default tablespaces.

```
Begin
      mgkmvew.p_stage_mview(
      src_alias_in => 'BPRA_BANNER',
      owner_in => 'FIMSMGR',
      table_in => '%',
      directory_in => 'DATA_PUMP_DIR');
End;
/
```

P_UNSTAGE_MVIEW procedure

The P_UNSTAGE_MVIEW procedure located in the MGKMVEW package safely drops the materialized view log from the master site (source), and drops the materialized view and synonyms from the Banner ODS.

```
PROCEDURE P_UNSTAGE_MVIEW(
         SRC_ALIAS_IN           VARCHAR2,
         {OWNER   VARCHAR2,
         TABLE_IN   VARCHAR2
         |  TABLES_IN              ODSSTG.STAGING_TABTYPE,  }
         PRESERVE_MV_TABS_IN    VARCHAR2 DEFAULT 'N');
```

P_UNSTAGE_MVIEW procedure arguments

Following are descriptions of the arguments for the P_UNSTAGE_MVIEW procedure.

SRC_ALIAS_IN

A logical name assigned to the database link owned by ODSSTG that points to the master site.

OWNER_IN, TABLE_IN | TABLES_IN

The owner and tables to be removed from the ODS. OWNER_IN must be a single owner, while TABLE_IN supports wildcards in order to remove a single or multiple tables.

The P_UNSTAGE_MVIEW procedure is overloaded to alternatively accept a TABLES_IN parameter. TABLES_IN is a collection of objects defined as fields of OWNER and TABLE_NAME, where both fields are of a VARCHAR2(30) data type.

PRESERVE_MV_TABS_IN

An optional string when 'Y' specifies the materialized view should be dropped, but the underlying table structure and data should remain in the ODS. Once complete, this data can no longer be refreshed based on records in the materialized view log. When the value is 'N' then the underlying table and data will be dropped along with the materialized view. The default value is 'N'.

Example: Remove materialized views from staging area

The following example will remove all of the Accounts Receivable materialized views from the Banner ODS staging area.

```
exec mgkmvew.p_unstage_mview('BPRA_BANNER', 'TAISMGR', '%');
```

Example: Remove table and materialized views from staging area, keep sources

The following will remove the ALUMNI.AABDUES table and the POSNCTL.NHRDIST materialized views from the staging area, but the underlying tables and data will be retained.

```
Declare
      Tabodsstg.staging_tabtype;
Begin
      Tab(1) := 'ALUMNI.AABDUES';
      Tab(2) := 'POSNCTL.NHRDIST';
      Mgkmvew.p_unstage_mview(
         Src_alias_in => 'BPRA_BANNER',
         Tables_in => tab,
         Preserve_mv_tabs_in => 'Y');
End;
/
```

makeMVs.sql script

The makeMVs.sql script is available in the `ia_admin\dbscripts\utility_scripts` directory. This is a sample script that shows how to mass-generate Materialized Views scripts for two schemas.

Refresh Materialized Views

The Materialized Views architecture uses Oracle's Fast Refresh process, which tracks only the changes since the last refresh. This incremental refresh functionality speeds the process of refreshing the materialized views in Banner ODS. If there is a primary key on the source table, the Fast Refresh uses that key to perform the refresh. When a change happens in the source, the change is put in the log and the key is logged. If a source table doesn't have a primary key, the materialized view is refreshed using RowID.

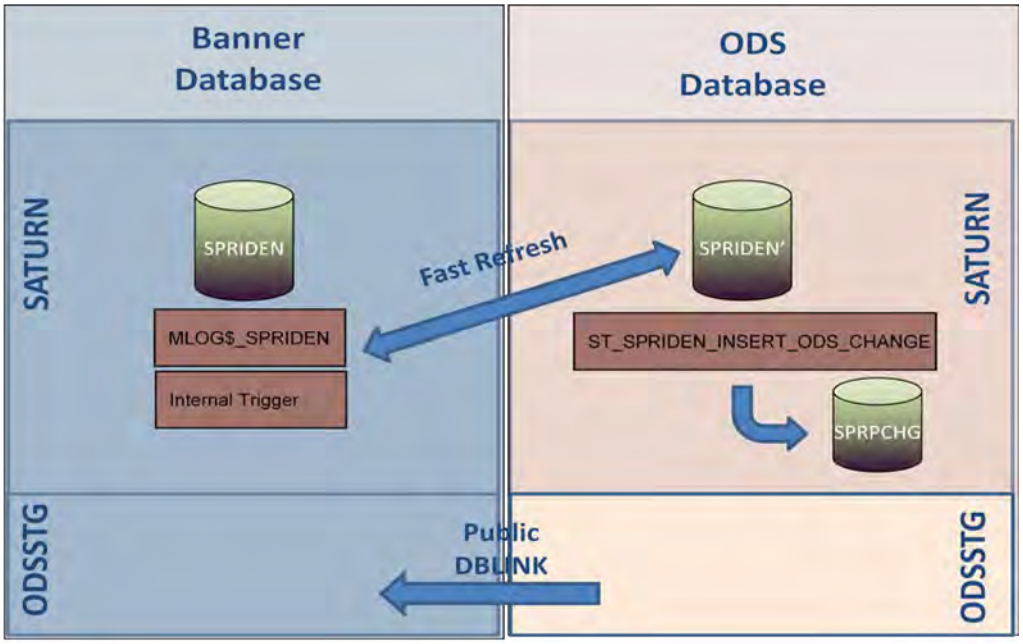
Because the materialized view architecture uses the Fast Refresh, a materialized view log is created for each table in the source system during the upgrade to the Materialized View architecture. The log specifies how to track changes. As part of the log creation, Oracle creates a trigger on the table. In the warehouse, when materialized views are created (with names same as in source database) the system gets all data and table structure from the source over the DB Link and includes the statement “refresh fast on demand with primary key”.

Materialized view refresh processing has been added to the beginning of the ETL jobs available in the Administrative User Interface (UI).

Note

Refer to Chapter 4, “Administrative User Interface” for details about using the Administrative UI to refresh the materialized views and maintaining the materialized views framework. ■

The following picture illustrates the objects used and actions performed during refresh of the SPRIDEN materialized views.



Changes to the SPRIDEN table in Banner fire the internal Oracle trigger on SPRIDEN and insert changes into the mview log (MLOG$_SPRIDEN). Changes continue to accumulate in the mview log until a materialized view refresh is performed. After a materialized view refresh occurs, the records from the mview log are pushed over to the SPRIDEN materialized view in Banner ODS to synchronize it with the SPRIDEN table in Banner. As changes are pushed into the materialized view, the ODS trigger (`ST_SPRIDEN_INSERT_ODS_CHANGE`) on the SPRIDEN materialized view fires and inserts a record into the ODS Change Table (SPRPCHG). These records are used to refresh the normal ODS Composite Table.

Extract, Transform, and Load process (ETL)

The ETL process uses OWB, triggers, and change tables to load and refresh data from the staging tables to the composite tables in the target database. The following figure illustrates these components:

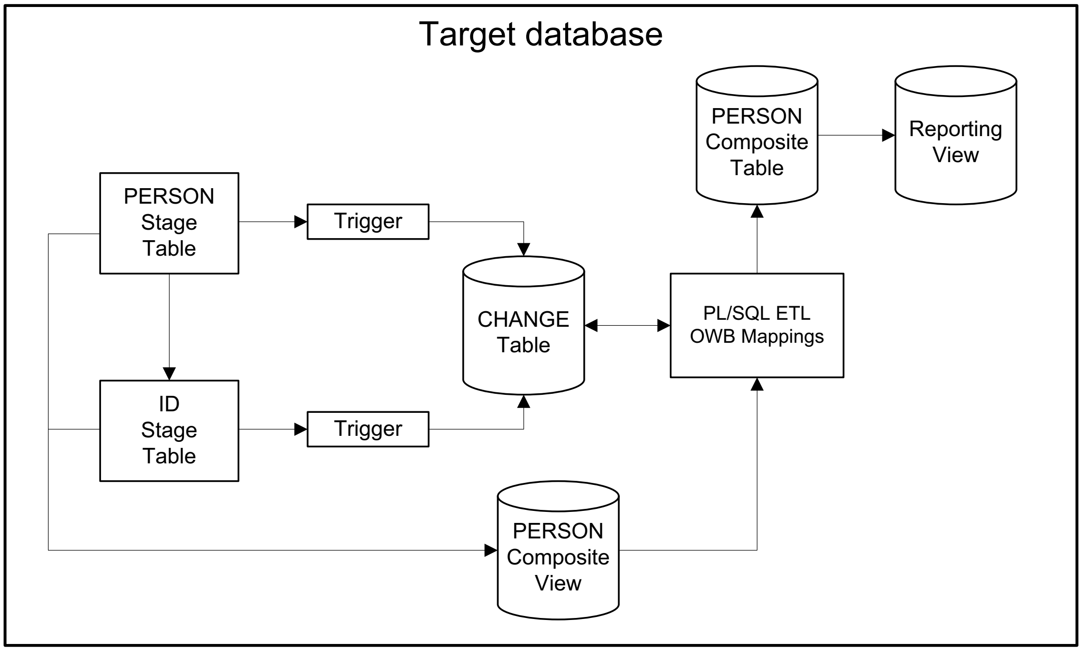


Figure 15: Target database components

The system uses the ETL processes to extract data from the staging tables and load it into the warehouse composite tables. All ETL activities are performed from within Oracle PL/SQL packages and deployed into a target database schema. The PL/SQL packages are created using Oracle Warehouse Builder (OWB). These packages are scheduled and run via the DBMS_JOBS queue in Oracle.

All objects are created in the target database including all change tables, triggers, packages and composite views. All source tables needed to create the composite views are replicated in the target database with the same schema name as in the source. For example, the target database has a SATURN schema which contains replicated source (Banner) tables.

Note

> If you use Materialized Views to replicate data, you can schedule the Materialized Views refresh jobs to keep the source database tables and the target staging tables synchronized.

You can submit and monitor the ETL jobs using the Administrative User Interface. Typically referred to as ‘mappings’, the packages, when executed, delete, update and load data from the staging to the composite tables based on the type of mappings executed.

During the initial load of the target database, data is extracted from the source database using Oracle views that include specific business logic (for example, Enrolled or In State Resident indicators). The extracted data is then migrated into denormalized composite tables within the target database. These composite tables represent a conceptual organizational structure (for example, Student, an Employee, or a Receivable Customer). To provide for data value security, the Administrative UI allows you to create Oracle Fine Grained Access rules and apply them to the composite tables to prevent information from being viewed without authorization.

The final layer of data access is the reporting views. These views allow calculated columns and increased flexibility in managing what data the end users can access. In select instances, such as the slotted concepts, data display rules are applied to user and institution profiles which filter out unwanted data.

To ensure that the data is current, you can incrementally refresh the target database on a scheduled basis. OWB packages combine the business logic views with the change tables located in the product schemas to determine what updates are applied to the target database composite tables.

You can manage all data loads and updates, fine grained access rules, meta data management, data display rules, and freeze data processing using the Administrative UI.

ETL components

The following section describe the various components used to accomplish the ETL process.

Stage tables

Information from the source database tables is replicated in the target database stage tables.

Database triggers

A single database trigger exists on each stage table, except for the validation tables. Triggers exist for all tables used in a view, including function tables. The triggers are created in the schema owner of the associated stage table.

Each trigger identifies Data Manipulation Language (DML) activity on the table. When a change is made to a source table, that change is replicated in the associated target database table. The change in the staging table causes the trigger on that table to fire. The trigger calls a stored PL/SQL procedure which inserts records into the appropriate change tables to reflect the change in the replicated table. The triggers flag changes on Banner replicated tables and create records in the appropriate change tables.

Triggers are created on the actual source replicated tables that provide source data for the target database. The triggers are not delivered with the baseline Banner applications.

Trigger packages

Trigger packages manage the trigger procedures. There is one procedure for each change table with each procedure managing a unique index on the change table. There is one package per product area within the target database, such as Student, Human Resources, Finance, Financial Aid, Advancement. ODSSRC owns the trigger packages.

As data is entered into the source database, it is typically processed one row at a time. For each field entered, the data is verified for field syntax, such as date or numeric format. Fields requiring additional verification are verified against rule tables. After the values are properly checked, the data is committed to the database table that will house the information. During the commit action, any Oracle triggers on the database table being updated are initiated and additional, but separate, logic is executed based on the parameters of the trigger (such as Before Insert and After Insert).

Triggers are built and enabled on all source database replicated tables that house information that is used in the target database. Therefore, when a target database trigger is fired, the trigger inserts the keys of the data being changed into the change tables along with a DML indicator. The existence of these rows in the change tables tells target database that the source has data waiting to be retrieved.

Note

The change tables only maintain the most recent database activity for a row of information for a specific key. When multiple actions occur against the same source database table and row, only the last action is represented in the change table. This allows the replication process to work faster, and decreases the amount of data captured in the change tables. ■

Change tables

Change tables maintain data about what tables and records have been changed, inserted, or deleted in the stage tables and the source database tables. There is not a one-to-one relationship between change tables and stage tables or between changes tables and composite tables. One change table exists for each logical group of information.

Change tables work like collector tables. They include four basic fields:

- Keys
- Table name
- Process ID

- Most recent DML.

Change tables reflect DML activity for specific target database stage tables, but are also used when multiple tables use the same key.

Example:

The SPRPCHG table stores DML activities for the Hold and the Person composite views.

Change tables are owned by their respective product schemas in the target database, and are identified using standard source table naming conventions. The column names start with the seven-character prefix of the table name. All columns in each of the change tables are identical with the exception of the key columns. Here, the key columns represent the product/database tables they are accessing, and also represent the keys that the target database uses when records change. All change tables are suffixed by 'CHG'.

The columns that compose the change table are the key columns relative to the composite view(s) it supports, along with the `TABLE_NAME` and the `PROCESS_ID` columns. The last two columns allow inserts into the table with a null `PROCESS_ID` by updates to Banner that take place during Incremental Refresh. Since the target database processes and deletes all rows in the change tables with a `NOT NULL PROCESS_ID`, the null value allows the row to stay until the next update. This ensures that it is not bypassed or inadvertently deleted.

Typically, a second index is created in the format of `TABLE_NAME`, `PROCESS_ID`, and `RECORD_ACTION` columns.

Example:

SPRPCHG – Change table for PIDM related Banner replicated tables

Column Name	Data Type	Column Comment
SPRPCHG_TABLE_NAME	VARCHAR2(30)	Used to identify which composite view (and/or target database table) is being populated by this specific row of data.
SPRPCHG_PIDM	NUMBER	The change table needs to hold as many keys as required to manage DELETE and UPDATE information in the target database. Keys do not need to identify a unique row, but must maintain some fields for comparison.
SPRPCHG_RECORD_ACTION	VARCHAR2(1)	Stores the last DML action for the key combination (*I*, *U*, or *D*).

Column Name	Data Type	Column Comment
SPRPCHG_PROCESS_ID	VARCHAR2(30)	Updated by the procedure UPDATE_CHANGE_TABLE which inserts non-null values to flag which rows are being processed during the incremental refresh process. This allows inserts to take place into the change table while replication is also taking place.
SPRPCHG_ACTIVITY_DATE	DATE	Reflects the actual date of insertion or update of the rows.

Change table triggers

The target database maintains triggers on all source replicated tables used to incrementally refresh data into the target. Although the triggers are enabled on the actual source replicated tables, they are referred to as 'change table triggers' because they populate the target database change tables with DML information. The trigger inserts rows of information in one or more change tables by invoking a procedure that packages all trigger insert actions for the target database change tables.

The triggers use basic logic except that the Exception routines allow for continued processing when encountering a DUP_VAL_ON_INDEX condition. This condition occurs when a row of data exists within the change table for the table's unique index. When encountered, the procedure updates rather than inserts the information in the change table by overlaying the DML activity and the activity date. This action causes only the most recent DML activity to be stored in the change table.

All triggers are owned and maintained within the product schema of the table to which the triggers are added. For example, SATURN would own Student Triggers.

Change table Triggers comprise of the following procedures:

- Each Banner product has a procedure that manages all change table triggers for that product area. For example, GOKODST for General and SOKODST for SATURN.
- The triggers are owned by the ODSSRC schema.
- The names for each procedure follow Banner standard naming conventions.

Composite views and functions or packages

Composite views exist in the target database under the ODSSRC schema. During the ETL process, when you perform a refresh of target data, the composite views are joined with the appropriate change tables and updated with the changed information.

In some cases, functions are used to calculate new data that is created from source data and loaded into the target database. Packages are used to group related procedures, functions, and cursors together. There is one package for each target database module of information; for example, Student, Finance, and Advancement. These packages are installed into the ODSSRC schema.

Composite views represent a composite (mixture) of the tables selected from Banner and allow for a single piece of data to be extracted row-by-row, with all the business logic included in the view itself. The column names are generic so that they can be used by all Ellucian product lines. Therefore, names familiar to Banner clients can appear to be more generic than the familiar Banner terminology. For example, Term becomes Academic Period, PIDM becomes UID (unique ID). The views are used for reporting in Banner. But, they are designed to become the Incremental Refresh data extraction view.

Views are created and maintained in the ODSSRC schema within the target database. Since these views are accessing data directly in the various source replicated tables, explicit SELECT grants are assigned to the schema when tables are staged in the target. Refer to the section “Multi-Entity Processing” on page 2-54 to see a list of schemas and what they own.

OWB mappings

Oracle Warehouse Builder (OWB) mappings, which are PL/SQL scripts, define the relationship of data between the composite views and composite tables. The Extract, Transform, and Load processes (ETL) built using OWB are the mappings that populate Banner ODS.

The OWB mappings are run during the initial load of Banner ODS and when you incrementally refresh Banner ODS. When run, the scripts load, update, or delete data in Banner ODS composite tables. Three scripts — Load, Update, and Delete — exist for each Banner ODS composite table. The different types of mappings perform the following functions:

- LOAD mappings: initially load Banner ODS composite tables by selecting all rows of data from the source system via the composite view.
- DELETE mappings: delete rows of data in Banner ODS when the change table reflects activity of any type for the key. This mapping uses the key in the change table since no data will be found in the composite view for deletes. This process also updates the `PROCESS_ID` value in the corresponding change table for all rows before any delete takes place.
- UPDATE mappings: insert records into Banner ODS based on keys in the composite view joined against rows in the corresponding change table.

Note

It is mandatory that you run the DELETE mapping before the related UPDATE mapping, otherwise no records will process in the UPDATE mapping.

The OWB user interface contains graphical editors that enable you to design a complete logical model of your warehouse. The OWB helps you plan how to extract data from a variety of sources, transform the data, and configure the data for loading into Banner ODS. The OWB code generator lets you deploy and populate the Banner ODS without manual coding, and integrates with the Oracle database and query tools.

Composite and slotted tables

Composite tables are the tables within Banner ODS that are loaded with data from the source system. Slotted tables store data values for a specific code related to a base table.

Composite tables

The composite tables are populated during the initial install process, and are also updated during the incremental refresh process. The composite tables are used for the following purposes:

- Denormalized tables are used to store “conceptual” structures of data.
- Normalized tables are used for quick data filtering or for unlimited repeating values.
- The MGRSDAX rule table is used to load the composite tables.

Slotted tables

The slotted tables have the following attributes:

- Used to denormalize Repeating Concepts (normalized tables.)
- Populated via rules from MGRSDAX.

Some GTVSDAX rules, but not values, are duplicated when MGRSDAX is initially populated. Use the Administrative UI to add or modify MGRSDAX rules' values to meet your institution’s needs.

Understanding composite tables and slotted tables

Banner ODS includes composite tables and slotted tables. Composite tables include the main data that is extracted from your source system and stored in Banner ODS. Slotted tables store data values for a specific code related to a base table.

Example

The `TEST_SCORES_SLOTTED` table in Banner ODS stores all valid Test Score values that were loaded from your source system to Banner ODS. When a report is created against Banner ODS, the system pulls data from the composite tables. The system checks codes stored on the slotted tables, as needed, and pulls the appropriate code values. If you choose to use Business Profiles, the system pulls the appropriate values for the profile with which the user is associated. The default business profile of INSTITUTION is used when specific display rules are not established.

Using slotted tables optimizes the speed queries since the system only has to check for specific code values as needed.

Updating slotted tables

It is important to keep data in the slotted tables synchronized with data in the composite tables. Whenever composite tables are updated, related slotted tables should also be updated.

Both composite and slotted tables are updated when refresh jobs are run to update Banner ODS data on a regular basis.

Reporting views

Data from each Banner ODS composite table is presented in one or more reporting views. Banner ODS reporting views are the views that your users use to create reports within Banner ODS. Users point their report writing tool at these views and build reports.

Run ETL load processes

You run load processes using the Administrative User Interface (UI) from the **Options>Schedule a Process>Select a Subprocess>Schedule Banner ODS Mappings** menu option. When you run a process, one or more LOAD mappings extract all the data from a composite view in the source system and move it into the corresponding target database composite table.

You can run a Load process periodically for one or more composite tables, for example, as an alternative to the Refresh process. To facilitate the use of a load at any time, the Load processes also purge the appropriate change tables that correspond to the composite tables being loaded.

You can disable the purge feature on Load mappings. To disable the change table purge for a Load mapping, you need to create records in the MTVPARM table. Refer to the “ETL MAP PACKAGE LOAD PURGE Parameter” section of the “Administrative User Interface” chapter for information about using this parameter to disable a change table purge for a Load mapping.

Run ETL Load or Refresh jobs in parallel

You can schedule the ETL Load and Refresh jobs to run in parallel to reduce the time it takes to load or refresh the entire database.

Run ETL Refresh jobs in parallel in Materialized Views framework

If the ETL Refresh jobs run at the same time, it's possible that materialized views shared by multiple areas of the warehouse may not get refreshed appropriately before the actual ETL Refresh job runs. To avoid this issue, you should refresh any related materialized views before you run the ETL Refresh jobs in parallel.

Example

Suppose that your institution runs the ETL refresh jobs Refresh General, Refresh Student, and Refresh Accounts Receivable at the same time each night. The source Banner SPRIDEN table is used for refreshing both the Student and General subject areas. This means it's possible that the ETL Refresh jobs could load the warehouse before all of the related materialized views have been refreshed. To ensure that all related materialized views are refreshed in the staging area before the ETL Refresh jobs actually load the target database, in this case you should schedule the following mviews refresh jobs to run before the ETL Refresh jobs:

- AR Refresh Group and AR Validation Refresh Group
- General Refresh Group and General Validation Refresh Group
- Student Refresh Group and Student Validation Refresh Group

Incremental refresh process

The term incremental refresh identifies how data synchronization occurs between source and target tables to ensure that accurate information is stored in the target database. Data that has changed in the source is captured and, using the ETL tool, is applied to the target database. During the process, the change tables bring over only the data that has changed, and then, using an ETL mapping, the change tables are deleted. This is followed by an update ETL mapping that inserts the new data. The incremental refresh process uses records in change tables to identify the records which need to be refreshed, and uses different mappings for load vs. refresh processes.

Typically, you will run a complete load, then run the refresh processes on a nightly basis to keep the target data synchronized with the source data. You should also run a incremental refresh process if data in Banner ODS has changed since the last time you ran the refresh.

Banner ODS to Banner EDW data flow

Banner EDW stores data that is fed from the Banner ODS. The data is stored on both primary storage and alternative storage. The data is cleansed and restructured to support queries, summaries, and analyses. The following picture illustrated the flow of data from Banner ODS to Banner EDW.

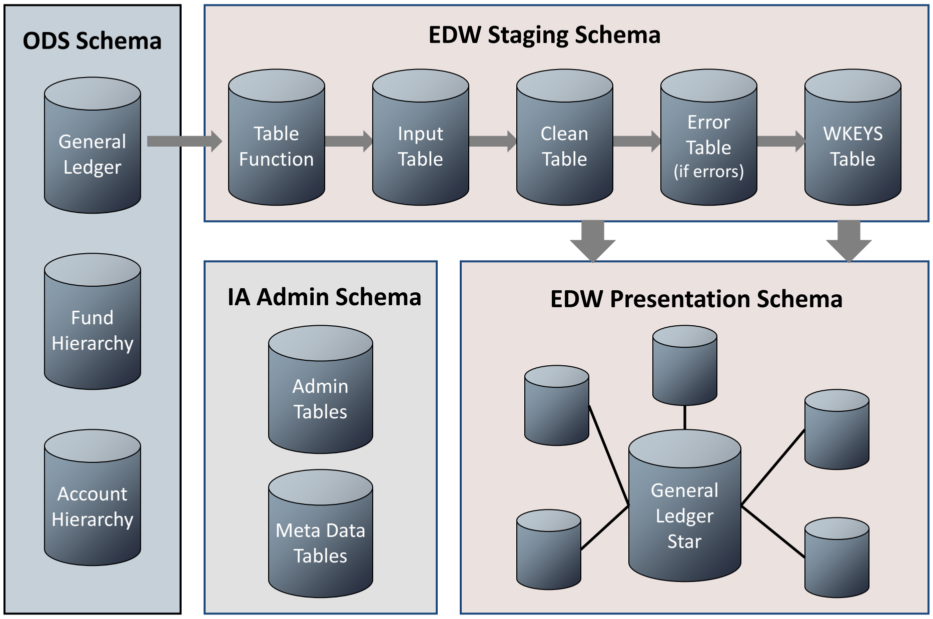


Figure 16: Banner ODS to Banner EDW data flow

Banner ODS is a relational data model that you can continuously and incrementally refreshed using the Banner Administrative UI. Banner EDW reorganizes, groups, and summarizes the information from the Banner ODS star schemas. This information can contain operational data or an event at a specific point-in-time. The Administrative UI schedules Banner EDW load mappings to take place at scheduled times. Banner EDW also includes operational star data models that you can refresh on a regular basis like the Banner ODS.

Banner EDW resides on the same machine and database as the Banner ODS, but resides under the schema owner name of EDWMGR. Banner EDW also uses the EDWSTG schema as a repository for staging tables used to process extracts. The Banner ODS is contained within the ODSMGR schema.

Banner ODS is a database of denormalized tables called composite tables. These composite tables store data contents from the administrative systems and are constructed specifically for reporting. Denormalizing combines data from many smaller tables into fewer, larger tables. This enhances data extraction and query access by eliminating the need to perform intensive performance table joins.

Data is retrieved from the source system(s) using composite views. These views use the existing business logic on the source system, and provide the extraction logic for the composite tables that reside on the reporting server. Banner ODS typically resides on a separate server to take advantage of the performance benefits associated with a query-only system. Business logic is not resident on the reporting server, ensuring that the Banner

ODS model can support all products. Because the Banner ODS is a query-only system, the data in the Banner ODS flows only one way — from the source administrative system to the Banner ODS, never from Banner ODS to the administrative system.

Banner ODS standard composite tables were created with your industry-wide business needs in mind. This enables you to create your own reporting views and reports based on the delivered tables.

Banner ODS also includes Reporting (presentation) views. Reporting views are the final views you use to create reports.

Banner EDW and its associated event-based processing is designed to capture point-in-time information for trend analysis and historical reporting. Banner EDW also includes operational and aggregate star data models that you can refresh on a daily basis in concert with the Banner ODS. These data models provide the ability to daily assess key institution measures and to drill down from these institution measures to the underlying details.

Banner EDW is designed to work with the Banner ODS as a source within the same environment. All data extraction, transformation, and load (ETL) activities are performed by Oracle PL/SQL packages generated by Oracle Warehouse Builder (OWB) and deployed into stage and production warehouse schemas. As with the Banner ODS, these packages are scheduled and run via the `DBMS_JOBS` queue in Oracle. You can submit and monitor the jobs within the Administrative UI.

Banner EDW source and target database instances reside in the same database, but are in different schemas. As Banner EDW loads, data is extracted from the Banner ODS using pipelined table functions. The extracted data is loaded into a staging area where data is cleansed based upon the institutional preferences.

The unique dimensional attribute combinations are then inserted into the dimension tables with a uniquely defined surrogate key. The facts of the extracted data are then loaded into the fact table(s) along with the surrogate keys defining each record's unique combination of dimensional attributes.

To provide for data value security, the Administrative UI enables Oracle's fine grained access rules to be created and applied to the dimension tables and fact tables.

Oracle Warehouse Builder (OWB) to perform ETL

The Oracle Warehouse Builder (OWB) user interface contains graphical editors that enable you to design a complete logical model of your warehouse. The OWB helps you plan how to extract data from the Banner ODS, and transform and configure the data to load into Banner EDW. The OWB code generator lets you deploy and populate Banner EDW without manual coding, and integrates with the Oracle database and query tools.

Administrative User Interface (UI)

The Administrative User Interface is a Web-based interface that uses the Banner Web Tailor. This Administrative UI is used for the following administrative functions:

- Preferences and Security - Use to manage security, set global preferences, and set up user accounts.
- Options - Use to control the processes to extract, cleanse, and load data into the system, schedule a process (execute and monitor ETL processes).
- View control reports, view and remove scheduled processes, and maintain freeze data.
- Meta Data - Use to view and manage the meta data supporting the systems.
- New Banner Web Tailor Administration - Use to customize a Web menu, procedure, graphic element, set of information text, or set of menu items. You can also update user roles, customize a Web module, Web rules, or Banner Web Tailor parameters; customize a login return location; and customize Banner Web Tailor overrides or global user interface settings.
- Cleansing - Use to maintain descriptions to be stored in Banner EDW, and to translate codes from the Banner ODS to Banner EDW.

EDW data loads and Incremental Refresh process

The term “incremental refresh” identifies how data synchronization occurs between the source and target set of tables to ensure that accurate information is stored in Banner EDW. Information that has changed in the Banner ODS is captured and, through the use of ETL tool sets, is applied to the target system, Banner EDW. During the process, the change tables bring over only the data that has changed, and then the change tables are purged.

The Banner EDW incremental refresh is similar to the Banner ODS incremental refresh in that it uses records in change tables to identify the records which need to be refreshed. However, unlike the Banner ODS which uses different mappings for load vs. refresh processing, the Banner EDW refresh uses the same mappings/process as the Banner EDW load but uses the corresponding change table to identify and process only the records which should be refreshed.

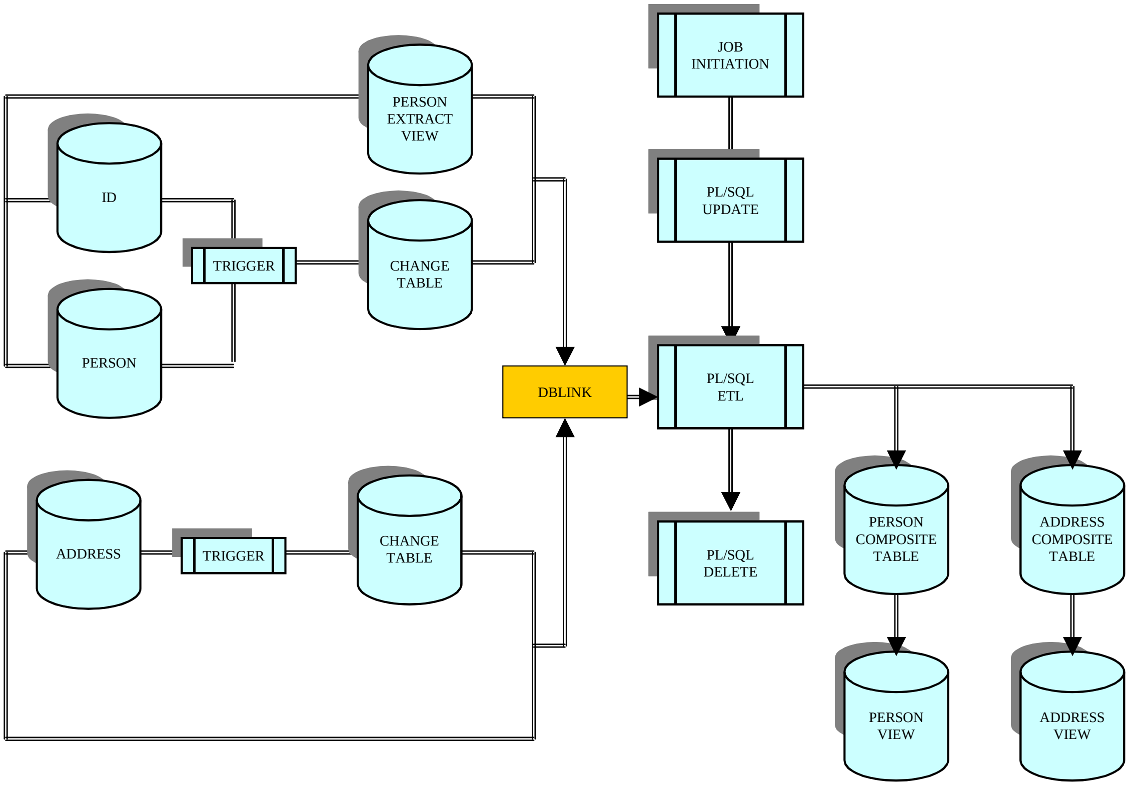
Once the Incremental Refresh process in the Banner ODS is processed, and before the Banner ODS change table records are purged, change records are inserted into Banner EDW change tables using the `MAINTAIN_EDW_CHANGE_RECORDS` procedure. There is a one-to-one correspondence between each fact table and its associated change table.

In addition to processing the records in the change tables, the Banner EDW Incremental Refresh process incorporates records in a fact table's corresponding stage ERROR table, which would contain any records that previously had cleansing “errors”. During the refresh, the ERROR table records' key values are entered into the corresponding change table and processed along with the change table records. Once combined with the change table records, the error table records are purged. However, any records with new cleansing errors are entered into the ERROR table during the refresh. They remain in this table till the next Incremental Refresh or Load process is run for that fact table or star.

This approach is easy to maintain and has negligible impact upon the production environment.

Typically, a complete load is run once, and then the Incremental Refresh processes for both the Banner ODS and the Banner EDW are run on a nightly basis to keep the data in sync. An incremental refresh should be run if data in the Banner ODS has changed since the last time a refresh was run.

The following image outlines the incremental refresh flow:



1. Submit the refresh job from the Administrative UI (Schedule a Process).
2. Select the Refresh process parameter.
3. During the ODSMGR refresh, before the Banner-side Banner ODS change tables are purged, insert records into EDWSTG change tables via the `MAINTAIN_EDW_CHANGE_RECORDS` procedure.

4. Insert the corresponding fact table's ERROR records' key values into the change table.

5. Use the Refresh process parameter in conjunction with the change table records.

6. Select the records to be processed.

7. Run the PL/SQL UPDATE process to delete records from the fact table to be refreshed.

8. Cleanse the refresh records.

9. Enter any records with cleansing records into the Error table.

10. Update/insert dimension records.

11. Insert refreshed records into the fact table.

12. Repeat for all jobs submitted until complete.

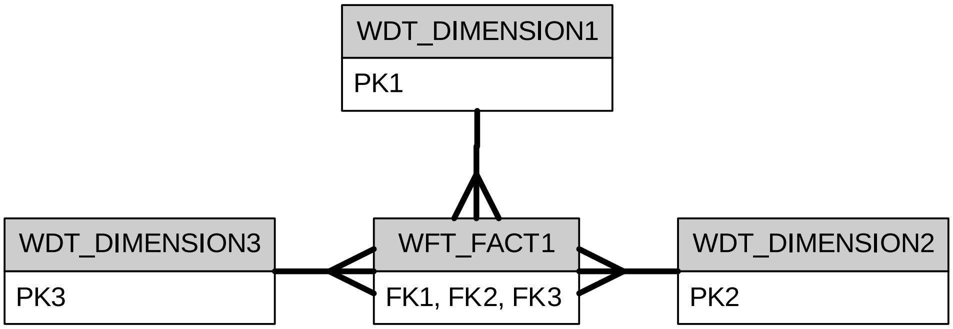
13. Verify the log file for a successful load and completion of all jobs.

Load process vs. Incremental Refresh process

Occasionally, a lot of data changes in the source system and, therefore, in the Banner ODS. For example, via a data import and grade rolls at the end of an academic period. These changes generate a lot of data in the Change tables. This can slow down the Incremental Refresh process. In these cases, it is more efficient to run the Load process instead of the Incremental Refresh process (for those affected tables). Determining when it is more efficient to run a load versus a refresh is somewhat subjective, and can differ between the Banner ODS composite views and the Banner EDW stars.

Banner EDW Star schemas

Star schemas are a standard dimensional data modeling technique used to design data warehouse reporting structures. Each star schema contains a centralized fact table and its associated dimension tables, which are typically referenced through foreign keys. The fact table stores measures while the dimension tables store attributes that users can use to sort, filter, and group the measures. These structures are referred to as “star schemas” because of their star-like appearance when viewing their entity relationship diagrams (ERDs). The following picture illustrates a simple fact and dimension table relationship including foreign key relationships.



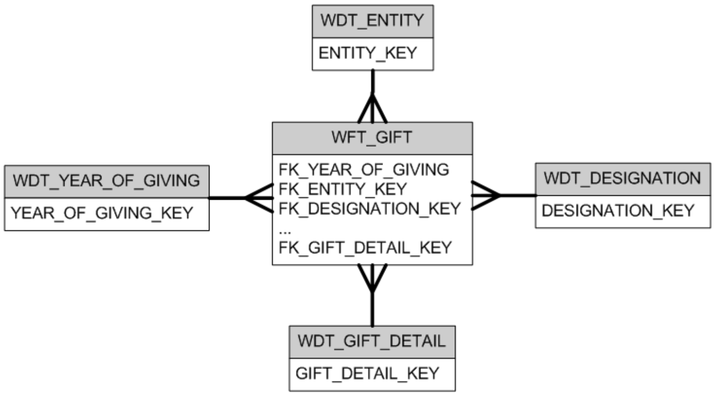
The Banner Enterprise Data Warehouse (Banner EDW) product provides the following types of star schema data models:

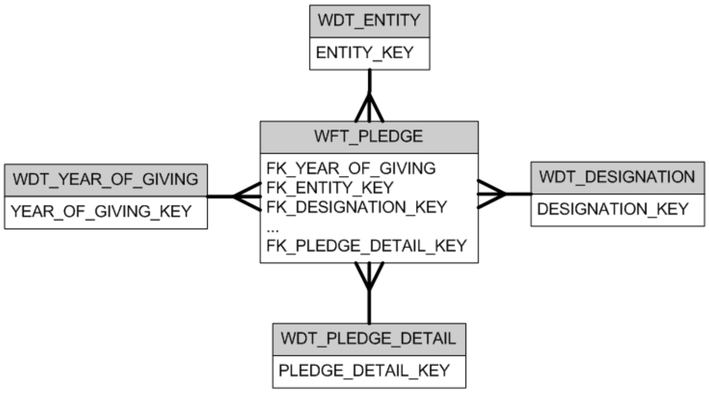
- "Operational stars"
- "Aggregate stars"
- "Snapshot stars"

Operational stars

Operational Star schemas support analysis of a particular subject (for example, gifts, pledges, and so on) by multiple dimensions such as year of giving, designation, and so on. Many of these dimensions are used (or conformed) across multiple operational stars. The operational stars are intended to be refreshed on a daily basis. Currently these operational stars are provided for recruiting and admissions, financial aid for new enrollments, student, and advancement subject areas. The fact tables for operational stars use the WFT prefix.

The following figure illustrates two operational star examples for advancement, gift, and pledge. Some of the dimensions are shown for each fact, including multiple shared dimensions.



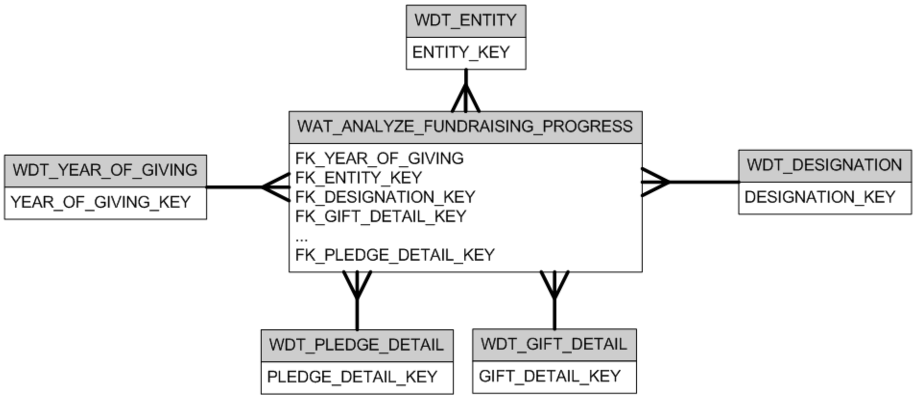


Several operational stars provide the option to create a snapshot star. The snapshot star provides the same data as the operational star, however, the data is saved for a specific point in time by adding the event dimension to the operational data.

Aggregate stars

Aggregate star schemas consolidate information from multiple operational stars to support faster performance analysis of a business concept such as Manage Prospect Pipeline or Analyze Fundraising Progress. The aggregate stars are intended to be refreshed on a daily basis. The Cognos packages for each business concept primarily use these aggregate stars as a data source to optimize performance. The fact tables for aggregate stars use the WAT prefix. Currently these aggregate stars are provided for recruiting and admissions, financial aid for new enrollments, student, and advancement subject areas.

The following figure illustrates an example of the Analyze Fundraising Progress aggregate star for Advancement. This aggregate star consolidates data from multiple operational stars (including gifts and pledges) into the WAT_ANALYZE_FUNDRAISING _PROGRESS fact table with foreign key relationships to several dimension tables.



Snapshot stars

Snapshot Star schemas capture data at institution-specified points in time for a business concept. Most of the aggregate business concepts provide the option to create a snapshot star. The snapshot star provides the same data as an aggregate star, however the data is saved for a specific point in time by adding the event dimension to the data. This event dimension time period could represent, for example, end of month processing for financial data, start or end of academic period for enrollment data, and so on. This allows your institution to historically build data that you can compare over time for longitudinal reporting flexibility. You can use the common event attribute to compare disparate time periods to one another.

There are two variations of snapshot star schemas in the Banner EDW. The first supports a single business need with a set of data combined into a single snapshot star schema sourced directly from the Banner Operational Data Store. For these, the snapshot includes a fact and dimension tables with all of the data to be used as dimension attributes with their associated specified measures. Cognos packages and cubes are delivered with the Banner EDW that use these snapshot stars as a data source.

For most of the business concepts supported by the operational and aggregate stars, a different snapshot process is used. For these, a set of operational and aggregate star schemas is saved at the same point in time with the same event dimension or attributes. Snapshot Cognos packages and cubes are delivered with the Banner EDW which contain the same attributes as the business concept packages and cubes along with an event dimension.

Warehouse tables

The warehouse includes the fact and dimension tables that are used to build the star schema data models.

Fact tables

The fact table is the primary table in a star schema that stores the numerical or event based performance measurements of the institution. Fact tables store dates, amounts or counts of information. Fact table names begin with `WFT_`. Total Credits, stored in the `WFT_ENROLLMENT` fact table, and Prospect Target Amount, stored in the `WFT_PROSPECT` fact table, are examples of measures.

Dimension tables

Dimension tables contain the descriptive attributes that define how you want to slice or look at the measures in a fact table. Dimension table names begin with `WDT_`. Gender and Citizenship_Ind are examples of dimension attributes on the `WDT_DEMOGRAPHIC` dimension table.

View fact and dimension tables

Meta Data is provided for the fact and dimension tables associated with each star. You can use either the published Meta Data reports or the Administrative User Interface (UI) related Meta Data pages to view meta data. To view fact and dimension table information displayed on Banner EDW Meta Data Reports, perform the following steps.

1. Log into the Administrative User Interface.

2. Select the **Meta Data** menu.

3. Select **Banner Enterprise Data Warehouse**.

4. Select the star you want to review. The Star Report for that star opens.

5. Select the target dimension or fact table you want to review. The selected report displays.

Note

Refer to the “Administrative User Interface” chapter of the *Banner EDW Administration Guide* to learn more about the meta data. ■

Banner EDW load process for snapshot stars

Use the following steps to move data from the Banner ODS to Banner EDW snapshot stars:

1. Extract data from the Banner ODS based upon parameters passed from the Administrative UI. This data is loaded into the INPUT table associated with the business area being loaded.

2. Load information within the INPUT table to the associated CLEAN table and run the cleansing process. The cleansing process uses values defined by the institution within the Administrative UI to manage descriptions, translate codes, and update them in the CLEAN table.

3. Use data from the CLEAN table to discern the unique combinations of dimensional attributes within the data extracted. New combinations of attributes are inserted into their associated dimension tables and assigned a surrogate key. The first dimension analyzed is the time dimension. If the combination of dimensional attributes within the time dimension already exists, the loading process halts unless the Replace Indicator check box is checked. This ensures that historical data is not overridden unless explicitly requested by an institution.

4. After loading the attributes into the dimension tables, join the CLEAN table with its various associated dimension tables to obtain the surrogate keys associated with each record. This data is loaded into the associated WKEYS table.

5. Run the `FACT_DELETE` mapping to delete records in the fact table for the defined time slice when the Replace Indicator check box is checked.

6. Load data from the WKEYS table into the fact table.

Banner EDW load process for operational stars

Use the following steps to move data from the Banner ODS to Banner EDW operational stars:

1. Extract data from the Banner ODS based upon parameters passed from the Administrative UI. This data is loaded into the INPUT table associated with the business area being loaded.

2. Load information within the INPUT table to the associated CLEAN table and run the cleansing process. The cleansing process uses values defined by the institution within the Administrative UI to manage descriptions, translate codes, and update them in the CLEAN table.

3. Use data from the CLEAN to discern the unique combinations of dimensional attributes within the data extracted. New combinations of attributes are inserted into their associated dimension tables and assigned a surrogate key.

4. After loading the attributes into the dimension tables, join the CLEAN table with its various associated dimension tables to obtain the surrogate keys associated with each record. This data is loaded into the associated WKEYS table.

5. Load data from the WKEYS table into the fact table.

Banner EDW Stars Reference Guide

The Banner EDW Stars Reference guide includes summary information about each of the stars in the warehouse with hypertext links to each of the star diagrams. The guide is delivered as zip file that is a compilation of the following PDF files:

- *Banner Enterprise Data Warehouse Stars Reference* guide
- Star diagram PDF files

The Banner EDW Stars Reference guide is delivered in the documentation zip file with your product release. You can also download the zip file from the Customer Support Center using the following steps:

1. Log in to the Customer Support Center.

2. Select **Documentation & Download Center**.

3. Select Product Cross Product-Enterprise Data Warehouse and click **List Available Documentation**.

4. Under Release 8.x, choose 8.x Data Model Guides and click **List Files**.

5. Choose which version of the Banner EDW Stars Reference guide that you want to download and click **Prepare Files for Download**.

6. Select **Click Here to Download Zip File** to confirm the download.

Cubes

Cubes are basically precalculated reports with data that you can rearrange and reformat. They provide the ability to manipulate predefined facts (measures) and dimensions (attributes) in various formats to provide different perspectives on an institution's business. Refer to the "Cubes" chapter of your Administration Guide for detailed star schema and cube information.

Using Cognos cubes as the user interface, you can browse data contained within the subset of the star schema in Banner EDW. This interface provides a predefined descriptive view of the information that enables you to understand data without requiring any detailed understanding of a database query language. The presorted data loaded into the cube can be retrieved quickly and can permit multiple dimensions and measures to be selected and reviewed as desired.

Cognos Transformer provides the Cognos ETL equivalent of OWB for the loading of the Cognos cubes. Transformer provides the ability to define relationships within your data warehouse and pre-aggregate the measures presented to end users within the cubes.

Cleansing

Data cleansing is the process of verifying the Banner ODS code values and if possible, translating them to standardized code values in Banner EDW. The ETL mappings initially load code values and descriptions from the Banner ODS into Banner EDW cleansing tables. Using the Administrative UI, the data warehouse administrator can set up cleansing rules specific for your institution.

You can use cleansing to accomplish the following activities:

- Translate a code value in the Banner ODS to a new value in the Banner EDW.
- Change a Banner ODS description value to a new value in the Banner EDW.
- Group a range of Banner ODS code values into one Banner EDW value.
- Translate multiple Banner ODS values into one Banner EDW value and description.

- Associate an effective date with code descriptions that can change over time.

User-defined fields

Banner EDW provides five additional user-defined fields on every dimension and fact table which allow you to extend the data in the warehouse. Including these user fields within the product tables and their related mappings means you need to make minimal changes to bring new data into the warehouse. The dimension user-defined fields are named USER_ATTRIBUTE_01-05, and the fact and aggregate user-defined fields are named USER_MEASURE_01-05. Follow the guidelines in this section to use the user-defined fields.

General guidelines

Considerations should be made when deciding what data to add to the user-defined fields. Use the following questions as guidelines when determining what data to add and where to add it.

What source data to add?

First, you must evaluate what the source of the new data is and whether this source is available within the source system and/or Banner ODS – these guidelines assume the data already exists in the staged source tables and/or Banner ODS and is available for you to use. If not, additional steps will be needed to stage the data from the source system and/or add it to the Banner ODS.

How will the new data be used?

Next, consider how the data will be used. Will it be used as an attribute or a measure? This will tell you whether it should be added to a dimension (attribute) or fact (measure) table.

If you are adding a new attribute to a dimension table shared by multiple organizations at the institution, consider whether or not those organizations will be interested in the same content. If so, you should gain input from all offices regarding the details of the new content so that you can create common attributes. If multiple organizations are not interested in the same content, consider adding the new data to a dimension table not shared by other organizations or to two different user attribute fields. This will enable user attribute columns to be optimally distributed among the storage structures while minimizing the need for customization.

Where to add data?

When trying to determine which dimension or fact table to add the user attribute to, look for dimensions or facts that contain similarly sourced data. Looking through the meta data using the Administrative UI can help you see where similar data already exists.

Is the data at same granularity as other data in star?

You must also consider whether the new data item is captured at the same level of granularity as the other data in the star. These guidelines assume the data is at the same level of granularity. If data is at a different level of granularity, additional changes will be needed within the ETL, table definitions, report structures, or the query retrieving the data.

Modifying code to support user-defined fields

Modifications to the delivered, baseline %EXTR packages will generally support all required modifications to Banner EDW source code associated with the warehouse Oracle environment. Those packages are *not* created as a part of deployed Banner EDW mappings but installed with other coded source.

Following are recommendations to keep in mind when modifying user-defined fields.

- Be sure to follow institution-defined processes for modifying, tracking, and documenting modifications to baseline code so that you can easily understand your changes and reapply them in future releases if necessary.
- Copy the table function package script (%EXTR) to an institution specific directory outside of the baseline EDW code tree.
- Rename the script to indicate that it is modified.
- Do not change the name of the object created by the script. Changing the name will require changes to the related OWB %INPUT mapping and redeployment of that mapping as well as recompiling dependent database objects.
- Construct your enhancements to the baseline source with consideration for other business areas as well as source management and testing processes during subsequent upgrades.
- Update the return row (ret_row) population section of the script to assign the desired attributes or measures to the appropriate dimension or fact table user column, or to modify the calculation of an existing column (such as age).

Terms - table function structure

The following components are used in the table function structure.

- Driving cursor – selects population that will drive extract

- Supporting cursors – select supporting data related to driving population
- Processing statements
  - Invoke the defined cursors to return the data as expected within the TABLE() function, including processing loops to support processing individual rows for, at a minimum, the related institutional entities and the rows/records returned by the driving population.
  - Return row definition – where selected values are assigned to staging input table columns for each row/record processed.
  - Initialization of interim storage components prior to ending each iteration of the processing loop.

Conformed dimensions

The Banner EDW uses conformed dimensions, which are dimensions shared among stars. When modifying a dimension that is shared by many stars, you need to consider the following

Will each star using a conformed dimension use the same user-defined attributes?

- If “Yes” - you must update each star’s table function and extract mapping to select the new user-defined field.
- If “No - (meaning you want to add the user-defined field to one star but not another) there is no issue, the other stars do not need to be modified and would continue to populate the field with a null value.

Will each star using a conformed dimension need different user-defined attributes for different stars?

Since dimensions are shared between stars, each user-defined field must have one unique source and thus one unique cleansing rule across all stars that share that dimension. This means that if you wish to add two *different* user-defined attributes to the same dimension table but for different stars, you must use two different user attribute fields. For example, if you want to add hair color and legislative district to the WDT_DEMOGRAPHIC dimension for different stars, you must use USER_ATTRIBUTE_01 for one attribute and USER_ATTRIBUTE_02 for the other.

Steps to add user-defined attribute and measure values to the warehouse

Step 1 Update the extract logic that selects values from the source tables to populate the warehouse tables.

With the exception of Advancement operational stars and all aggregate stars which are populated via OWB mappings, table functions as found in %EXTR packages are used to extract the data from the Banner ODS to the Banner EDW for loads and refreshes.

The steps needed to add user-defined attributes and measure values to the warehouse depend on the type of star you want to update. Refer to the steps in the following sections to select the appropriate source value and populate the appropriate warehouse target table and column.

- “For snapshot stars” on page 2-48
- “For operational stars using the architecture which sources Banner EDW from the Banner ODS (non-Advancement operational stars)” on page 2-49
- “For Advancement operational stars using the architecture which sources Banner EDW directly from the staged source system tables” on page 2-50
- “For aggregate stars regardless of architecture” on page 2-50
- “For frozen stars regardless of architecture” on page 2-51

For snapshot stars

To add values to the user-defined attribute or measure fields, you only need to update the associated extract package for the star (or if appropriate an extract package, such as EDW_GENERAL_EXTR, used to populate columns in both the snapshot and operational stars). If the change relates to a dimensional attribute, you must also update the corresponding cleansing rule. No OWB changes are needed.

1. Update the snapshot star’s extract package table function with desired changes.

   There is one primary extract package per snapshot star. The naming convention is EDW_<snapshot star>_EXTR. For example, the Snapshot Employee Star’s extract package is EDW_EMPLOYEE_EXTR. However, other %EXTR packages such as EDW_GENERAL_EXTR may be used to support population of columns common to both Snapshot and Operational stars.

2. Within the package, locate the corresponding driving or supporting cursor which will select your attribute or measure and update it. Most (though not all) changes will be made within the %pop cursor.

3. Within the package, locate the return row definition for the staging table input column corresponding to your user-defined attribute or measure and update it to be populated with the modified cursor value. Note that some changes, such as changes to calculate

age based on values such as the first day of an academic period (rather than sysdate) will only require changes to the return row.

4. Recreate the package under the EDWSTG schema.

5. Grant execute on the package to IA_ADMIN.

   This ensures that permissions were not lost. Remember that you run load and refresh jobs from the Administrative User Interface using records in the IA_ADMIN schema tables.

Note

Population processes for snapshot stars will only impact data populated in the related Star Schema *after* the change and related Cleansing setup is completed. You will *not* be able to reload data to existing snapshots except for those milestone events related to 'Final State' data storage in the source system (such as 'final' for academic period or fiscal period.) Reloading milestone events (not 'final') will invalidate the time perspective of the data. ■

For operational stars using the architecture which sources Banner EDW from the Banner ODS (non-Advancement operational stars)

To add values to the user-defined attribute or measure fields, you only need to update the extract package and if it is a dimension attribute, the corresponding cleansing rule. No OWB changes are needed.

1. Update the Operational Star's extract package containing its table function with desired changes.

   For operational stars, there is one extract package per product area. The naming convention is EDW_<product>_EXTR. For example, all student operational stars table functions are contained within the extract package EDW_STUDENT_EXTR. The product extract packages are EDW_GENERAL_EXTR, EDW_STUDENT_EXTR, EDW_FINAID_EXTR and EDW_RAP_EXTR.

2. Within the package, locate the table function that populates the operational star to which you want to add the user field.

3. Within the table function, locate the corresponding driving or supporting cursor which will select your attribute or measure and update it to select your new field.

   Since the cursors in these packages can be shared across stars and packages, be sure this change will not negatively impact other stars. For example, the EDW_STUDENT_EXTR package makes repeated reference to cursors in other packages such as EDW_GENERAL_EXTR.

4. Within the identified table function for the star to be updated, locate the return row definition for the staging table input column corresponding to your user-defined attribute or measure and update it to be populated with the modified cursor value.

5. Recreate the package under the EDWSTG schema.

6. Grant execute on the package to IA_ADMIN.

   This ensures that permissions were not lost. Remember that load and refresh jobs are run from the Administrative User Interface using records in the IA_ADMIN schema tables.

For Advancement operational stars using the architecture which sources Banner EDW directly from the staged source system tables

To add values to the user-defined attribute or measure fields, you need to update the extract OWB mappings and if it is a dimension attribute, the corresponding cleansing rule.

1. Locate your mapping within the appropriate OWB Project – either APM_GENERAL or APM_ADVANCEMENT. Source specific mappings end with the suffix ‘A’ for Advance or ‘B’ for Banner Advancement.

2. If you are adding an attribute, update the associated D1 mapping within the Project>Oracle>EDWSTG module, for example, <dimension>_D1A or <dimension>_D1B, to map the new value to the dimension’s user attribute.

3. If you are adding a measure, update the associated F1 mapping within the Project>Oracle>EDWMGR module, for example, <fact>_F1A or <fact>_F1B, to map the new value to the fact table’s user measure.

4. Save the changes.

5. Deploy the mapping.

For aggregate stars regardless of architecture

Use the following steps to add values to the user-defined attribute or measure fields of an aggregate star.

1. Locate the aggregate mapping within the appropriate OWB Project.

   - There are no source specific aggregate mappings.
   - If changing an Advancement aggregate, the aggregate mapping naming convention is <aggregate>_A1
   - If changing a non-Advancement aggregate, the aggregate mapping naming convention is EDW_<aggregate>_AGG_INSERT

2. Map the new value to the aggregate star's user measure. It is likely that the user measure for an aggregate will be sourced from an updated operational star's user measure.

3. Save the changes.

4. Deploy the mapping.

For frozen stars regardless of architecture

Follow these guidelines when updating frozen stars.

- The corresponding operational or aggregate star which sources the frozen one must be modified to include the user measure.
- Once the corresponding operational or aggregate star is updated, no changes are required to populate the frozen star, links between the frozen star's user measure fields and the corresponding star's user measure fields are already in place.

Step 2 Link cleansing rules to new user-defined dimension attribute.

Note

This step only applies if you've added a user-defined dimension attribute; it does not apply to user-defined measures. ■

Perform the following steps within the Administrative UI.

1. Select Options>Set Up and Maintain Cleansing Processes.
   - If the user-defined value you've added already can be linked to an existing Cleansing rule, skip to the next step.
   - Else, go to Set Up and Maintain Cleansing Rules and create a new cleansing rule with appropriate source query. See the Set Up Cleansing chapter for more details.

2. Select Options>Set Up and Maintain Cleansing Processes>Set Up and Maintain Cleansing Data Elements.

   2.1. Select the dimension table that will be updated with your user-defined attribute and select **Search**.

   2.2. Select the USER_ATTRIBUTE column 01-05 you modified in Step 1.

   2.3. Within the Update a Cleansing Data Element screen, update the **Rule Name** field with the appropriate cleansing rule.

   2.4. Click **Save**.

Your new user-defined attribute value will now be cleansed by the appropriate cleansing rule.

Step 3 Update meta data to reflect the change (Optional)

If you use the published meta data to document source to target relationships, refer to the "Meta Data (Banner ODS and Banner EDW)" section of the Administrative User Interface chapter to add corresponding data for the new column.

1. Create a new target column for the updated dimension or fact table, naming the column as it is in the database, i.e., USER_ATTRIBUTE_01-05 or USER_MEASURE_01-05.
2. Give the column a new Business Name, Definition, and other information to match the new use for the column.
3. Save the changes.
4. From the same target column page, select Add Local Mapping at the bottom.
5. Select and save the appropriate source details.
6. Publish meta data.

The new column will be available.

Step 4 Reload data via the Administrative UI to populate the new field

1. Select the appropriate jobs to reload the data that will access the new field.
2. If the new field is in an operational star, you must also reload the related aggregate stars.

Note

Population processes for snapshot and frozen stars will only impact data populated in the related star schema after the change and related Cleansing setup is completed. You will not be able to reload data to existing snapshots or frozen concepts except for those milestone events related to 'Final State' data storage in the source system (such as 'final' for academic period or fiscal period.) Reloading milestone events (not 'final') will invalidate the time perspective of the data. ■

3. Confirm that the correct data is loaded into the modified dimension or fact.

Step 5 Update delivered Cognos models to expose and appropriately label the user-defined attribute or measure (Optional)

Note

You only need to perform this step if you are using the delivered Cognos content. ■

The delivered Cognos Framework Manager Models have the following three layers.

- Database View
  - Reference to database objects
  - Table names and select columns given business names
  - User attribute and measure columns already present in database layer but do not have business names
  - Fact tables obtain “Measure” suffix
  - Some concatenated unique keys added
  - Some formatting and aggregation defined
- Business View
  - Business-centric grouping of objects
  - Business-centric table aliases created
  - Business-centric relationships/joins created
  - Alternative aggregation query items defined, for example, averages, headcounts, percents, counts
- Presentation View
  - End user experience/layer that is published to the end-users
  - Commonly used items grouped together
  - Folders used to remove clutter
  - Commonly used filters, calculations created
  - Customizable parameter driven items created

Perform the following steps to update the Cognos packages.

1. Update the Database View layer for the table/column you modified.

2. Rename the USER_ATTRIBUTE/MEASURE_01-05 and their corresponding short description (SD) and long description (LD) columns in the table you modified to have business names.

3. Update the Business View layer to use these new names.

4. Update the Presentation View layer to include the new code and description fields in the appropriate Query Subject.

5. Update the appropriate Package Definitions to select the new fields to be published.

   You will then be able to access the new fields via the newly published packages and use them through the various Cognos Studios such as Report Studio and Query Studio.

6. Update the Cube transformer model with the new columns and republish it. (Optional)

Multi-Entity Processing

The Multi-Entity Processing (MEP) framework is available for all target database composite views, composite tables, and reporting views. This enables all information from multiple sources (data sources, institutions, campuses, etc.) that is located in one database to be selectively assigned security access as needed in the target database.

Example:

> You can take existing data from one database for use in multiple institutions, move information into the target database, selectively restrict the user access to data by institution, and so on.

The MEP columns only appear on generated meta data reports in the Administrative UI if MEP is set up for your institution.

Note

> To use MEP with your source system and target database, Professional Services must provide the needed analysis, subsequent product enhancements, and set up. This includes identifying source tables that require MEP, and the target database objects to be modified. ■

Administrative User Interface

The Administrative User Interface (UI) is Web-based and uses Banner Web Tailor. The Administrative UI is used to set up and maintain the target database and warehouse, including initiating and monitoring ETL processes. Administrative functions include:

- Preferences and Security - Use to manage security, set global preferences, and set up user accounts.
- Options - Use to control the processes to extract and load data into Banner ODS and Banner EDW, schedule a process, view control reports, view and/or remove scheduled processes, and maintain information about saving (freezing) data.
- Meta Data - Use to view and manage the meta data supporting the systems.
- New Banner Web Tailor Administration - Use to customize a Web menu, procedure, graphic element, set of information text, or a set of menu items. You can also update user roles, customize a Web module, Web rules, or Banner Web Tailor parameters; customize a login return location; and customize Banner Web Tailor overrides or global user interface settings.

Banner ODS data model

Banner ODS delivers a data model that includes data from a number of higher education administrative system modules. The administrative system modules supported by the Banner ODS data model include Student, Financial Aid, Advancement, Human Resources and Finance — including Accounts Receivable. Each module, or area of information, includes a number of tables in the administrative systems. The data model brings the appropriate data elements, from multiple tables in the source system, into a different table structure in the Banner ODS to support the reporting needs of the entire institution.

The data model represents the data elements that are included in Banner ODS. Banner ODS shows the individual table and the relationship with other tables stored within the model. It further includes all the data elements available in Banner ODS composite tables and/or the reporting views related to the object described.

Multiple source databases

The EDW architecture supports stage tables from different source databases. The only requirement to load information from multiple sources into the EDW stage tables is that the schema and table names in the source databases must be unique.

Note

Because the schema and table names in the source databases must be unique, you cannot load information from two different Banner databases into the EDW.

Source Alias

The Source Alias (`source_alias`) uniquely identifies each source database. You specify the `source_alias` during the installation or upgrade process. The `source_alias` is then used to create a parameter in Web Tailor, which associates each `source_alias` to a database link owned by ODSSTG. This approach allows the database link to the source to be changed while minimizing the disruption to the existing EDW functionality.

Source Alias in Streams framework

In the Streams framework, the Source Alias is used as a prefix when naming the various Streams components. The prefix identifies the source database and the suffix identifies the Streams component. For example, the Streams component BANNER$APP is associated with the source alias of BANNER, and is an apply process.

The following table lists the database location and suffix for each Streams component. The Source Alias is added to the beginning of each Name Suffix to uniquely identify the Streams component.

Streams Component	Database location	Name Suffix
Capture	Source	$CAP
Capture queue	Source	$CAPQ
Capture queue table	Source	$CAPQT
Propagation	Source	$PROP
Apply queue	Banner ODS	$APPQ
Apply queue table	Banner ODS	$APPQT
Apply	Banner ODS	$APP

Add a source database

Use the following steps to add subsequent source databases after an initial source database has been configured.

1. Run the source install steps on the source database. Refer to the Banner ODS Installation or Upgrade Guide for the source install steps.

   This creates the ODSSTG administrative user with the necessary privileges, compiles the support package, and creates a database link from the ODSSTG user to the ODSSTG user on the EDW.

2. Create a database link connecting the ODSSTG user on the EDW instance to the ODSSTG user on the new source.

Streams users only perform step 3; Mviews users can skip to step 4.

3. As the ODSSTG user on the EDW, execute the following procedure from SQL*Plus.

```
   SQL> SET SERVEROUTPUT ON
   SQL> MGKSTRC.P_CREATE_LOCAL_ENV(database link, source
alias);
```

   where you enter your institution’s values for the parameter in parentheses.

4. Create schemas in the EDW.

For each schema in the source database that includes tables that will be staged in the EDW, create a schema in the EDW with the same name.

5. Add the new schemas to the EDW using the Administrative User Interface.

   5.1. Follow the steps in the section “Add a schema” in Chapter 4, “Administrative User Interface”.

   5.2. Repeat step 5 for each new schema you want to add to the EDW.

6. Stage new tables in the EDW using the Administrative User Interface.

   6.1. Follow the steps in the section “Add a non-baseline staging table to the Banner ODS” in Chapter 4, “Administrative User Interface”.

   6.2. Repeat step 6 for each new schema you added.

Validation table data and incremental refresh

The Banner Operational Data Store (Banner ODS) was designed with validation table codes and descriptions stored on each individual data record. This design expedites the display of information as it eliminates the need for excessive joins of as many as ten or fifteen additional tables. During the design phase of Banner ODS, several methodologies on managing validation table change requirements were discussed with institutions. The consensus was that it is preferable to build internal institutional policies and procedures to ensure that the descriptions are not changed, but new codes are added.

This is similar to the way in which Banner Course Catalog process works. If the title of the course changes, the institution creates a new catalog record with the new title for the new effective term.

Example

If a description such as “Bowling Basics” changes to “Bowling Fundamentals”, it is assigned a new code so that Banner ODS reflects the past data for “Bowling Basics” and the new values are reflected for “Bowling Fundamentals”.

To change a column description, the institution policy requires to either initiate a reload of all affected tables (time intensive) or create a script to update all columns in Banner ODS to alter the old value to the new value.

Note

To ensure data integrity, do not apply updates to existing values in the validation tables once Banner ODS is in production and the incremental refresh cycle is implemented. Else, there will be inconsistencies in the information displayed between the source system and Banner ODS. ■

To further explain the difficulty in incrementally refreshing tables based on coded description changes and not the result of data value changes, it is necessary to understand the efforts required to implement a validation to data table refresh. First, the source system would have to be enhanced to maintain triggers on each validation table to track all DML activity. While it is possible to apply triggers to each of these tables, the trigger event is likely to have performance impact on the source system. This is because it requires the trigger to populate an entry into a change table for every row in each source data table that is populated with the altered validation table value. This requires a full table scan of every affected source table as the source system does not maintain keyed links between the validation tables and the data tables.

For example, the validation table STVDEPT is used enterprise wide in Banner Student, Banner Advancement, and Banner HR systems in eighty four (84) different tables. If a value were to be changed in the STVDEPT table, then the trigger on the STVDEPT table would have to read all 84 of the source tables to identify the key(s) of each row that contained the altered DEPT value, and then populate that key into the change table. Given the size of many of these data tables, the commit time required for the end users to wait on the change of the validation table in Banner would freeze their Banner session until the change table population took place.

Table indexes

Indexes are added based on the reporting needs of the Banner ODS and Banner EDW tables as well as performance for the incremental refresh process. The `IA_ADMIN.MGBINDX` table stores a list of the delivered indexes for tracking and documentation purposes. You populate this table using the following query for a release:

```
SELECT &sysid, x.table_name, x.index_name, column_name,
uniqueness, descend,column_position,&relno, 'NO',SYSDATE, 1
   FROM user_indexes x, user_ind_columns y
WHERE x.table_name = y.table_name AND
              x.index_name = y.index_name AND
              x.table_name  LIKE 'M%' AND
              x.table_name NOT IN
            (SELECT DISTINCT table_name
               FROM all_tab_columns
            WHERE column_name LIKE '%FREEZE_EVENT%')
ORDER BY x.table_name, x.index_name, column_position;
```

The `MGBINDX` table is used in the Banner ODS Checks and Balances process to verify that baseline indexes are valid and present. If your institution has created additional indexes, the differences are reported in the control report as warnings. To include the additional indexes in the Banner ODS Checks and Balances process, insert the new index information into the `IA_ADMIN.MGBINDX` table using SQL. Refer to the

MGBINDX_DATA_ODS.SQL script in the dbscripts directory for a syntax example. Set the LOCAL_IND = 'YES' to identify this as your institution's index. The local records in this table will be preserved with future upgrades. We recommend that you do not delete baseline rows from the MGBINDX table.

The Banner ODS metadata also uses the delivered indexes when documenting the Recommended Search Columns. The script update_recsearchconds.sql (located in the dbscripts/utility_scripts directory) is used to generate that information based on the actual indexes in the database. If you add local indexes, it is recommended that you run the script (from the IA_ADMIN account) so the list of Recommended Search Columns accurately reflects the database.

Product-specific information

This section discusses the Banner ODS or Banner EDW information unique to individual Banner products.

Banner Common

The Banner ODS VALIDATION reporting view provides access to all of the Banner product validation table values to be used when creating a pull-down list of values (LOV) for parameters. This reporting view can be used by a variety of reporting tools. The MGT_VALIDATION table is the source for the reporting view and is used to build the LOV views that reside in the ODSLOV schema. The source for the MGT_VALIDATION ODS composite table is a series of composite views listed below. These views retrieve the values from specific product validation tables that are used within the Banner ODS.

Performing a select distinct on a code within a reporting view may be a valid solution to generate a List of Values. However, this method will likely cause a performance impact on the system. The VALIDATION reporting view can instead be used as a pull-down list. It provides the appropriate Banner Validation Table name as a filter for VALIDATION.TABLE_NAME.

The information on the List and Detail Reports pages can be viewed online or exported to a CSV file (Microsoft Excel format) or XML file for printing or additional manipulation. Following are the composite views:

- AA_VALIDATION
- AF_VALIDATION
- AG_VALIDATION
- AN_VALIDATION
- AP_VALIDATION
- AR_VALIDATION

- AS_VALIDATION
- AT_VALIDATION

Each of these Banner composite views extracts values from validation tables in their respective Banner product areas. Also included are the status indicators, effective dates, and sometimes the qualifiers.

Within Banner Finance, there are several groups of values stored within the FTVSDAT System Data Maintenance table. To properly represent some of these values, they have been pulled into the AF_VALIDATION composite view with the TABLE_NAME as follows:

- GRANT_CATEGORY represents all grant categories stored within FTVSDAT.
- GRANT_SUBCATEGORY represents all grant sub categories stored within FTVSDAT.
- GRANT_TYPE represents all grant types stored within FTVSDAT.

Values have been added to table FTVFSPD to represent beginning and ending periods. The added values are ‘00’, ‘13’, and ‘14’. The FTVFSYR table has for its description, the Fiscal Year converted to a four-digit year.

In specific situations, Banner source tables were not used. The following is a compiled list of data element names used in place of Banner specific tables names.

The hard coded TABLE_NAMES are as follows:

- ACADEMIC_TITLE
- ACCOUNT_ATTRIBUTE_TYPE
- ACCOUNT_ATTRIBUTE_VALUE
- ACCOUNT_CLASS
- ACCOUNT_LEVEL_1
- ACCOUNT_LEVEL_2
- ACCOUNT_LEVEL_3
- ACCOUNT_LEVEL_4
- ACCOUNT_POOL
- ACCOUNT_SET_CODE
- ACCOUNT_TYPE_ATTR_TYPE
- ACCOUNT_TYPE_ATTR_VALUE
- ACCOUNT_TYPE_LEVEL_1

- ACCOUNT_TYPE_LEVEL_2
- ACCOUNT_TYPE_SET_CODE
- ADVISOR_NAME_LFMI
- ASSIGNMENT_GRADE
- CALENDAR_MONTH
- CALENDAR_YEAR
- COLLECTION_AGENCY_NAME
- CONTRACT_NUMBER
- CONTRACT_TYPE
- COURSE_IDENTIFICATION
- COURSE_REFERENCE_NUMBER
- EMPLOYEE_STATUS
- EMPLOYEE_TIME_STATUS
- ENDOWMENT_FUND
- ENTITY_TYPE
- FINANCIAL_AID_SOURCE_TYPE
- FINANCIAL_AID_TYPE
- FINANCIAL_MANAGER
- FISCAL_QUARTER
- FUND_ATTRIBUTE_TYPE
- FUND_ATTRIBUTE_VALUE
- FUND_LEVEL_1
- FUND_LEVEL_2
- FUND_LEVEL_3
- FUND_LEVEL_4
- FUND_LEVEL_5
- FUND_POOL
- FUND_SET_CODE
- FUND_TYPE_ATTR_TYPE
- FUND_TYPE_ATTR_VALUE

- FUND_TYPE_LEVEL_1
- FUND_TYPE_LEVEL_2
- FUND_TYPE_SET_CODE
- GENDER
- INSTALLMENT_PLAN
- INSTRUCTOR_NAME
- INTENDED_TIME_STATUS
- INTERNAL_ACCOUNT_TYPE
- INTERNAL_FUND_TYPE
- LOCATION_LEVEL_1
- LOCATION_LEVEL_2
- LOCATION_LEVEL_3
- LOCATION_LEVEL_4
- LOCATION_LEVEL_5
- ORGANIZATION_ATTR_TYPE
- ORGANIZATION_ATTR_VALUE
- ORGANIZATION_LEVEL_1
- ORGANIZATION_LEVEL_2
- ORGANIZATION_LEVEL_3
- ORGANIZATION_LEVEL_4
- ORGANIZATION_LEVEL_5
- ORGANIZATION_LEVEL_6
- ORGANIZATION_LEVEL_7
- ORGANIZATION_LEVEL_8
- ORGANIZATION_POOL
- ORGANIZATION_SET_CODE
- ORG_FINANCIAL_MANAGER
- POSITION_STATUS
- POST_SECONDARY_SCHOOL
- PREF_CLAS

- PRINCIPAL_INVESTIGATOR
- PROGRAM_ATTR_TYPE
- PROGRAM_ATTR_VALUE
- PROGRAM_LEVEL_1
- PROGRAM_LEVEL_2
- PROGRAM_LEVEL_3
- PROGRAM_LEVEL_4
- PROGRAM_LEVEL_5
- PROGRAM_SET_CODE
- RECEIVABLE_CONTRACT
- RECEIVABLE_DELINQUENCY
- RECEIVABLE_EXEMPTION
- SECONDARY_SCHOOL
- SPORTS

Banner Finance

The following table explains the use for FIELD_CODE and LEDGER_IND within the TRANSACTION_HISTORY reporting view for Banner Finance. The LEDGER_IND and FIELD_CODE work together to drive what ledger amount field was updated.

LEDGER_ IND	Ledger	FIELD_ CODE	Amount Field Updated	Description
G	General	01	Sum_Periodic_Debits	Debits
		02	Sum_Periodic_Credits	Credits
O	Operating	01	Curr_Adopted_Budget	Current Period Original Budget
		02	Curr_Budget_Adjustments	Current Period Budget Adjustment
		03	Curr_Year_To_Date_Activity	Current Period Activity
		04	Curr_Encumbrances	Current Period Purchase Order and General Encumbrance
		05	Curr_Budget_Reservation	Current Period Requisition Budget Reservation
		06	Curr_Accumulated_Budget	Current Period Accounted Budget

LEDGER_ IND	Ledger	FIELD_ CODE	Amount Field Updated	Description
		07	Curr_Temporary_Budget	Current Period Temporary Budget
		08	Curr_Grant_Activity	Obsolete
E	Encumbrance	01	Original_Amount	Original Encumbrance Amount
		02	Curr_Adjustments	Encumbrance Adjustments
		03	Curr_Liquidations	Encumbrance Liquidations

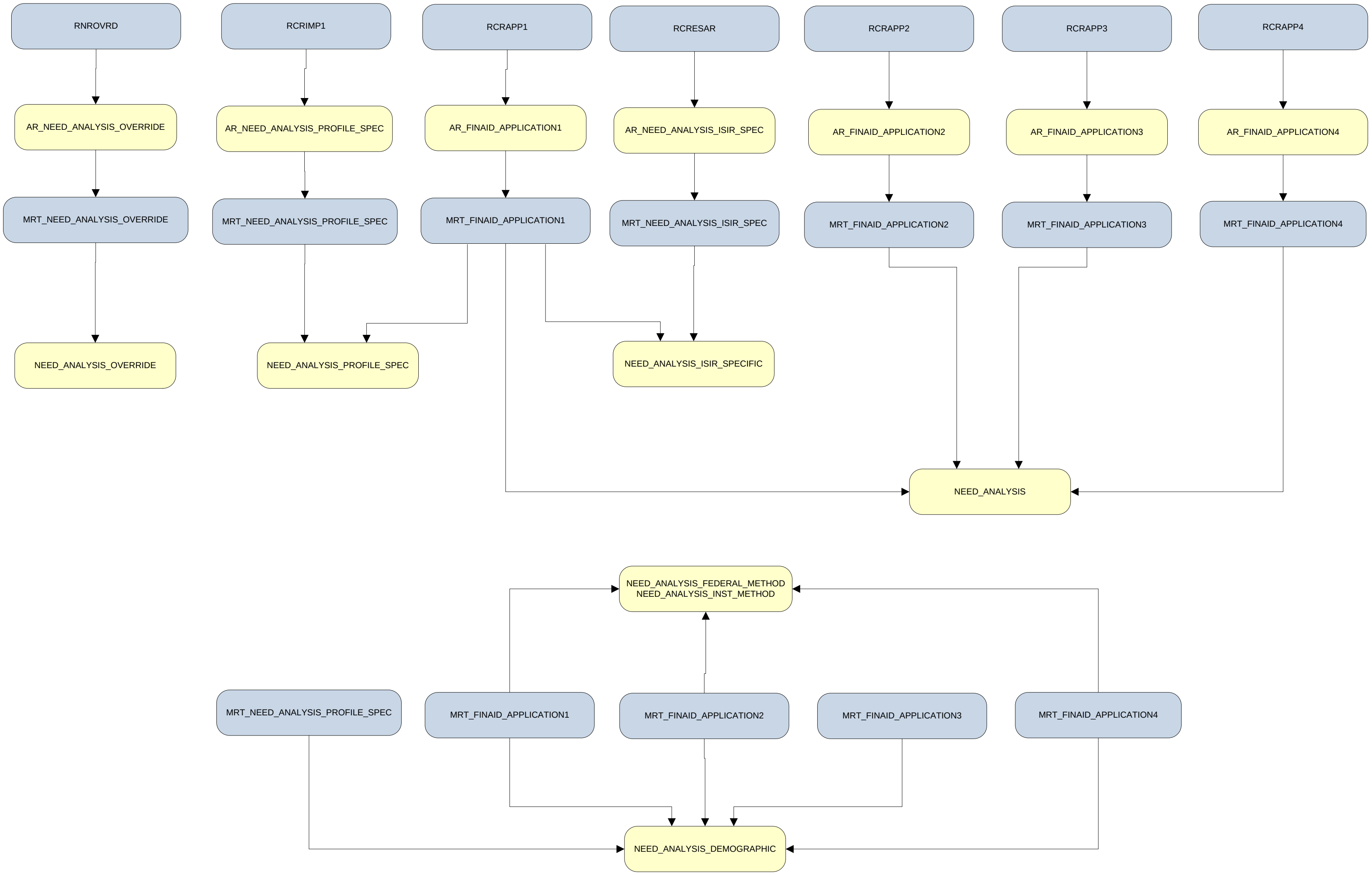
Banner Financial Aid key views architecture

Due to the complex architecture of some Banner Financial Aid views, the following flow charts illustrate how those Banner Financial Aid reporting views are built from Banner to the Banner ODS.

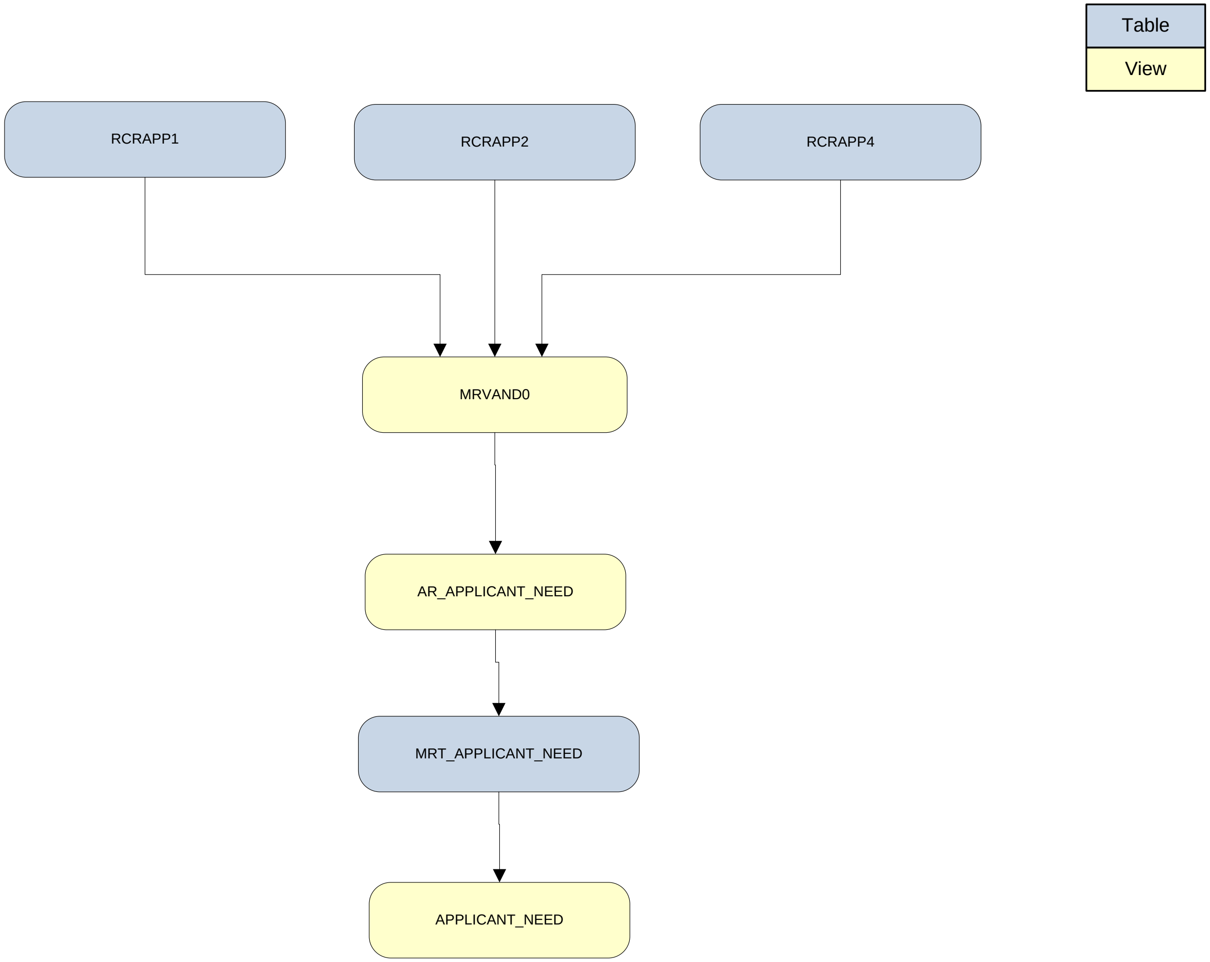
Note

These diagrams only refer to key views and key tables used within the reporting views. ■

Financial Aid Need Analysis reporting view flow



Applicant Need reporting view flow



Banner Student

When a new base student record is created in Banner, a new record is created in the Banner ODS table MST_BASE_STUDENT. Each record in this table contains a range of academic periods in which the student status allows the student to register. If the status prevents the student from registering, then the beginning and ending academic periods in the Banner ODS record are the same and match the Banner effective term.

Banner Student data extraction for the MST_GENERAL_STUDENT composite table

Creating a new record within one of a number of Banner tables indicates to the Banner ODS that the student has activity within the specific term. As a result, a new record is created in the MST_GENERAL_STUDENT table for the student and term when the Banner ODS is loaded or refreshed.

Following is a list of Banner tables that define student activity in the Banner ODS:

- SGBSTDN - student base table
- SFBETRM - student registration table
- SHRTTRM - institutional course maintenance term header table
- SHRTRAM - attendance period by transfer institution table
- SHRDGMR - degree table
- SGRCHRT - student cohort table
- SGRSPRT - sport table
- SGRSATT - student attribute table
- SGRSACT - student activity table
- SGRCOOP - cooperative education table
- RPRATRM - applicant award by term table
- RORSTAT - applicant status table
- TBRACCD - account charge/payment detail table
- TBBCSTU - contract student authorization table

The MST_GENERAL_STUDENT table also contains information about each student's program of study. This table contains one record per student per academic period with student activity per curricula.

Additional 'Zero' record in the Banner ODS tables

In Banner, the values for student classification and academic standing are specific for a student, academic period, and their primary program level only. In the Banner ODS, many reports require student classification and academic standing data for all student curricula, regardless of the level value. To create comprehensive reports while limiting the number of outer-joins used, a single record with a value of zero for the key fields (person_uid, student_level, and academic_period) is inserted into the `MST_STUDENT_CLASSIFICATION` and `MST_ACADEMIC_STANDING` composite tables as a step in the load mappings. Existing student classification and academic standing values are displayed if they exist for a specified student, level, and academic period. Otherwise, the NULL values from this new record are displayed.

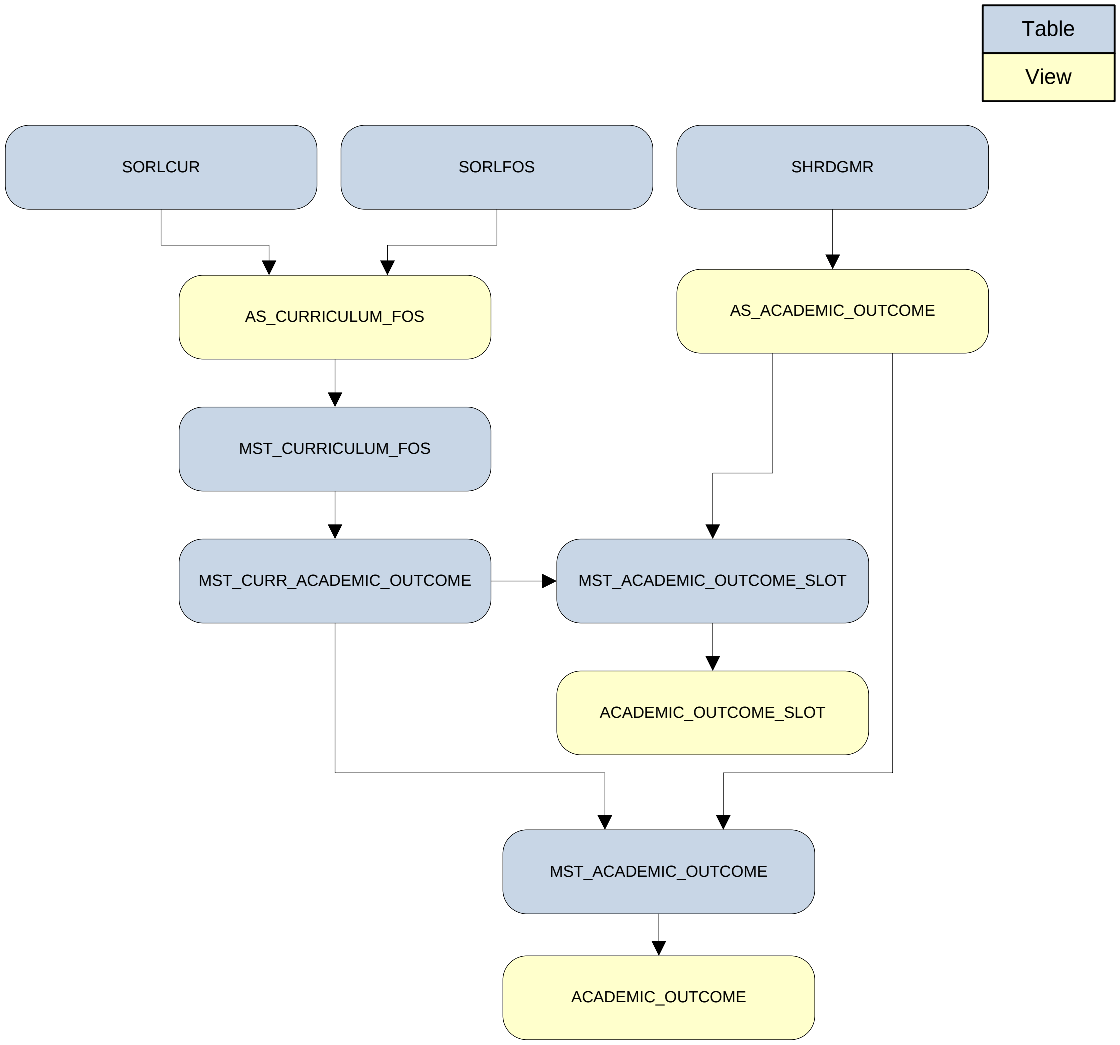
Key Banner Student views architecture

Due to the complex architecture of some Banner Student views, the following flow charts illustrate how those Banner Student reporting views are built from Banner to the Banner ODS.

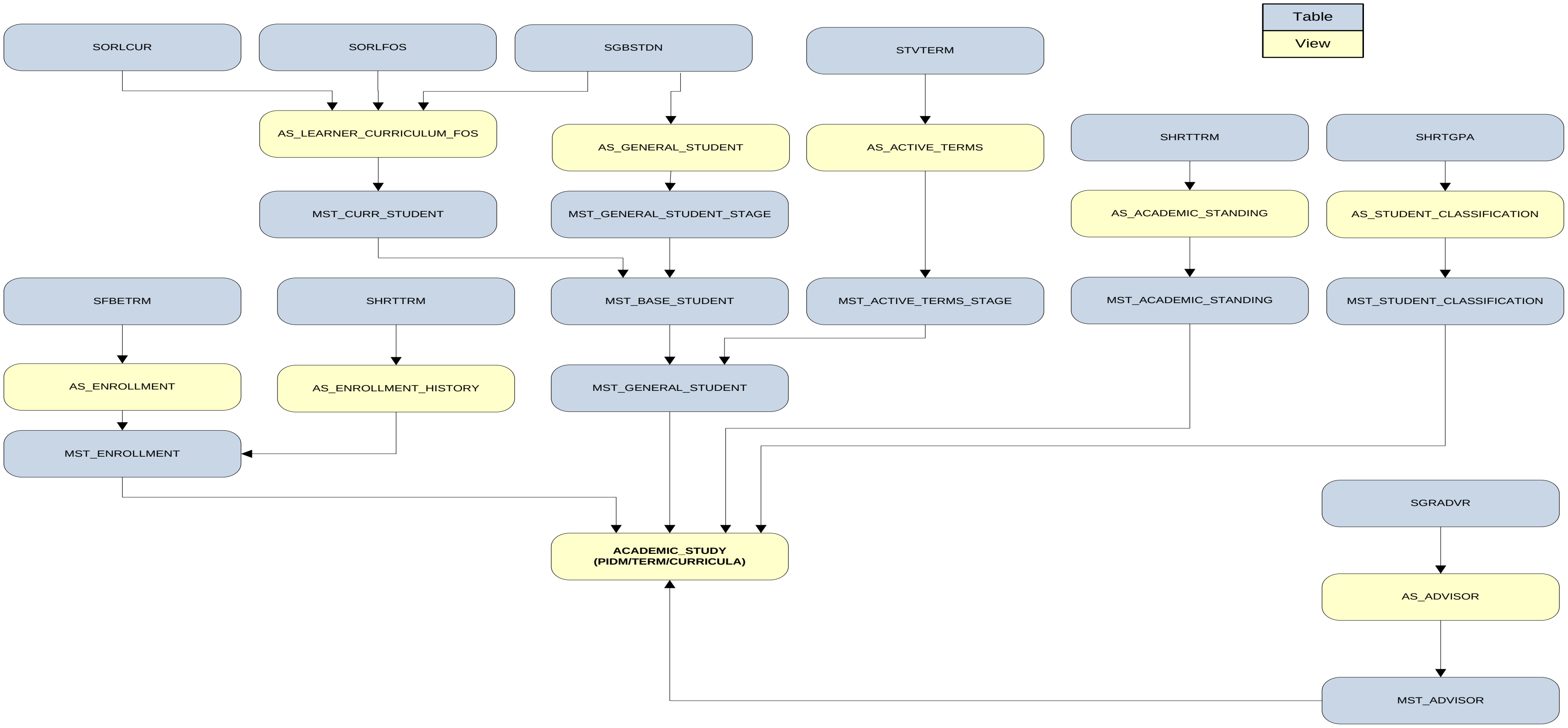
Note

These diagrams only refer to key views and key tables used within the reporting views. ■

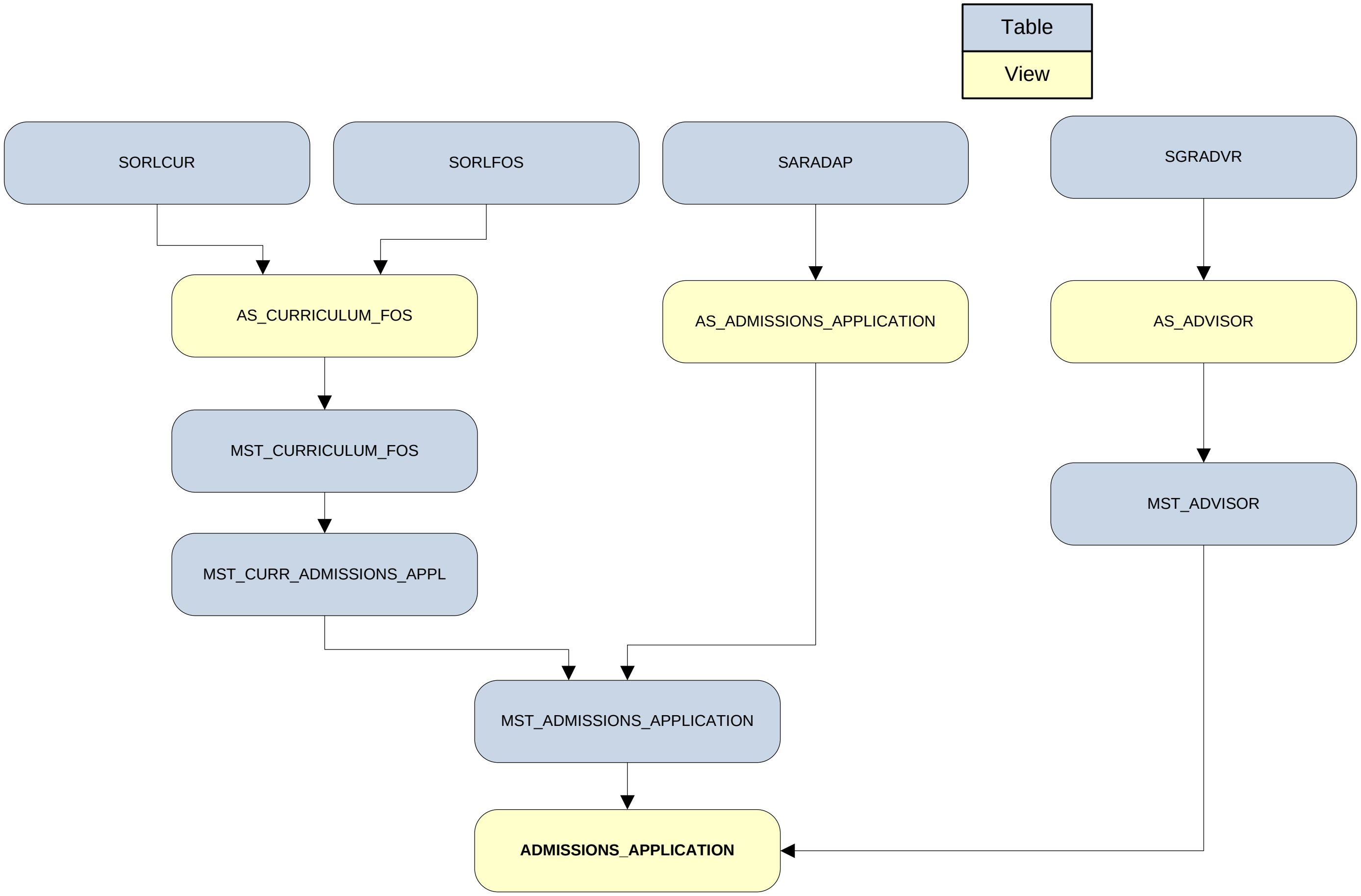
Academic Outcome and Academic Outcome Slot reporting view flow



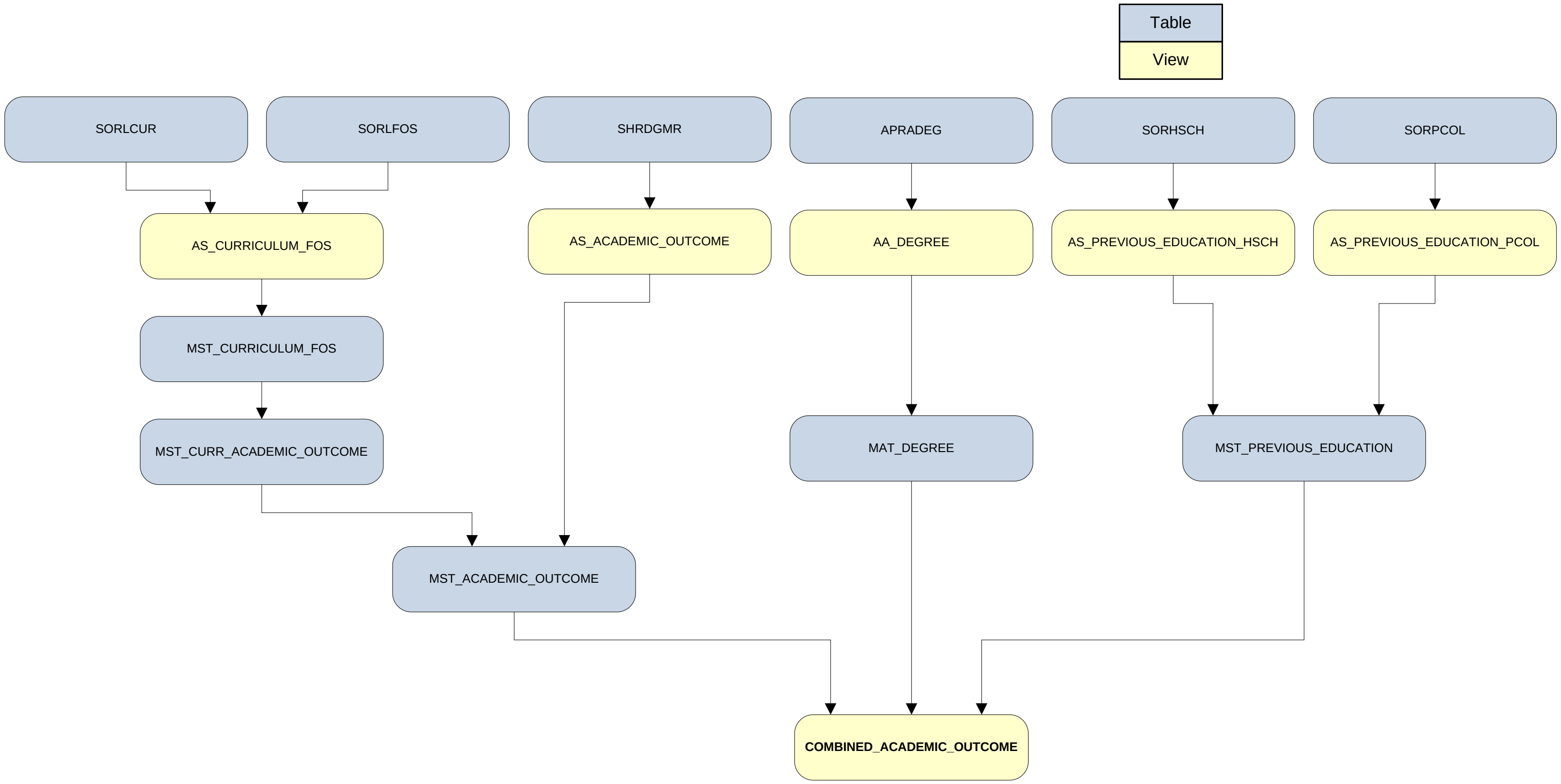
Academic Study reporting view flow



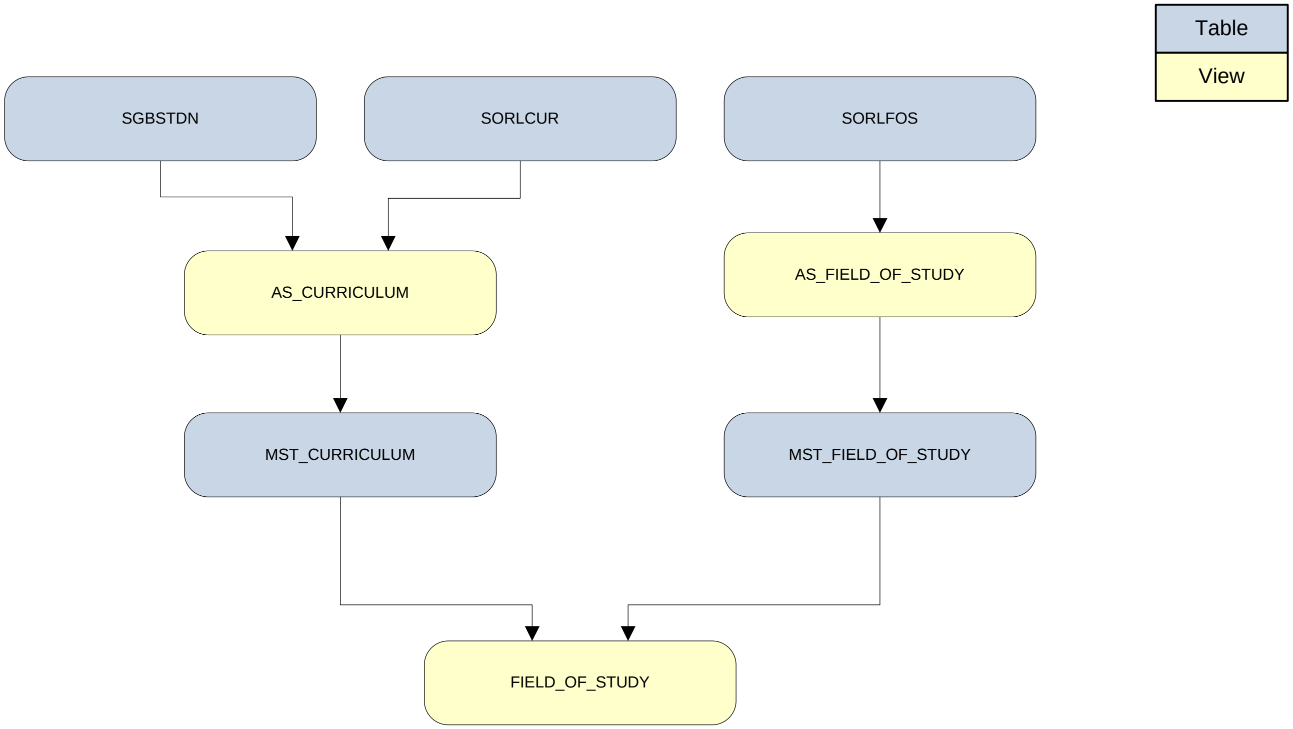
Admissions Application reporting view flow



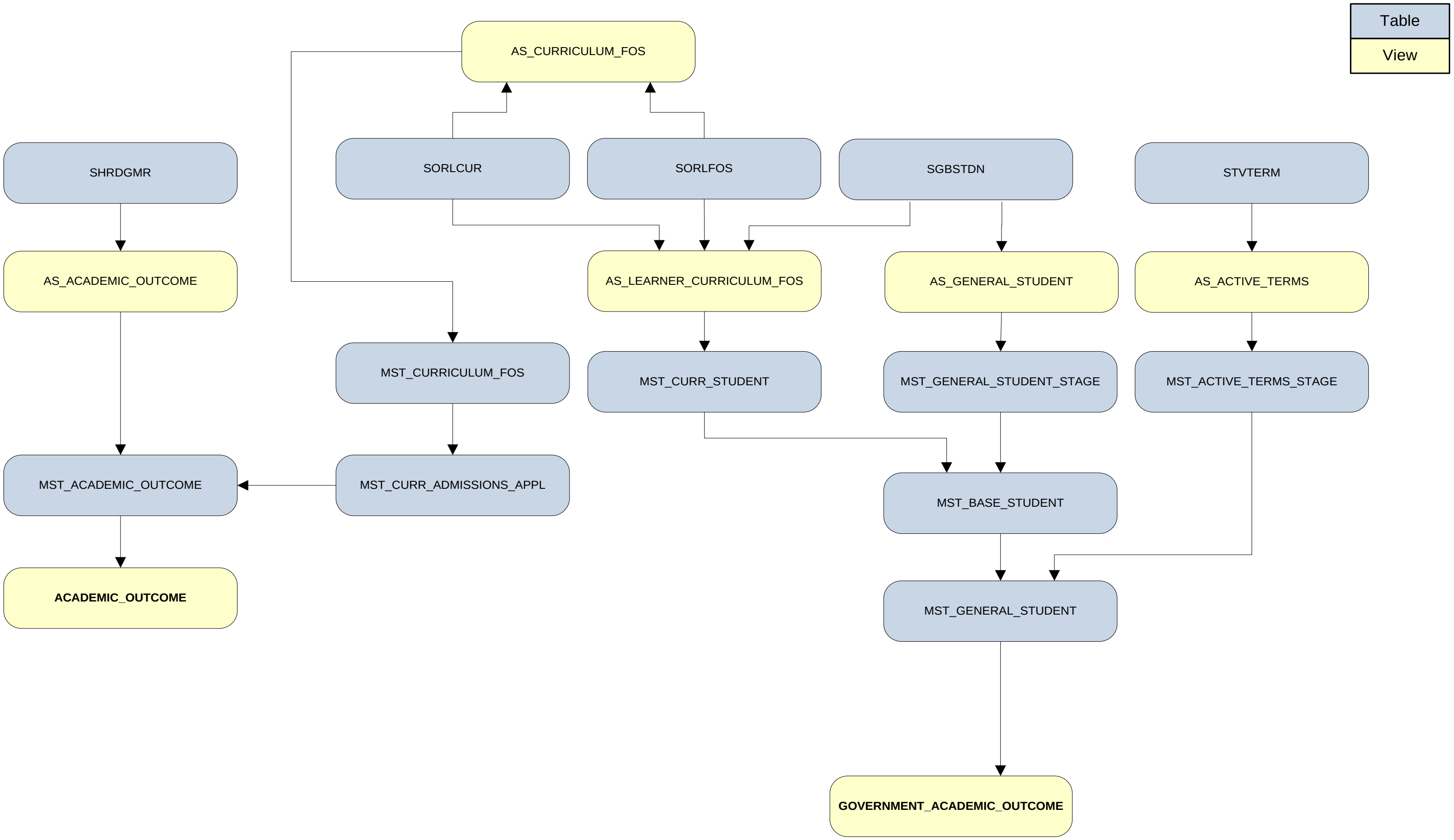
Combined Academic Outcome reporting view flow



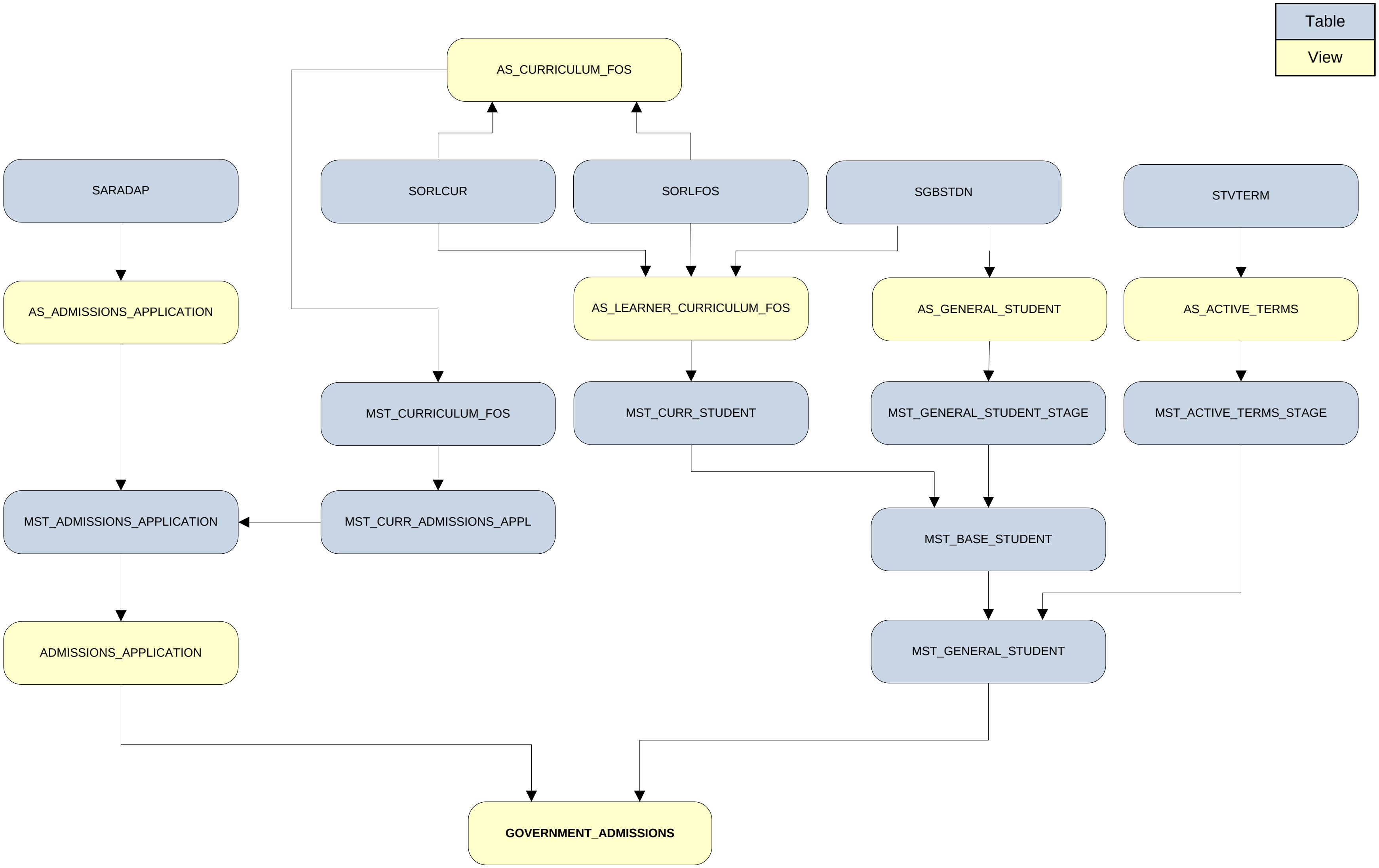
Field of Study reporting view flow



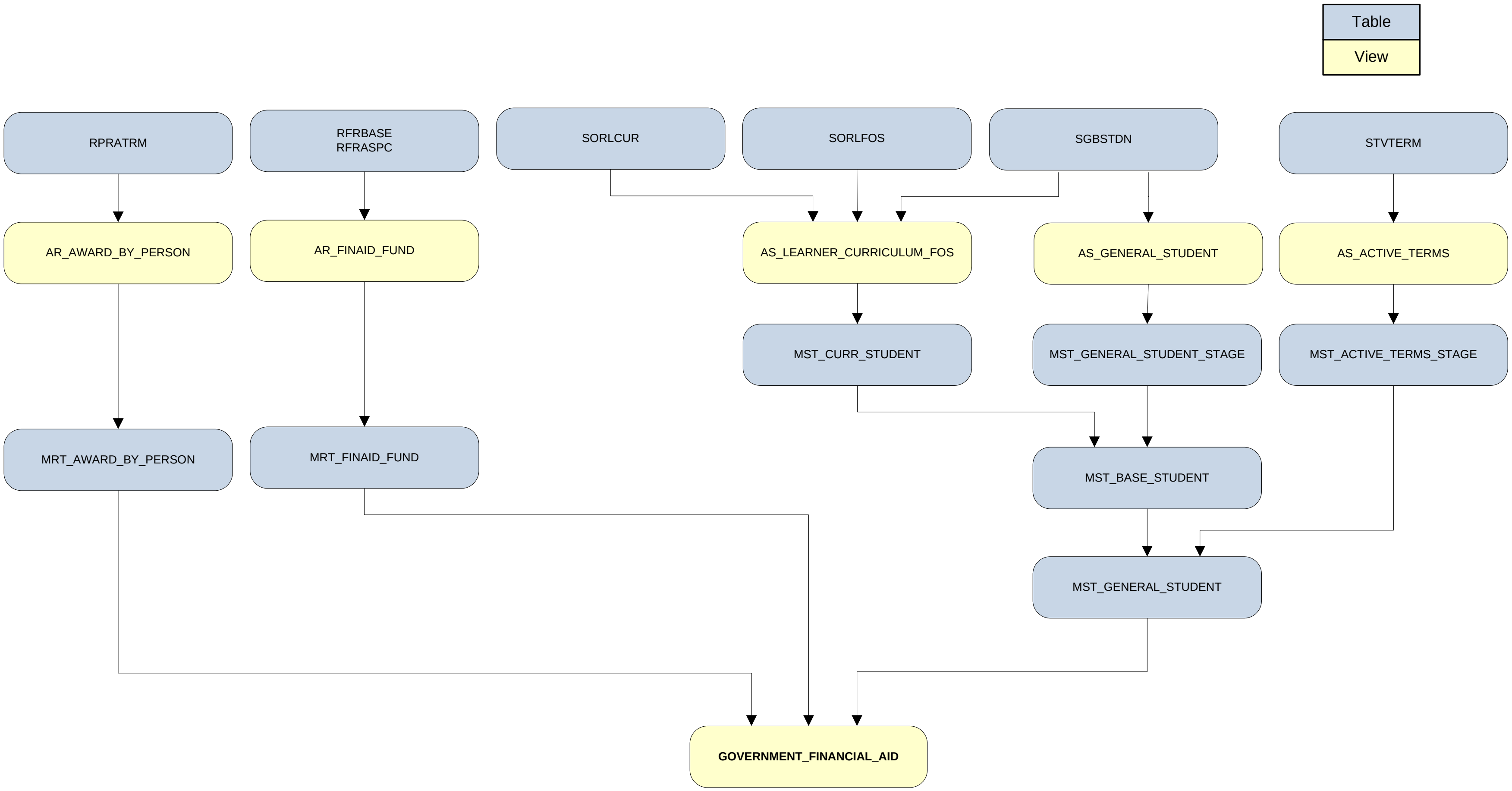
Government Academic Outcome reporting view flow



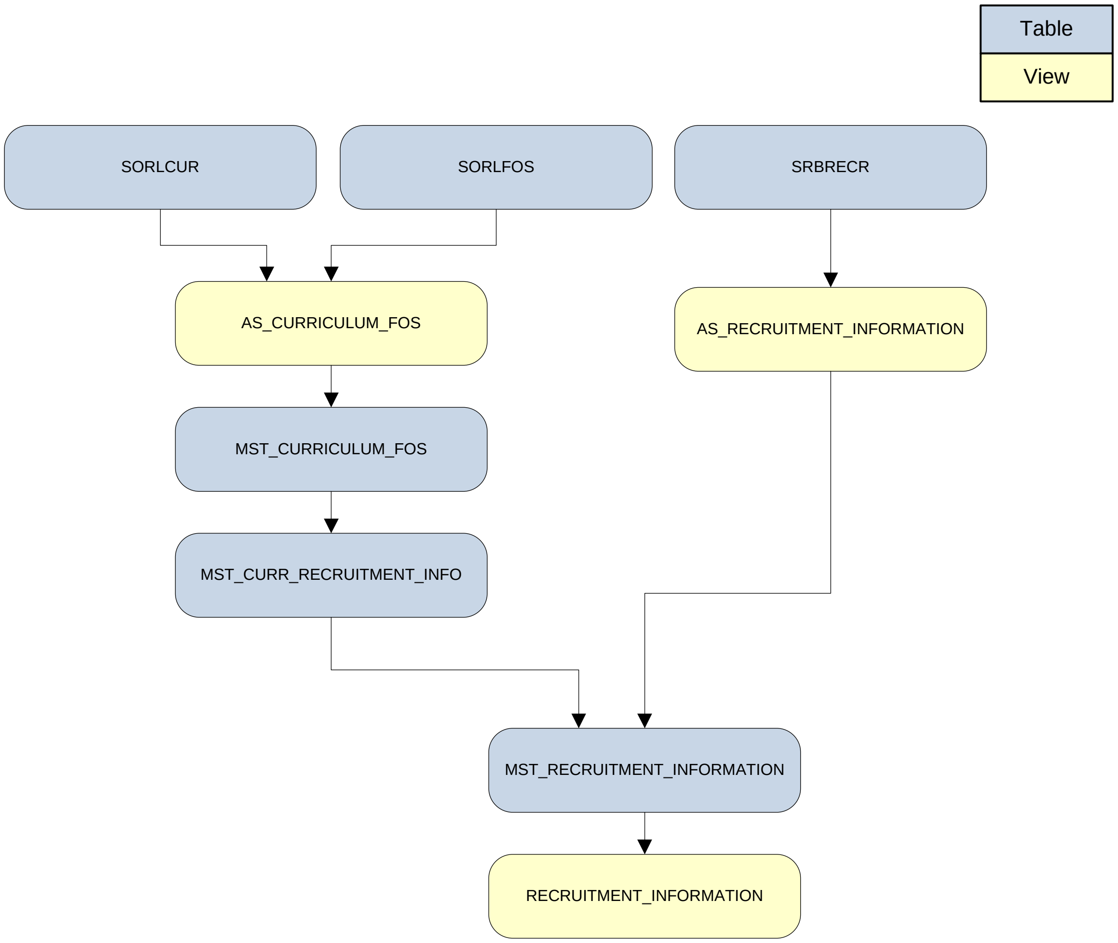
Government Admissions reporting view flow



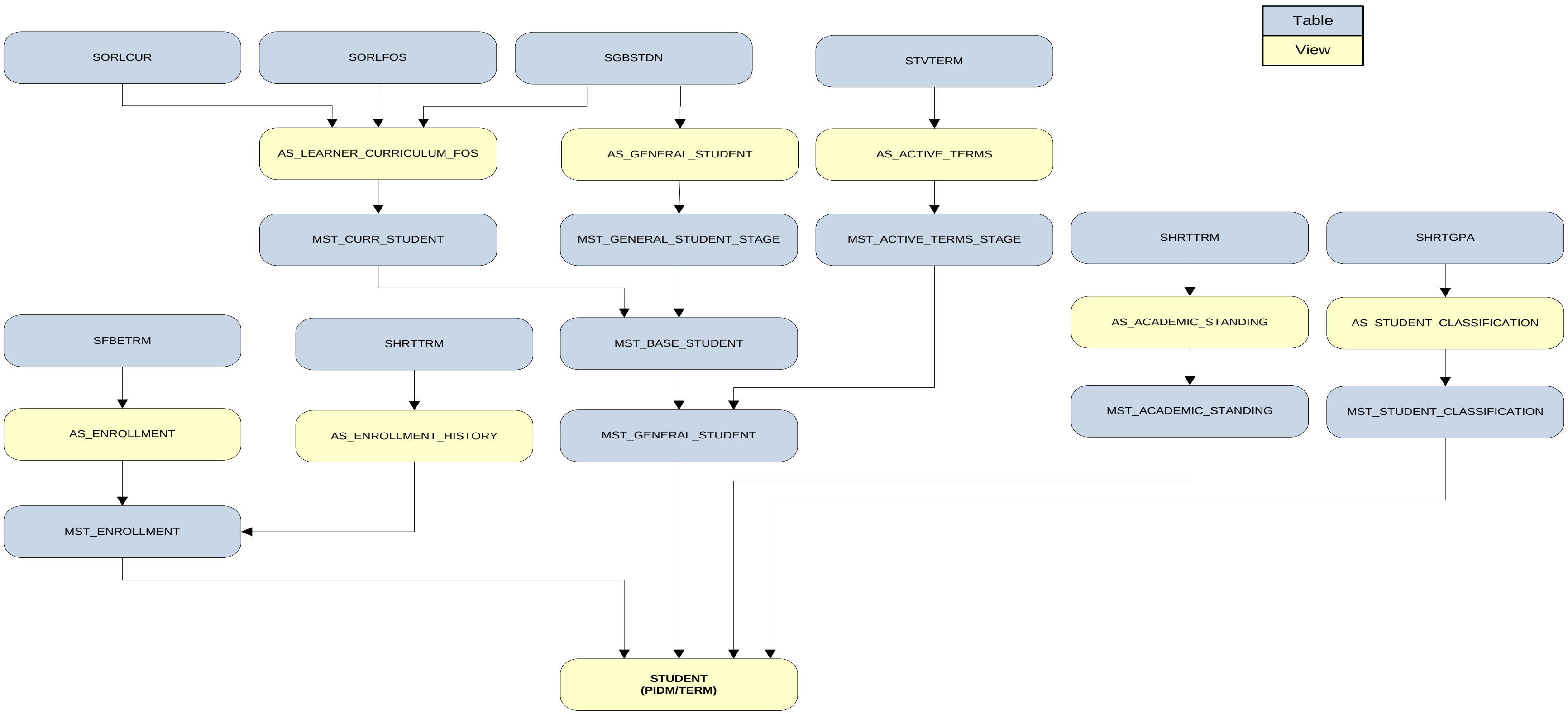
Government Financial Aid reporting view flow



Recruitment Information reporting view flow



Student reporting view flow



Composite views and meta data

The composite views gather Banner source data necessary to populate and maintain the information stored in the Banner Operational Data Store (Banner ODS). This source information then updates the information that resides in the Banner ODS database.

Note

Any institution change to a composite view impacts the Banner ODS maintenance processes. ■

The Banner ODS composite view meta data is also available as published meta data. Use the following steps to view published composite view meta data reports using the Administrative UI.

1. Select **Meta Data** from the Administrative menu.
2. Select **Banner Operational Data Store**.
3. Select the **Banner ODS Composite View Meta Data Reports** link located at the top right-hand corner of the page.
4. Select a subject area.

   The Composite View Meta Data Reports page opens listing the view name and description.
5. To display the column details associated with the selected composite view, select one of the composite views. A description of each field on the report appears below:

Field	Description
Description	Business description of the composite view target, including the key and frequency of data returned by the view.
Target Column	Name of the column in the composite view target.
Business Definition	Definition of the target column in business terms.
Database Data Type	Used for formatting purposes when writing reports.
Business Data Type	Used to store client-specific data about a given column. This field is empty by default.
Domain Value	Used to store client-specific data about a given column. This field is empty by default.

Field	Description
Source Name	Name of the source table, FUNCTION, CONSTANT, or CALCULATION.
Source Column	Name of the source column from the source table or view, if the source is a table or view; name of the PL/SQL function, if the source is `FUNCTION;` description of the constant, if the source is `CONSTANT;` description of the calculation used, if the source is `CALCULATION.`

Naming conventions

This section describes the naming conventions and standards applied to scripts and database objects used to create and maintain the BPRA solutions.

Banner EDW standards (EDWMGR/EDWSTG schemas)

Warehouse tables

Star schema tables (EDWMGR schema)

Object name:

Maximum length is 30 characters. See the table below.

1st Character	*W* - Warehouse	System Descriptor
2nd Character	*D* - Dimension *F* - Fact	Star Schema Table Type
3rd Character	*T* - Table *Z* - Snapshot Table	Object Identifier
4th Characters	_ (underscore)	
5-5th-30th Characters	Unique Descriptor	

Examples:

`WDT_TIME`, `WFT_EMPLOYEE_DETAIL`

Additional detail:

Script names are the same as the object name.

Staging tables (EDWSTG schema)

Object name:

Maximum length is 30 characters. See the table below.

1st Character	*W* - Warehouse	System Descriptor
2nd Character	*T* - Temporary	Warehouse Table Type
3rd Character	*T* - Table	Object Identifier
4th Characters	_ (underscore)	
5-5th-30th Characters	Unique Descriptor, ending in any: _INPUT _CLEAN _ERROR _WKEYS	

Examples:

`WTT_DEGREE_DETAIL_INPUT`, `WTT_ENROLLMENT_WKEYS`

Additional detail:

Script names are the same as the object name.

Sequences

Object name:

Maximum length is 30 characters. See the following table:

1st Character	*W* - Warehouse	System Descriptor
2nd Character	*D* - Dimension	Product Identifier
3rd Character	*S* - Sequence	Object Identifier
4th Characters	_ (underscore)	
5-5th-30th Characters	Unique Descriptor, ending with `_SEQ`	

Examples:

`WDS_GIFT_SEQ`, `WDS_JOB_SEQ`

Additional detail:

Sequences are created within the scripts that create the dimension tables.

Indexes and Constraints

Primary key indexes and constraints

Object name:

Maximum length is 30 characters. See the following table:

1st Character	PK_ (underscore)	Primary Key Prefix
4th-30th Characters		Table Name or Abbreviation (includes the first 4 characters, e.g., WFT_)

Examples:

PK_WFT_EMPLOYEE, PK_WFT_OPERATING_LEDGER

Foreign key constraints

Object name:

Maximum length is 30 characters. See the following table:

1st - 2nd Character	FK	Foreign Key Prefix
3rd Character	n	Where n is a one-up number
4th Characters	_ (underscore)	
5th-30th Characters		Child Table Name (omits the first 4 characters, e.g., WFT_)

Administrative standards (IA_ADMIN schema)

Administrative tables

Object name

Maximum length is 30 characters. See the table below.

1st Character	*M*- Mart	System Descriptor
2nd Character	*D* - Control Reports *G*, *T* - General Purpose	Table Purpose
3rd Character	*B* - Base *R* - Repeating *T* - Temporary *V* - Validation	Table Type
4th-7th Characters	Unique Descriptor	

Examples:

`MDBLOGH`, `MTVPARM`

Additional detail

Script names *must* follow unique 7-character naming standards. Script names are the same as the object name.

Administrative packages

Object name

Maximum length is 30 characters. See the table below.

1st Character	*M* - Mart	System Descriptor
2nd Character	*G* - General Purpose	Product Identifier
3rd Character	*K* - Package	Object Identifier
4th-7th Characters	Unique Descriptor	

Examples:

`MGKSECR`, `MGKPARM`

Additional detail

Script names *must* follow unique 7-character naming standards. Script names are the same as the object name.

Meta data tables and views

Object name

Maximum length is 30 characters. See the table below.

1st Character	*W* - Warehouse	System Descriptor
2nd Character	*M* - Meta Data	Table Purpose
3rd Character	*T* - Table or *V* -View	Object Identifier
4th Character	_ (underscore**)**	
5th-30th Characters	Unique Descriptor	

Examples:

`WMT_SOURCE`, `WMV_TARGET_OBJECT`

Additional detail

Script names are the same as the object name.

Sequences

Object name

See the table below.

1st Character	*M* - Mart	System Descriptor
2nd Character	*G* - General Purpose	Product Identifier
3rd Character	*S* - Sequence	Object Identifier
4th-7th Characters	Unique Descriptor	

Examples:

MGSHOST, MGSPARM, MGSPIDM, MGSSDAX

Additional detail

Script names *must* follow unique 7-character naming standards. Script names are the same as the object name.

3 Enhanced Warehouse Architecture

The Banner Enterprise Data Warehouse (EDW) includes an enhanced warehouse architecture for the Advancement data structures delivered with the warehouse.

Note

The information in this chapter applies only to the Advancement-related data structures in the warehouse. If your warehouse includes other data structures, for example, student or financial aid, the "Architecture" chapter of the *Banner Enterprise Data Warehouse Handbook* details the architecture components related to those data structures.

Enhanced warehouse architecture components

The enhanced warehouse architecture includes several components illustrated in the following figure.

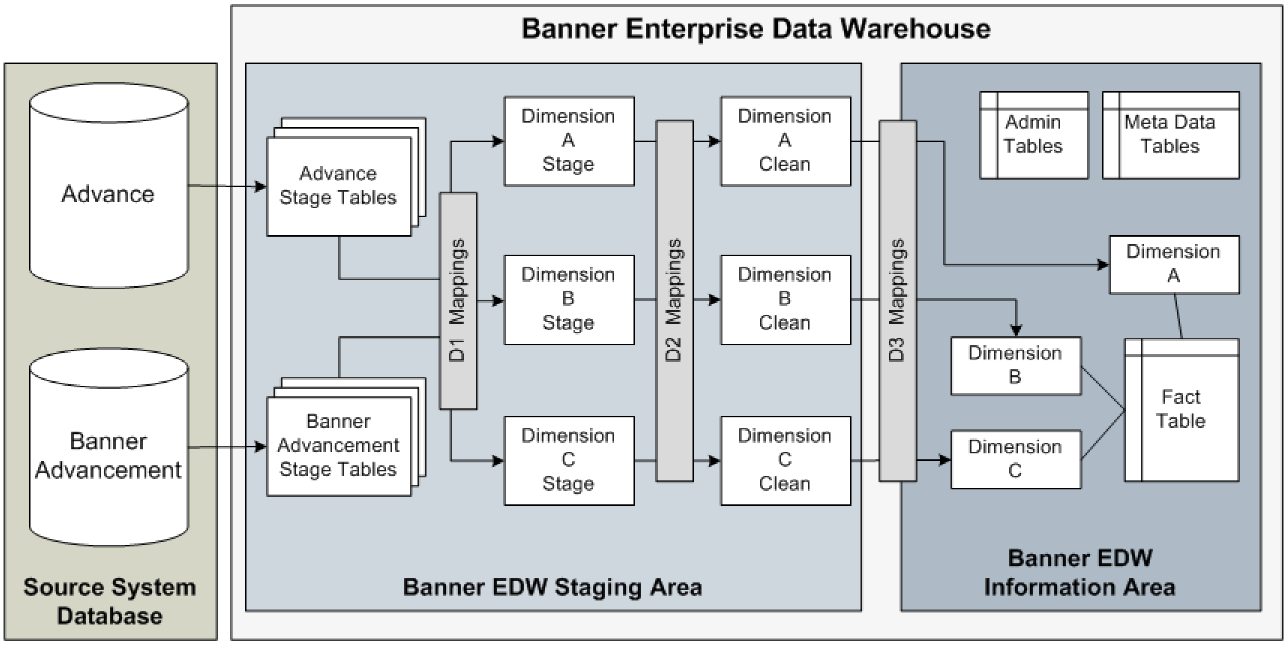


Figure 17: Enhanced Warehouse Architecture

At the highest level the components are grouped as follows:

- Source System database - Advance or Banner Advancement source databases
- Warehouse Staging Area - area of the EDW that houses the source staged tables, dimension staged tables, and cleansing staged tables
- Warehouse Information Area - area of the EDW that houses the actual fact and dimension tables from which the reports and scorecards draw information

Source system database

The starting point for any performance or reporting analysis solution is your source system data. The information stored in the source transactional database is ultimately the information that you want to analyze.

The Advancement Analytics for Cognos product is specifically designed to accept information from the Advance and Banner Advancement products. The product uses an open design and can accept information from other sources.

Warehouse staging area

One benefit of a data warehouse is that it gives you a multi-dimensional view of your data. The warehouse is designed to manage complex queries associated with institutional reporting without the overhead associated with a transactional system.

One challenge of working with a data warehouse is synchronizing data changes between the source system and the warehouse. Current data is key to accurate reporting. Populating the warehouse with information from the source system is a multi-layered process. The data from the source system need to be flattened out and aligned into the appropriate dimensions in the warehouse. In addition, before some source data are loaded into the warehouse it may need to be cleaned up to make the output suitable for consistent reports. For example, the source system may include null values that need to be changed to identify the empty fields.

The warehouse staging area exists to help meet data synchronization challenges and to enhance the process of moving data from the source system to the warehouse. The following figure illustrates the progression of information from the source system through the warehouse staging area.

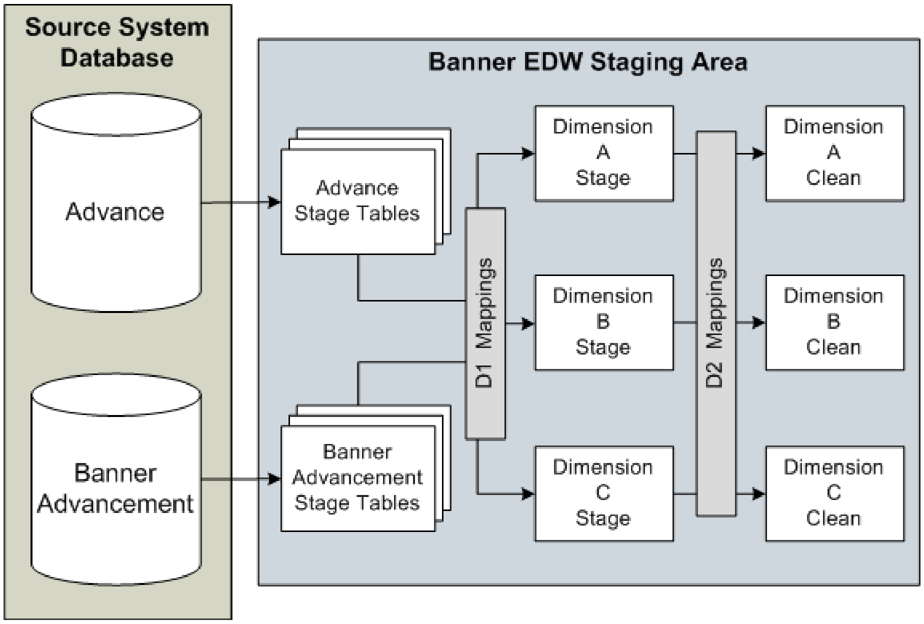


Figure 18: Warehouse staging area

Source stage tables

Information from source database tables is replicated in the warehouse source stage tables using Oracle Streams Replication and Change Data Capture processing. The source stage tables are basically copies of tables from the source database. Replicating the tables in this way makes it easier to use Oracle Streams to maintain and update exact copies of source data tables in the warehouse staging area.

The source stage tables become the starting point of information for the final data warehouse tables. Information in the stage tables will be cleansed and associated with dimensions and facts before being loaded into the final dimension and fact tables of the warehouse.

Dimension stage tables

The dimension stage tables are warehouse tables used to reorganize source stage table information into business-related information areas or dimensions. These dimension tables are a part of the staging area and act as intermediary tables housing information on its way to the final dimension and fact tables.

Business logic mappings (D1) process information from the source stage tables to load the dimension stage tables. There is one business logic mapping for each dimension table.

Dimension clean tables

The EDW cleansing process provides data translation functions that can define data translation values and replace values for null data. During the EDW cleansing process these cleansing table functions define rules that run against individual columns in the dimension stage tables transforming the data and loading them into the dimension clean tables. The dimension clean tables are warehouse tables that house cleansed versions of the data in the dimension stage tables. You manage these cleansing rules through the EDW Administrative User Interface (UI). Refer to the “Administrative User Interface” chapter of this Handbook for more information about using the Administrative UI to maintain EDW cleansing.

Warehouse information area

The warehouse information area is the portion of the EDW that houses the actual fact and dimension tables from which the reports and scorecards draw information. The following figure illustrates the components of the warehouse information area.

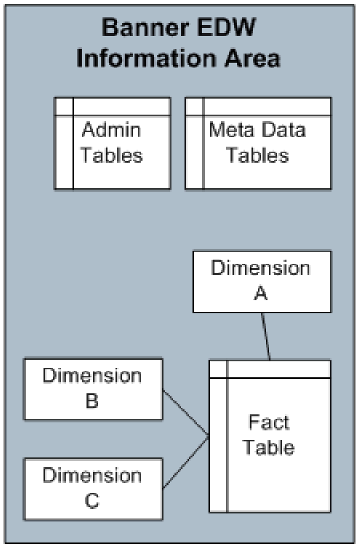


Figure 19: Warehouse information area

Fact tables

Fact tables store the numerical performance measurements, such as amounts or counts, tracked by your institution. Hard Credit Amount is an example of a measure that is stored in the WFT_GIFT fact table. The warehouse includes a fact table for each numerical measure. The fact table names begin with `WFT_`.

Dimension tables

Dimension tables store the descriptive attributes that define how you want to slice or look at the measures in a fact table. Campaign is an example of a dimension that is stored in the WDT_CAMPAIGN table. The dimension table names begin with `WDT_`.

Admin Tables

Admin tables store the information related to ETL load jobs, cleansing jobs, and warehouse security. The tables support the functions of the Administrative User Interface.

Meta data tables

The meta data tables store the information that defines the warehouse meta data.

User-defined fields

EDW provides five additional user-defined fields on every dimension and fact table which allow you to extend the data in the warehouse. Including these user fields within the product tables and their related mappings means you need to make minimal changes to bring new data into the warehouse. The dimension user-defined fields are named USER_ATTRIBUTE_01-05, and the fact and aggregate user-defined fields are named USER_MEASURE_01-05. Follow the guidelines in this section to use the user-defined fields.

General guidelines

Considerations should be made when deciding what data to add to the user-defined fields. Use the following questions as guidelines when determining what data to add and where to add it.

What source data to add?

First, you must evaluate the sources of the new data and whether the sources are available in the source system and/or warehouse – these guidelines assume the data already exist in the staged source tables and/or warehouse and are available for you to use. If not, additional steps will be needed to stage the data from the source system and/or add it to the warehouse.

How will the new data be used?

Next, consider how the data will be used. Will it be used as an attribute or a measure? This will tell you whether it should be added to a dimension (attribute) or fact (measure) table.

If you are adding a new attribute to a dimension table shared by multiple organizations at the institution, consider whether or not those organizations will be interested in the same content. If so, you should gain input from all offices regarding the details of the new content so that you can create common attributes. If multiple organizations are not interested in the same content, consider adding the new data to a dimension table not shared by other organizations or to two different user attribute fields. This will enable user attribute columns to be optimally distributed among the storage structures while minimizing the need for customization.

Where to add data?

When trying to determine which dimension or fact table to add the user attribute to, look for dimensions or facts that contain similarly sourced data. Looking through the meta data using the Administrative UI can help you see where similar data already exists.

Is the data at same granularity as other data in star?

You must also consider whether the new data item is captured at the same level of granularity as the other data in the star. These guidelines assume the data is at the same level of granularity. If data is at a different level of granularity, additional changes will be needed within the ETL, table definitions, report structures, or the query retrieving the data.

Conformed dimensions

The EDW uses conformed dimensions, which are dimensions shared among stars. When modifying a dimension that is shared by many stars, you need to consider the following

Will each star using a conformed dimension use the same user-defined attributes?

- If "Yes" - you must update each star's table function and extract mapping to select the new user-defined field.
- If "No - (meaning you want to add the user-defined field to one star but not another) there is no issue, the other stars do not need to be modified and would continue to populate the field with a null value.

Will each star using a conformed dimension need different user-defined attributes for different stars?

Since dimensions are shared between stars, each user-defined field must have one unique source and thus one unique cleansing rule across all stars that share that dimension. This means that if you wish to add two *different* user-defined attributes to the same dimension table but for different stars, you must use two different user attribute fields. For example, if you want to add hair color and legislative district to the WDT_DEMOGRAPHIC dimension for different stars, you must use USER_ATTRIBUTE_01 for one attribute and USER_ATTRIBUTE_02 for the other.

Steps to add user-defined attribute and measure values to the warehouse

Step 1 Update the extract logic that selects values from the source tables to populate the warehouse tables.

The steps needed to add user-defined attributes and measure values to the warehouse depend on the type of star you want to update. Refer to the steps in the following sections to select the appropriate source value and populate the appropriate warehouse target table and column.

- "For Advancement operational stars" on page 3-7
- "For aggregate stars" on page 3-7
- "For frozen stars" on page 3-8

For Advancement operational stars

To add values to the user-defined attribute or measure fields, you need to update the extract OWB mappings and if it is a dimension attribute, the corresponding cleansing rule.

1. Locate your mapping within the appropriate OWB Project – either APM_GENERAL or APM_ADVANCEMENT. Source specific mappings end with the suffix 'A' for Advance or 'B' for Banner Advancement.

2. If you are adding an attribute, update the associated D1 mapping within the Project>Oracle>EDWSTG module, for example, <dimension>_D1A or <dimension>_D1B, to map the new value to the dimension's user attribute.

3. If you are adding a measure, update the associated F1 mapping within the Project>Oracle>EDWMGR module, for example, <fact>_F1A or <fact>_F1B, to map the new value to the fact table's user measure.

4. Save the changes.

5. Deploy the mapping.

For aggregate stars

Use the following steps to add values to the user-defined attribute or measure fields of an aggregate star.

1. Locate the aggregate mapping within the appropriate OWB Project.
   - There are no source specific aggregate mappings.
   - If changing an Advancement aggregate, the aggregate mapping naming convention is <aggregate>_A1

2. Map the new value to the aggregate star's user measure. It is likely that the user measure for an aggregate will be sourced from an updated operational star's user measure.

3. Save the changes.

4. Deploy the mapping.

For frozen stars

Follow these guidelines when updating frozen stars:

- The corresponding operational or aggregate star which sources the frozen one must be modified to include the user measure.
- Once the corresponding operational or aggregate star is updated, no changes are required to populate the frozen star, links between the frozen star's user measure fields and the corresponding star's user measure fields are already in place.

Step 2 Link cleansing rules to new user-defined dimension attribute.

Note

This step only applies if you have added a user-defined dimension attribute; it does not apply to user-defined measures.

Perform the following steps within the Administrative UI.

1. Select **Options>Set Up and Maintain Cleansing Processes**.
   - If the user-defined value you have added already can be linked to an existing Cleansing rule, skip to the next step.
   - Else, go to **Set Up and Maintain Cleansing Rules** and create a new cleansing rule with appropriate source query. See the "Set Up Cleansing" chapter for more details.

2. Select **Options>Set Up and Maintain Cleansing Processes>Set Up and Maintain Cleansing Data Elements**.

   2.1. Select the dimension table that will be updated with your user-defined attribute and select **Search**.

   2.2. Select the USER_ATTRIBUTE column 01-05 you modified in Step 1.

   2.3. Within the Update a Cleansing Data Element screen, update the **Rule Name** field with the appropriate cleansing rule.

   2.4. Click **Save**.

Your new user-defined attribute value will now be cleansed by the appropriate cleansing rule.

Step 3 Update meta data to reflect the change (Optional)

If you use the published meta data to document source to target relationships, refer to the "Meta Data (Banner ODS and Banner EDW)" section of the "Administrative User Interface" chapter to add corresponding data for the new column.

1. Create a new target column for the updated dimension or fact table, naming the column as it is in the database, that is, USER_ATTRIBUTE_01-05 or USER_MEASURE_01-05.

2. Give the column a new Business Name, Definition, and other information to match the new use for the column.

3. Save the changes.

4. From the same target column page, select **Add Local Mapping** at the bottom.

5. Select and save the appropriate source details.

6. Publish meta data.

The new column will be available.

Step 4 Reload data via the Administrative UI to populate the new field

1. Select the appropriate jobs to reload the data that will access the new field.

2. If the new field is in an operational star, you must also reload the related aggregate stars.

Note

Population processes for snapshot and frozen stars will only impact data populated in the related star schema after the change and related Cleansing setup is completed. You will not be able to reload data to existing snapshots or frozen concepts except for those milestone events related to 'Final State' data storage in the source system (such as 'final' for academic period or fiscal period.) Reloading milestone events (not 'final') will invalidate the time perspective of the data. ■

3. Confirm that the correct data is loaded into the modified dimension or fact.

Step 5 Update delivered Cognos models to expose and appropriately label the user-defined attribute or measure (Optional)

Note

You only need to perform this step if you are using the delivered Cognos content. ■

The delivered Cognos Framework Manager Models have the following three layers.

- Database View
  - Reference to database objects
  - Table names and select columns given business names
  - User attribute and measure columns already present in database layer but do not have business names
  - Fact tables obtain “Measure” suffix
  - Some concatenated unique keys added
  - Some formatting and aggregation defined
- Business View
  - Business-centric grouping of objects
  - Business-centric table aliases created
  - Business-centric relationships/joins created
  - Alternative aggregation query items defined, for example, averages, headcounts, percents, counts
- Presentation View
  - End user experience/layer that is published to the end-users
  - Commonly used items grouped together
  - Folders used to remove clutter
  - Commonly used filters, calculations created
  - Customizable parameter driven items created

Perform the following steps to update the Cognos packages:

1. Update the Database View layer for the table/column you modified.

2. Rename the USER_ATTRIBUTE/MEASURE_01-05 and their corresponding short description (SD) and long description (LD) columns in the table you modified to have business names.

3. Update the Business View layer to use these new names.

4. Update the Presentation View layer to include the new code and description fields in the appropriate Query Subject.

5. Update the appropriate Package Definitions to select the new fields to be published.

   You will then be able to access the new fields via the newly published packages and use them through the various Cognos Studios such as Report Studio and Query Studio.

6. Update the Cube transformer model with the new columns and republish it. (Optional)

Data replication and ETL process

Replication the data between your source database and the warehouse database is key to performing accurate data analysis and creating current reports. Data replication is accomplished using the Extract Transform and Load (ETL) process.

ETL process flow

The ETL process includes three major areas of processing that are illustrated in the following picture.

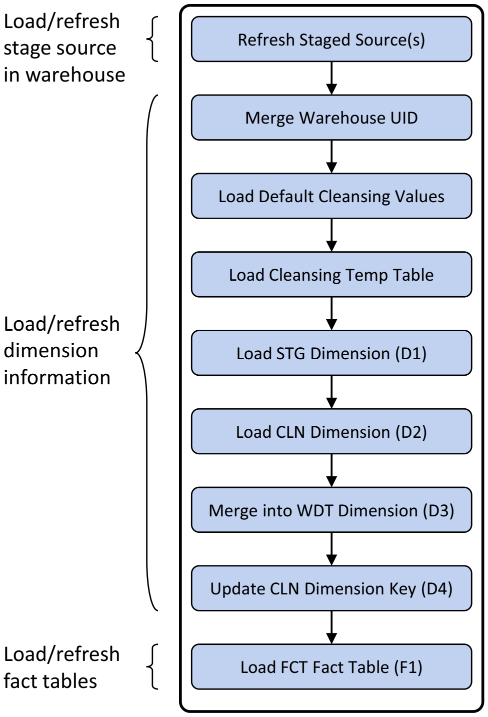

Load/refresh stage source in warehouse
Refresh Staged Source(s)
Merge Warehouse UID
Load Default Cleansing Values
Load Cleansing Temp Table
Load/refresh dimension information
Load STG Dimension (D1)
Load CLN Dimension (D2)
Merge into WDT Dimension (D3)
Update CLN Dimension Key (D4)
Load/refresh fact tables
Load FCT Fact Table (F1)

Figure 20: ETL process flow

Load/refresh stage source in warehouse

The warehouse staging area is populated with data from the source database using Oracle Streams Replication and Change Data Capture processing. Any insert, update, or delete actions performed on the source database tables are also performed on the warehouse stage tables to synchronize the two.

The following picture shows the components used to synchronize data between the source database and the staging area of the warehouse.

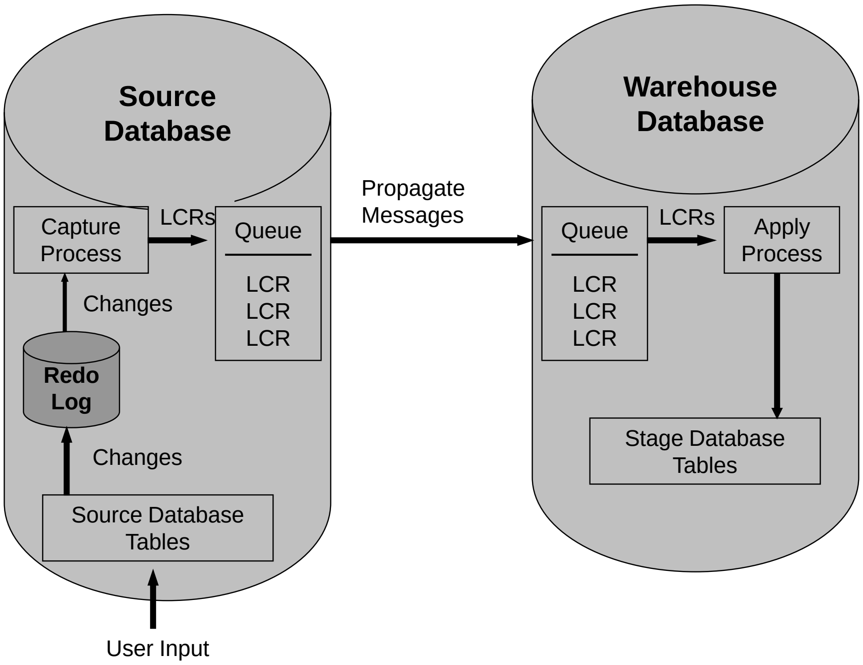


Figure 21: Data replication between source database and warehouse database

The replication process uses the following components of Oracle Streams.

Oracle Streams Component	What it does...
Capture Process	The Streams capture process mines changes from the redo log on the source database whenever possible and from the archived logs if it falls behind the generated redo logs. Changes in the redo log that match specified rules are converted into messages called logical change records (LCRs), which are placed in a queue associated with the capture process.

Oracle Streams Component	What it does...
Propagation Schedule	The Streams propagation schedule moves the messages from the source database queue to a queue on the warehouse. Each message remains in the source queue until the destination confirms that it has received it. This confirmation guarantees messages are never lost during the propagation stage.
Apply Process	The Streams apply process in the warehouse removes the messages from the queue and applies them directly to the source stage tables in the warehouse. Any errors encountered while applying the messages are placed into an error queue. The messages in the error queue can be reprocessed once any issues have been resolved.

During the installation process, the system creates all the components required to set up the Oracle Streams framework. This includes setting up the Streams queues, propagation schedule, capture and apply processes, and populating the staging tables in the warehouse with data from the source database tables.

Refer to the following Oracle documents for more information about maintaining and troubleshooting the Oracle Streams environment in the EDW.

- “Oracle Streams Concepts and Administration Guide”
- “Oracle Streams Replication Administrator's Guide”
- “Streams Complete Reference FAQ” (MetaLink Document ID: 752871.1)

DDL handler

A DDL handler is assigned to each Streams apply handler to replicate DDL statements from the source database to the stage source tables in the warehouse. DDL statements run against a source table are replicated to the warehouse unless a statement includes table dependencies. This means DDL statements executed on the source database that create, alter, or drop columns, non-foreign key constraints, and indexes are replicated to the warehouse; however, using the DDL handler allows the replication process to ignore the same types of statements for table triggers and foreign keys. These changes will not be replicated in the warehouse.

The user who executes a DDL statement on the source database must also exist in the warehouse for the change to be replicated successfully in the warehouse. When a DDL command is executed on the source database and the object schema is not specified, then the DDL command will only be applied successfully on the warehouse if the user who executed the command also exists in the warehouse.

⚠ **Warning**

Be aware that source objects with system-generated names will have different names in the warehouse. This means that DDL statements involving these objects will not be replicated successfully, and will result in DBA_APPLY_ERROR records being created. The majority of the system-generated names in the source system that may be affected are NOT NULL constraints on table columns. ■

Load/refresh dimension information

Dimension information goes through multiple steps during the load process to move it from the staged source area of the warehouse to the final warehouse dimension tables. The following dimension loading components process the dimensional attributes used by all of the fact tables.

- Business Logic Mappings - process information from the staging source tables to load the dimension stage tables. There is one business logic mapping for each dimension table. These mappings are source-specific to reflect the unique tables and objects referenced from each system.
- Cleansing Table Functions - include a cleansing process that pushes data from the dimension stage tables to the dimension clean tables. The cleansing process is a template that you can define for the dimensions your institution needs to process. The cleansing process is where you define data translations and replacement values for null data.
- EDW Staging Schema - processes data from the dimension clean tables into the final warehouse dimension tables.

OWB mappings

Oracle Warehouse Builder (OWB) mappings, which are PL/SQL scripts, define the relationship of data from the source stage tables to the final dimension and fact tables in the warehouse. The Extract, Transform, and Load processes (ETL) built using OWB are the mappings that populate the warehouse.

The OWB mappings are run to initially populate the warehouse and to incrementally refresh the warehouse tables. When run, dimensional information is refreshed and the fact tables are repopulated.

The OWB user interface contains graphical editors that enable you to design a complete logical model of your warehouse. The OWB helps you plan how to extract data from a variety of sources, transform the data, and configure the data for loading into the warehouse. The OWB code generator lets you deploy and populate the warehouse without manual coding, and integrates with the Oracle database and query tools.

Several related mappings take information through incremental steps within the staging area to populate the final warehouse tables. The following picture illustrates the mappings used to load demographic data from the source stage tables to the warehouse dimension table.

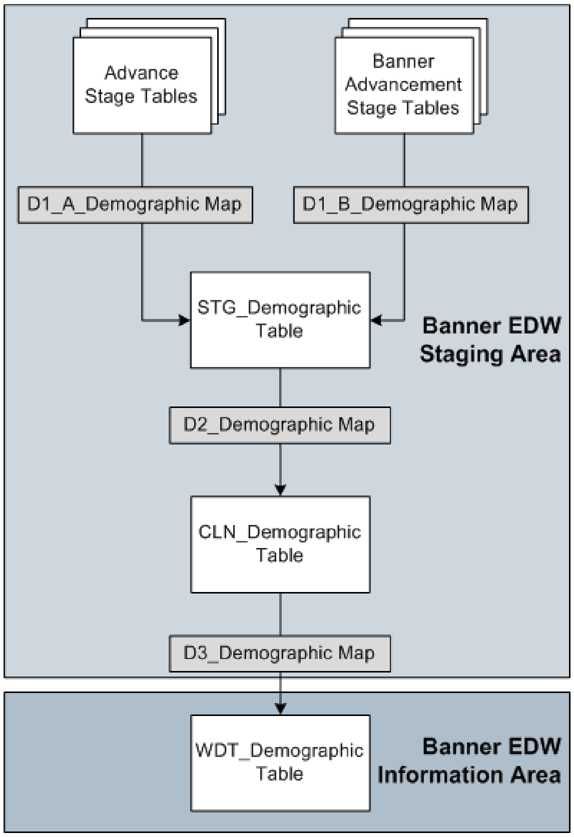


Figure 22: Demographic dimension mappings

D1 mappings

D1 mappings load dimensionally structured data from the staged source into the dimension stage tables (STG version of dimension tables.) The number of records inserted into a dimension stage table depends on the cardinality of information in the dimension relative to the fact tables by which it will be referenced. Each record will contain a LKP column which uniquely identifies the associated information from the staged source.

The warehouse includes source-specific D1 mappings to load information from both the Advance and Banner Advancement staged sources into the dimension stage tables. For each dimension stage table there are two D1 mappings that could load that table. A D1_A mapping to load Advance source staged information and a D1_B mapping to load Banner Advancement source staged information.

D2 mappings

D2 mappings take the records loaded into the dimension stage tables in the D1 mapping and process them through a table function which replaces NULL values where necessary and applies optional cleansing transformations. The resulting records are loaded into the dimension clean tables (CLN version of dimension tables.)

D3 mappings

D3 mappings merge the unique combinations of key attributes from the dimension clean tables into the actual warehouse dimension tables. As new records are populated within the dimension, the accompanying trigger will generate the associated surrogate key.

D4 mappings

D4 mappings update each dimension clean table with its associated surrogate key from the dimension table. This enables the use of the LKP column, which is used as a lookup to retrieve a dimension’s surrogate key.

Load/refresh fact tables

The fact loading components include one fact mapping for each fact table in the EDW. Each fact mapping uses a driving fact query from the source system and the dimension clean tables as look up tables to load data into the fact tables in the EDW.

The fact and dimension tables are also the sources of data to populate the aggregate fact tables and group dimensions. These objects are integral components that enhance query performance and flexibility.

F1 mappings

F1 mappings truncate and load fact tables with metrics from source tables and their associated dimensional keys.

Warehouse schemas

The following schemas exist in the warehouse.

Schema	Owns
IA_ADMIN	• Meta data tables • Any objects used or associated with the Administrative User Interface like the parameter table, the data display rules table, and the security tables
ODSSTG	Oracle Streams objects This user is created in the source and warehouse systems.
EDWSTG	Staging dimension tables (STG) and clean dimension tables (CLN)
EDWMGR	Final dimension and fact tables

Administrative User Interface

The Administrative User Interface (UI) is a Web-based tool that uses Ellucian's Banner Web Tailor application. You can use the Administrative UI to set up and maintain the warehouse, including initiating and monitoring ETL processes. Administrative functions include:

- Preferences and Security - Use to manage security, set global preferences, and set up user accounts.
- Options - Use to control the processes to extract and load data into warehouse, schedule a process, view control reports, view and/or remove scheduled processes, and maintain information about saving (freezing) data.
- Meta Data - Use to view and manage the meta data supporting the systems.
- New Banner Web Tailor Administration - Use to customize a Web menu, procedure, graphic element, set of information text, or a set of menu items. You can also update user roles, customize a Web module, Web rules, or Banner Web Tailor parameters; customize a login return location; and customize Banner Web Tailor overrides or global user interface settings.

Refer to the *Administrative User Interface Online Help* system for more information about using it.

Cleansing

Data cleansing is the process of ensuring that dimensional attributes are populated with appropriate data in the warehouse. Null values associated with required attributes are populated with data, and transformations of data are optionally performed based on user-defined rules. Using the Administrative UI, the data warehouse administrator can set up cleansing rules specific for your institution.

You can use cleansing to accomplish the following activities:

- Translate a code value in the source system to a new value in the warehouse
- Change a source system description value to a new value in the warehouse
- Group a range of source system code values into one warehouse value
- Translate multiple source system values into one warehouse value and description
- Associate an effective date with code descriptions that can change over time

Multiple source databases

The enhanced warehouse architecture supports stage tables from different source databases. The only requirement to load information from multiple sources into the warehouse stage tables is that the schema and table names in the source databases must be unique.

Note

Because the schema and table names in the source databases must be unique, you cannot load information from two different Banner databases into the warehouse.

Source Alias

The Source Alias (`source_alias`) uniquely identifies each source database. You specify the `source_alias` during the installation or upgrade process. The `source_alias` is then used to create a parameter in Web Tailor, which associates each `source_alias` to a database link owned by ODSSTG. This approach allows the database link to the source to be changed while minimizing the disruption to the existing warehouse functionality.

The Source Alias is used as a prefix when naming the various Streams components. The prefix identifies the source database and the suffix identifies the Streams component. For example, the Streams component BANNER$APP is associated with the source alias of BANNER, and is an apply process.

The following table lists the database location and suffix for each Streams component. The Source Alias is added to the beginning of each Name Suffix to uniquely identify the Streams component.

Streams Component	Database location	Name Suffix
Capture	Source	$CAP
Capture queue	Source	$CAPQ
Capture queue table	Source	$CAPQT
Propagation	Source	$PROP
Apply queue	EDW	$APPQ
Apply queue table	EDW	$APPQT
Apply	EDW	$APP

Add a source database

Use the following steps to add subsequent source databases after an initial source database has been configured.

1. Run the source install or upgrade steps on the source database. Refer to the *Banner EDW Installation Guide* or *Upgrade Guide* for the install or upgrade steps.

   This creates the ODSSTG Streams administrative user with the necessary privileges, compiles the Streams support package, and creates a database link from the ODSSTG user to the ODSSTG user on the EDW.

2. Create a database link connecting the ODSSTG user on the EDW instance to the ODSSTG user on the new source.

3. As the ODSSTG user on the EDW, execute the following procedure from SQL*Plus.

```
   SQL> SET SERVEROUTPUT ON
   SQL> MGKSTRC.P_CREATE_LOCAL_ENV(database link, source
alias);
```

   where you enter your institution’s values for the parameters in parentheses.

4. Create schemas in the EDW.

   For each schema in the source database that includes tables that will be staged in the EDW, create a schema in the EDW with the same name.

   4.1. Click **Staging** from the Administrative UI menu.

   4.2. Click **Maintain Stage Tables**.

   4.3. Choose a **Source Database**. You will only need to select a source database if your institution uses multiple source databases.

   4.4. Click **Add Another Schema to this List**.

   4.5. Select a schema from the **Schema to Add** list.

   4.6. Click **Submit**.

   4.7. Repeat step 4 for each new schema you want to add to the EDW.

5. Stage new tables in the EDW using the Administrative User Interface.

   5.1. Click **Staging** from the Administrative UI menu.

   5.2. Click **Maintain Stage** Tables.

   5.3. Choose a **Source Database**. You will only need to select a source database if your institution uses multiple source databases.

   5.4. Click the table owner for the area of tables that you want to add.

   5.5. Select tables from the **Tables to Add** list. Use Shift-click to select a contiguous range of tables or Ctrl-click to select non contiguous tables.

   5.6. Enter a **Run Date** (format dd-mon-yyyy) and **Runtime** (format hh24:mi:ss) to schedule when to run the job that will add the stage tables to the EDW. Enter *NOW* in each field to run the job immediately.

   5.7. Click **Submit** to schedule the job to run.

   The selected tables are added to the stage environment. A local record for each table is also created in the MGBSTRM table if a record doesn’t already exist in the table.

   5.8. Repeat step 5 for each new schema you added.

Warehouse UID

The Warehouse UID is a unique identifier for an entity within the warehouse. The Warehouse UID allows you to include information from multiple sources, for example, Banner products and Advance, in the same warehouse and relate entities that you need to track as a single entity.

The `WDT_WAREHOUSE_ENTITY` dimension table in the warehouse lets you define multiple source IDs that will map to the unique `WAREHOUSE_ENTITY_UID` (the Warehouse UID). The `WDT_WAREHOUSE_ENTITY` table includes a field for the `ADVANCE_ID` and `BANNER_PIDM` because warehouse expects information from two possible sources - Advance and Banner Advancement. These are the two unique IDs for the Advance and Banner Advancement systems.

The `WDT_WAREHOUSE_ENTITY` table also includes placeholders for five additional systems. The table includes five more System Name and System ID fields that you can define to add unique ID information from five other source systems.

OWB lookup commands ensure that the ID loaded in the warehouse is the Warehouse UID and not an ID from the source system.

Naming conventions

This section describes the naming conventions and standards applied to scripts and database objects used to create and maintain the warehouse components.

Administrative standards (IA_ADMIN Schema)

Administrative tables

Object name

Maximum length is 30 characters. See the table below.

1st Character	*M*- Mart	System Descriptor
2nd Character	*D* - Control Reports *G, T* - General Purpose	Table Purpose
3rd Character	*B* - Base *R* - Repeating *T* - Temporary *V* - Validation	Table Type
4th-7th Characters	Unique Descriptor	

Examples:

`MDBLOGH`, `MTVPARM`

Additional detail

Script names *must* follow unique 7-character naming standards. Script names are the same as the object name.

Administrative packages

Object name

Maximum length is 30 characters. See the table below.

1st Character	*M* - Mart	System Descriptor
2nd Character	*G* - General Purpose	Product Identifier
3rd Character	*K* - Package	Object Identifier
4th-7th Characters	Unique Descriptor	

Examples:

`MGKSECR`, `MGKPARM`

Additional detail

Script names *must* follow unique 7-character naming standards. Script names are the same as the object name.

Meta data tables and views

Object name

Maximum length is 30 characters. See the table below.

1st Character	*W* - Warehouse	System Descriptor
2nd Character	*M* - Meta Data	Table Purpose
3rd Character	*T* - Table or *V* -View	Object Identifier
4th Character	_ (underscore**)**	
5th-30th Characters	Unique Descriptor	

Examples:

`WMT_SOURCE`, `WMV_TARGET_OBJECT`

Additional detail

Script names are the same as the object name.

Sequences

Object name

See the table below.

1st Character	*M* - Mart	System Descriptor
2nd Character	*G* - General Purpose	Product Identifier
3rd Character	*S* - Sequence	Object Identifier
4th-7th Characters	Unique Descriptor	

Examples:

MGSHOST, MGSPARM, MGSPIDM, MGSSDAX

Additional detail

Script names *must* follow unique 7-character naming standards. Script names are the same as the object name.

EDW standards (EDWMGR/EDWSTG Schemas)

Warehouse tables

Star schema tables (EDWMGR Schema)

Object name:

Maximum length is 30 characters. See the table below.

1st Character	*W* - Warehouse	System Descriptor
2nd Character	*A* - Aggregate fact *D* - Dimension *G* - Group dimension *F* - Fact	Star Schema Table Type
3rd Character	*T* - Table *Z* - Snapshot Table	Object Identifier
4th Characters	_ (underscore)	
5-5th-30th Characters	Unique Descriptor	

Examples:

```
WDT_DEMOGRAPHIC, WFT_ADDRESS
```

Additional detail:

Script names are the same as the object name.

Staging tables (EDWSTG Schema)

Object name:

Maximum length is 30 characters. See the table below.

1st-3rd Characters	*CLN* - clean version of the dimension table *STG* - stage version of the dimension table	Stage Table Type
4th Characters	_ (underscore)	
5th-30th Characters	Unique Descriptor, describing name of the dimension For WDT_DEMOGRAPHIC _DEMOGRAPHIC	Dimension Descriptor

Examples:

```
STG_DEMOGRAPHIC, CLN_DEMOGRAPHIC
```

Additional detail:

Script names are the same as the object name.

Indexes and constraints

Primary Key indexes and constraints

Object name:

Maximum length is 30 characters. See the following table:

1st Character	PK_ (underscore)	Primary Key Prefix
4th-30th Characters		Table Name or Abbreviation (includes the first 4 characters, e.g., `WFT_`)

Example:

```
PK_WFT_EMPLOYEE, PK_WFT_OPERATING_LEDGER
```

Foreign Key constraints

Object name:

Maximum length is 30 characters. See the following table:

1st - 2nd Character	FK	Foreign Key Prefix
3rd Character	n	Where n is a one-up number
4th Characters	_ (underscore)	
5th-30th Characters		Child Table Name (omits the first 4 characters, e.g., WFT_)

Example:

```
FK1_ENROLLMENT, FK2_ENROLLMENT
```

4 Administrative User Interface

The Administrative User Interface (UI) enables you to easily perform the tasks required to set up and maintain the Banner Operational Data Store (Banner ODS) and the Banner Enterprise Data Warehouse (Banner EDW) at your institution. Your institution may licence either one or both products. This chapter includes information about using the Administrative UI to maintain both the Banner ODS and Banner EDW. You can reference the information that is appropriate for your institution’s environment. Review the map below to become familiar with the location of the options on the Administrative UI menus.

Preferences and Security

Institutional Preferences

Set Up Users & PIN

Set Up Data Display Rules

Set Up Banner ODS Security Rules

Set Up Banner EDW Security Rules

Options

Set Up Parameters

Schedule a Process

View Control Reports

View and/or Remove Scheduled Processes

Freeze Data Maintenance

Set Up and Maintain Cleansing Processes

List Events for a Banner EDW Star

Staging

Maintain Stage Tables

Report Staging Area Status

Refresh Staging Area Status

Refresh Staging Tables

Reconcile Stage Tables

Meta Data

Banner Operational Data Store

Banner Enterprise Data Warehouse

Maintain Banner ODS Meta Data

Maintain Banner EDW Meta Data

New Web Tailor Administration

Customize a Web Menu or Procedure

Customize a Graphic Element

Customize a Set of Information Text

Customize a Set of Menu Items

Update User Roles

Customize a Web Module

Customize Web Rules

Customize Web Tailor Parameters

Customize a Login Return Location

Customize Web Tailor Overrides

Customize Global User Interface Settings

There are a number of tasks involved in setting up and maintaining Banner ODS and Banner EDW. Some tasks are performed one time when you initially install and implement Banner ODS and Banner EDW. Other tasks are performed during implementation and on an ongoing basis. Each task is listed below, and is described in detail in later sections of this guide.

- Set up institutional preferences
- "Set up Users and PINS (Banner ODS and Banner EDW)" on page 4-3
- "Data Display Rules (Banner ODS)" on page 4-6
- "Set up Fine-Grained Access Security (Banner ODS and Banner EDW)" on page 4-17
- "Set up and Synchronize Data (Banner ODS)" on page 4-55
- "Set up Parameters (Banner ODS and Banner EDW)" on page 4-58
- "Schedule a Process (Banner ODS and Banner EDW)" on page 4-65
- "Freeze Data Maintenance (Banner ODS and Banner EDW)" on page 4-135
- Review and maintain "Meta Data (Banner ODS and Banner EDW)" on page 4-148

You can also use Web Tailor to perform some security functions and set some security-related preferences. In addition, Web Tailor gives you some options for customizing the appearance and behavior. For more information on using Web Tailor, see the "Web Tailor User Guide."

⚠ **Warning**

Because Banner ODS and Banner EDW contain sensitive business information, you should take standard precautions to prevent unauthorized access. User IDs and PINs should, of course, be kept secret, since anyone with a valid ID and PIN, and URL, can gain access to the system. ■

This section outlines all the tasks, and offers suggestions about when you want to perform them.

Set up Users and PINS (Banner ODS and Banner EDW)

Anybody using a reporting tool to access Banner ODS or Banner EDW must be defined as an Oracle User. Use your normal Oracle processes procedures to create user accounts.

After a user account is created, that user can report against Banner ODS or Banner EDW. Each user is listed in the Administrative UI on the View Banner EDW Business Profile and User Associations page. From that page, you can assign security rules for each user using a Business Profile. See "Set up Fine-Grained Access Security (Banner ODS and Banner EDW)" on page 4-17 for more information.

You should set up user accounts for Banner ODS and Banner EDW users at your institution based on how each user needs to use Banner ODS or Banner EDW. Banner ODS and Banner EDW include two types of users:

- Administrative Users—who require a user account so they can use the Administrative UI to set up and maintain Banner ODS and Banner EDW.
- Oracle Users—who require an Oracle user account (set up in your source system) so that they can use a reporting tool to access Banner ODS or Banner EDW and build reports.

Some users may be both Administrative and Oracle users, in which case they need a user account of both types. In these cases, you can use the same user ID in both systems (Administrative UI and Oracle).

PINs are disabled if the number of login attempts is exceeded (set on Web Tailor "Customize Web Rules" screen). They can be easily enabled on the Update a User Account screen using this checkbox.

Create Users and PINs

Administrative UI users set up and maintain Banner ODS and Banner EDW at your institution. Each Administrative user must have a unique ID and PIN created for them in order to gain access to the Administrative UI.

1. Click **Preferences & Security** from the Administrative menu.
2. Click **Set Up Users & PIN**.
3. Click **Create a New User Account** from the Set Up User and PIN page. The Create a New User Account page opens.
4. Enter a User ID.

A User ID can be one to nine characters, is limited to numbers and upper case letters, and may not contain spaces. (If you enter lower case letters, they will be converted to upper case letters.)

5. Enter First, Middle, and Last Names (only Last Name is required.)

6. Enter a PIN (It must be exactly six numbers; it cannot contain letters or special characters.)

7. Indicate whether the PIN is enabled or disabled.

8. Click **Create**.

Update Existing Users

Use this option to update misspelled or changed names, or to enable or disable a PIN.

If a user's login attempts are exceeded (as set up in Web Tailor, Customize Web Rules page), their PIN is disabled. Use this page to enable their PIN.

1. Click **Preferences & Security** from the Administrative menu.

2. Click **Set Up Users & PIN**.

3. Click an entry from the **Name** column on the Set Up Users and PIN page.

4. Change the fields. Only the Last Name field is required.

Note

The PIN must be exactly six numbers, and cannot contain letters or special characters. ■

5. Click **Update** to save. Or, click **Delete** to remove the User Account.

Update User Roles

User roles define which tabs of the Administrative UI a user can access. In turn, the roles permit or restrict a user to perform various tasks within the Administrative UI. When you create an Administrative user, the user is assigned the following user roles in Web Tailor: BPRA Meta Data, BPRA Options, BPRA Security, and Web Tailor Administration. This gives the user access to all options within both Banner ODS/Banner EDW, except for Banner ODS Staging, and New Web Tailor Administration menus. You may want to change a user's access, for example, to disable a user's ability to change security settings.

Perform the following steps within the Administrative UI to assign roles to a user by changing the user's defined roles in Web Tailor.

1. Click **New Web Tailor Administration** from the Administrative UI menu.
2. Click **Update User Roles**.
3. Enter or select the **User ID** to which you want to assign roles.
4. Click **Submit**.
5. Check which roles to assign to the user. Refer to the "User Roles" descriptions to determine which roles to assign to each user.
6. Click **Submit Changes**.

User Roles

The following table defines each of the User Roles available within the Administrative User Interface.

User Role	Description
BPRA Meta Data	Allows a user to access the Meta Data tab within the Administrative User Interface where the user can: • View the published Banner ODS and Banner EDW meta data. • Update, edit, and publish the Banner ODS and Banner EDW meta data. • View and recover baseline records marked for deletion.
BPRA Options (Jobs, Parameters, FreezeData)	Allows a user to access the Options tab within the Administrative User Interface where the user can: • Define internal system parameters. • Run jobs to load and update data in the Banner ODS. • Manage aspects of freezing data in the Banner ODS. • Define and maintain data cleansing translation rules.View lists of events related to Banner EDW Stars.

User Role	Description
BPRA Security	Allows a user to access the Preferences & Security tab within the Administrative User Interface where the user can: • Set institutional preferences. • Create and delete users and reset user's PINs. • Create and maintain data display rules • Create and maintain fine-grained access security rules defining which values a user can view in the Banner ODS and the Banner EDW.
BPRA Staging	Allows a user to access the Staging tab within the Administrative User Interface where the user can: • Add and remove staging tables in the Banner ODS. • Run the Staging Area Status report and view a list of stage tables, the status of processes, and details of staging errors.
Web Tailor Administrator	Allows a user to access the Web Tailor Administration tab within the Administrative User Interface where the user can customize various aspects of the Administrative Tool.

Data Display Rules (Banner ODS)

Display rules enable you to control and customize how data stored in the Banner ODS composite tables is displayed in the reporting views. There are two types of display rules: positional rules and hierarchical rules.

There are also a number of display rules used to determine a value stored in either a Banner ODS composite table or displayed in a Banner ODS Reporting view. All display rules are stored in Banner ODS database table - MGRSDAX.

Positional display rules

Positional display rules define the specific location (by position) of data in a view. Slotted views or tables require a set of positional display rules to store information in a meaningful way.

Example - Positional Display Rules (for Slotted views)

The TEST view in Banner ODS displays all valid test score values loaded from your source system to Banner ODS. This data is stored in a vertical presentation as "one row per person per test". The corresponding TEST_SLOT view provides an alternative horizontal presentation, that ‘flattens’ the data to “one row per person with the details of (up to) seven test scores.” Positional display rules are required to define which seven test scores will be included, and in what position or order they will appear within this “slotted” presentation. These Display rules are used to build the underlying MST_TEST_SLOT table.

Hierarchical display rules

Hierarchical display rules define a specific order in which to retrieve a set of related data. Hierarchical display rules are required for a subset of (non-slotted) Reporting views.

Example – Hierarchical Display Rules (for applicable non-slotted views)

The PERSON_ADDRESS and ADDRESS_ BY_RULE view displays one address per entity per ADDRESS_RULE (stored in MGRSDAX as an Internal Code under the Internal Group of ADDRESS, and must end in ADDR) to be used for mailing purposes.

The mailing address displayed is based on the hierarchical display rules created to determine which address types should be retrieved for the mailing address. You can create a series of hierarchical display rules based on priority, so that if “address type 1” does not exist, get “address type 2” and so on.

To invoke the ADDRESS_BY_RULE reporting view rule, add a Filter/WHERE clause that states “where ADDRESS_RULE = IC_REG_ADDR.” This will retrieve the first current address found in the source system for the hierarchy you created.

When Banner ODS is first installed, MGRSDAX (Banner ODS table that stores display rules) is populated with specific rules from your source system, as well as rules delivered with the product. The records (or display rules) in MGRSDAX match external codes (institution specific values) with internal codes (system defined values). After Banner ODS system is installed, you must then use the Administrative UI Preferences and Security option and Set Up a Display Rule to review and update the display rules in MGRSDAX. This ensures that display rules match your criteria, and are set up to meet your reporting needs.

Multiple display rules can also be managed, or assigned, using business profiles. (See “Set up a Display Rule” on page 4-12 for information on setting up business profiles.)

Note

Business profiles are only used when more than one Oracle user is used to access the data from your institution supported report writer. ■

If business profiles are used, then the system pulls the appropriate values for the profile with which the user is associated, if a rule exists for that profile.

⚠ **Warning**

If multiple profiles exist for that user, then the first profile with a matching display rule is used. ■

Note

If no display rules are found for any profiles assigned to the user, the display rule for the default profile (INSTITUTION) is used. ■

For reporting views such as the TEST_SLOT view, use business profiles to designate unique sets of test score data and the positional order of that data within the view for different business offices and users at the institution.

For hierarchical reporting views such as the PERSON_ADDRESS view, business profiles enable you to designate unique sets of (mailing) address type hierarchies for different business offices and users.

Example:

A display rule consists of one or more related records in MGRSDAX. Records that share the same Profile Code, Internal Group and Internal Code values make up a single display rule. The display rule also includes the Business PROFILE_CODE that defaults to INSTITUTION or is set to an institution defined value.

MGRSDAX is delivered with the following records that all have an Internal Group value of ADDRESS, and the business profile of INSTITUTION.

Internal Group: ADDRESS

Profile Code	Internal Code	Internal Code Sequence	External Code
INSTITUTION	ALUMMAIL	1	BUS
INSTITUTION	ALUMMAIL	2	ART
INSTITUTION	ALUMMAIL	3	RES
INSTITUTION	ALUMMAIL	4	CPS
INSTITUTION	RECRLETR	1	ACCEPT
INSTITUTION	RECRLETR	2	CHKL
INSTITUTION	RECRLETR	3	COLLEGE NIGHT
INSTITUTION	RECRLETR	4	DCSN
INSTITUTION	RECRLETR	5	INTERVIEW ONE

The first four records also share the same Internal Code value of ALUMMAIL. These four records make up the Display Rule that defines which Mail codes to retrieve for Advancement-related reporting views. The last five records share the Internal Code value RECRLETR. These five records make up the display rule that defines which MAIL internal codes to retrieve for the COMMUNICATION_SLOT and Recruiting-related reporting views.

By editing the above values to reflect the Advancement and Recruiting Mail internal code values used by your institution, your users can then report on the desired data. Before your users begin creating reports, you need to review all of the delivered display rules, and edit them to reflect your institution's specific values.

Note

After changing display rules for views that work from slotted database tables, the corresponding slotted tables must be reloaded before the updated values will display in the reporting views seen by your users. By default, this happens during the incremental refresh cycle, which typically occurs nightly. However, if you want to see more immediate results, reload the corresponding slotted table(s) manually via the Schedule a Process page. See "Schedule a Single Process" on page 4-69 ■

Note

Also note that there are few reporting views, like the PERSON_ADDRESS and ADDRESS_BY_RULE, that go directly against the rules in the MGRSDAX database table and do not need to be reloaded for you to view the changes. ■

Display Rule Information in Published Meta Data

Meta Data includes a business definition for each reporting view. When the reporting view being defined uses display rule entries from Banner ODS MGRSDAX database table, the required rule code, INTERNAL _GROUP and INTERNAL_CODE values are explained as part of the business definition. Most reporting views that require MGRSDAX rules have a column labeled PROFILE_CODE, and a column with the name of the view and XXXXXXXX_RULE that are used as the INTERNAL_GROUP for that set of display values.

When the reporting view has a column that uses the MGRSDAX database table, that is explained in the column business definition.

Display Rule Cross-Reference Chart

Display rules are defined by a set of records stored in Banner ODS database table, MGRSDAX. You can use the Display Rule Cross-Reference Chart to identify display rule value combinations as they are delivered.

The Display Rule Cross-Reference Chart lists all views, tables, procedures or packages that use the MGRSDAX table. The chart enables you to see the rule values that are set up to retrieve the data, and how your solution is impacted if changes are made to the display rules on MGRSDAX. The codes on the chart followed by an asterisk (*) indicate user defined rules that can be changed to fetch the EXTERNAL_CODE or REPORTING_DATE.

To open the Display Rule Cross-Reference Chart, access the `displayRulesXREF.csv` file delivered with the product documentation.

You can open the file in Microsoft Excel or a similar spreadsheet application. You can reorganize the columns as needed. A description of each column on the chart follows.

Column	Description
REPORTING_VIEWS	The view that is directly affected by a change to an MGRSDAX value in Banner ODS.
INTERNAL_GROUP	Value Banner ODS is using to connect the set of display rules with the reporting view and/or column that are to use them. These values are coded within Banner ODS and must be used for the purpose specified.
INTERNAL_CODE	Institutions may define any values as required to represent the business rules of the institution. Some values are extracted from Banner GTVSDAX rules for institutions that use the O:A views.
EXTERNAL_CODE	*X* identifies valid institutions values must be provided.
REPORTING_DATE	*X* indicates that the Reporting Date is used for sequence of display values.
TABLES	Banner ODS Composite table used as the basis for the selection of values based on the display rules defined by the institution on the MGRSDAX database table.

Column	Description
COMPARISON COLUMN	Column within Banner ODS composite table that is used to retrieve data based on the value in the either MGRSDAX_EXTERNAL_CODE or MGRSDAX_REPORTING_DATE. ***Example:*** TEST rule: The MSKTEST package gets the MGRSDAX_EXTERNAL_CODE value from MGRSDAX based on the MGRSDAX_INTERNAL_GROUP = 'TEST' and the MGRSDAX_INTERNAL_CODE = 'STDNTEST'. This value is then used to retrieve records from the TEST column in MST_TEST to populate the MST_TEST_SLOT table and the TEST_SLOT view.
REPORTING	Package View Name of the package or view in which MGRSDAX is referenced. Procedure/Function Name of the process or function being used by MGRSDAX.

Records with the same Profile Code, Internal Group and Internal Code combination make up one display rule. The display rules that are delivered have a default business profile code of INSTITUTION.

Note

When more than one Internal Code is listed, there are multiple display rules for the value in the Internal Group. For example, there are several ADDRESS rules listed for different departments like Admissions (Internal Code = ADMSADDR), Faculty (Internal Code = FACLADDR), Recruiting (Internal Code = RECRADDR), etc. ■

Note

When more than one Profile Code is listed, there are multiple display rules for the value in the Internal Group. ■

The example below will help you tie together one use of the chart with the Administrative UI.

Example:

You want to see what display rules exist for (or are impacted by) the VENDOR reporting view because you want to change the external code for that reporting view. Follow the steps below:

1. The copy of the chart is already sorted in alphabetical order by Reporting View. Look in the Reporting View column (the first column) of the chart. Find VENDOR. It is near the end of the list.

   You will find that the assigned Profile Code is INSTITUTION, the Internal Group is ADDRESS and the Internal Code is VENDADDR for VENDOR.

2. Open the Set Up a Display Rule web page in your Administrative UI.

3. Select the Profile Code (INSTITUTION), Internal Group (ADDRESS) and Internal Code (VENDADDR) from the drop-down lists.

4. Click **Search**. The Select an Existing Display Rule page opens. This page shows the display rule for the reporting view VENDOR.

5. To change the External Code, click **BU** under the External Code column.

   The Update an Existing Display Rule page opens. You can change the external code from this page.

6. Click **Save**.

Set up a Display Rule

You may want to create new display rules by adding new internal codes for a business purpose, or by adding additional external codes not currently defined.

Note

You may want to set up your business profiles before you set up display rules. ■

To create a new rule, follow the steps below:

1. From the Administrative menu, Click **Preferences & Security**.

2. Click **Set Up Data Display Rules**.

Note

If a PROFILE_CODE is to be used in the display rule, it must be set up first. See “Set up Fine-Grained Access Security (Banner ODS and Banner EDW)” on page 4-17 for information on setting up business profiles. ■

3. Click **Create** from the Set Up a Display Rule page.

4. Enter the information for the new display rule, or click an existing code from one of the drop-down lists. Each field is described below:

Field	Description
Business Profile Code	Business Profile for which you want to set up display rules. You can create additional Business Profiles from the Create a Banner ODS Business Profile or Create a Banner EDW Business Profile web page. *INSTITUTION* is the default code for users for whom no other business profile is defined.
Internal Group	High-level group of rows of data (Internal Codes) that are categorized together to provide multiple entries for a single concept. The value is predefined in the system. It should *not* be changed, but new internal groups can be added for client specific processing. (Click the appropriate value from the Internal Group list.)
Internal Code 1	Specific code relationships for source system concepts. This field is used internally within PL/SQL functions and procedures to determine which row(s) to retrieve from the MGRSDAX table. You can add new internal codes to be used for business purposes, and then click the appropriate code when writing a report. (Click the appropriate value from the Internal Code list.)

Field	Description
Internal Code Sequence Number	Internal sequence number that provides either a hierarchy or positional identifier: ***Hierarchy identifier*** Order in which to retrieve display rule driven data (such as the Address Type for a designated mailing address in the PERSON_ADDRESS view). Each sequence number should be a single numeric value. Give the most important code value a sequence number of 1, and number each subsequent value consecutively (such as 2, 3, 4). ***Positional identifier*** Position within a slotted view where a repeating group should appear. The sequence number entered by the user must correspond to the slotting concept applicable to the slotted view for which the display rule is being created (such sequence numbers 1-7 for the seven available test score slots in the TEST_SLOT view).
External Code	Institution-specific values that usually come from rules and validation tables in the transactional or administrative source system. Enter the values used by your institution to define either the hierarchy or the positional value for the particular display rule. **Note:** Change this value so that the external codes match your institution's code value for each display rule value.

5. Click **Save**. The Update an Existing Display Rule page opens.

Note

After changing display rules the corresponding slotted tables must be reloaded for those changes to take effect. By default, this happens during the incremental refresh cycle, which typically occurs nightly. However, if you want to see more immediate results, reload the corresponding slotted table(s) manually via the Schedule a Process page. See "Schedule a Single Process" on page 4-69 ■

Update Display Rules

You may want to display different types of test scores, address information, etc. If the display rule already exists, then you can use the steps below to add, update, duplicate or delete display rules.

Note

You can use these steps for every Profile Code, Internal Group and Internal Code combination listed in the table in the "Display Rule Cross-Reference Chart" section. ■

1. From the Administrative menu, click **Preferences & Security**.

2. Click **Set Up Data Display Rules**. The Set Up a Display Rule page opens.

3. Choose a Business Profile, Internal Group, and Internal Code combination from the drop-down lists on the Set Up a Display Rule page. Or, you can show all groups and codes.

4. Click **Search**. The Select an Existing Display Rule page opens.

Note

Use the Meta Data reporting view business definition and the Display Rule Cross Reference chart, available from the **Help** button in the Administration UI, to identify Internal Group and Internal Code combinations that make up a Display Rule. Information about this chart is available in the "Display Rule Cross-Reference Chart" section. ■

5. Review all information for the selected combination. Determine the data on which your users want to report (it may be different from what is delivered). Create a list of the data you want to use in place of the data that was delivered.

6. Choose an External Code link from the External Code column to edit a record. The Update an Existing Display Rule page opens.

7. Make your change

8. Click **Save** to save the display rule. Click **Delete** to remove the display rule.

Note

After changing display rules, the corresponding slotted tables must be reloaded for those changes to take effect. By default, this happens during the incremental refresh cycle, which typically occurs nightly. However, if you want to see more immediate results, reload the corresponding slotted table(s) manually via the Schedule a Process page. See "Schedule a Single Process" on page 4-69. ■

Duplicate Display Rules

To save time, you can copy the settings from an existing display rule and use it to create a new display rule.

1. From the Administrative menu, Click **Preferences & Security**.
2. Click **Set Up Data Display Rules**. The Set Up a Display Rule page opens.
3. Choose a Business Profile, Internal Group and Internal Code combination from the drop-down lists on the Set Up a Display Rule page. Or, choose to show all groups and codes.
4. Click **Search**. The Select an Existing Display Rule page opens.
5. Choose an external code link from the External Code column. The Update an Existing Display Rule page opens.
6. Enter the External Code information or select it from the drop-down list.
7. Click the **Duplicate**. The Create a New Display Rule page opens.
8. Replace the information for the existing display rule with the information for the new display rule.
9. Click **Save** to save your settings.

Note

After changing display rules, the corresponding slotted tables must be reloaded for those changes to take effect. By default, this happens during the incremental refresh cycle, which typically occurs nightly. However, if you want to see more immediate results, reload the corresponding slotted table(s) manually via the Schedule a Process page. See "Schedule a Single Process" on page 4-69. ■

Reload using a Single Extract Transform and Load (ETL) Slot Process

Changes made to a display rule affect all associated slotted tables and reporting views. The ETL slot process must be rerun before any changes made to slotted tables or display rules can be viewed in the slotted reporting views. If only one slotted table was changed, then this process enables you to quickly run a single slot process. Use the following steps to schedule when you want to run a slot process job.

1. Click **Options** from the Administrative UI menu.
2. Click **Schedule a Process**. The Select a Process page opens.

3. Click **Schedule Banner ODS Mappings**, **Schedule Banner EDW Snapshot Mappings**, or **Schedule Banner EDW Operational Mappings**. The Select a Subprocess page opens.

4. Click **Run A Single ETL Slot Package** from the Select a Subprocess page. The Schedule a Process page opens.

5. Choose the table from the Slotted Table to Reload drop-down list.

6. Enter the required Scheduling Parameters information.

    6.1. Enter a Run Date (format dd-mon-yyyy) and Runtime (format hh24:mi:ss).

    6.2. If you want to run the process on a recurring basis, enter an Interval.

    Click the link next to the Interval field. A sample Interval window opens. Click the link under the Interval Expression column for the interval in which you want to schedule a process. For example, to run a process every day at the same time click SYSDATE+1.

7. Click **Save** to save the information about this job. The job is entered into the job queue to run at the specified day and time.

Set up Fine-Grained Access Security (Banner ODS and Banner EDW)

Banner ODS and Banner EDW include two types of users:

- Oracle users who require an Oracle user account in your source system so they can access Banner ODS and Banner EDW to build reports
- Administrative users who require a user account in the Administrative UI so they can use the UI to maintain Banner ODS and Banner EDW.

This section explains how fine-grained access security applies to the first type of users - Oracle users when they access Banner ODS or Banner EDW for reporting.

Fine-grained access security lets you selectively restrict an Oracle user's access to rows of Banner ODS or Banner EDW data based on the value of a specific data element. For example, you might allow a user to see data only from their own department. After you set up security rules and assign them to Oracle users, the rules are applied when the user searches for information within Banner ODS or Banner EDW.

Note

This security applies to the rows of data returned, not the columns. To ‘mask’ columns of data for a given reporting view, create a copy of the view with those columns removed that contain sensitive data.

Secured access to data is controlled by Oracle Policies, in conjunction with the security rules set up in the Administrative UI. A policy is an Oracle construct that applies a WHERE clause predicate to any queries made against a table. A security rule is simply data in Banner ODS or Banner EDW security tables that determine what that WHERE predicate should look like for a given user.

By default, Banner ODS and Banner EDW are delivered with no Policies (no security restrictions) on any tables. Therefore, you can set up data access values (security rules) for given users without affecting any user's ability to access Banner ODS or Banner EDW data. However, once Policies are defined for the tables, users can only access data to which they have been granted permission.

Once a policy is set up on a Banner ODS or Banner EDW table, Oracle calls the MGKSECR package to create a WHERE clause predicate any time that the database table is accessed, such as using a SELECT query. The MGKSECR package, in turn, uses the security rules data to generate the appropriate WHERE clause predicate for the current Oracle User ID. For users with access set to "all" (either All Banner ODS Data, All Banner EDW Data, All Data for that Area, or All Data for all columns and rules in the table), MGKSECR does not generate a predicate, thereby allowing those users full access to that data. For rules that list access to particular values, for example campus codes of A, B, or C, MGKSECR generates a corresponding WHERE clause code with the appropriate level of restriction.

Note

Security rules are cumulative -- they are joined with an AND clause. Users must be granted access rights for each rule in a table in order to gain access. For example, if a table has three security rules defined, and two of the rules give *all* access, but the third rule gives the user access to *none*, that user will not have access to any data in that table. ■

You can manage users by grouping similar users together as business profiles. You can also manage Security and Display Rule assignments as a group rather than at the individual user account level.

Use Banner ODS menu selections in the order below to set up your security:

1. Set up Organizational Areas.

   Set up one or many organizational areas by grouping similar areas together. See "Set up and Maintain Organizational Areas" on page 4-19 for additional information.

2. Set up and Maintain User ID Translations, and Set up Business Profiles

   These menu options can be completed in any order.

   - User ID Translations

     Bring Banner data into Banner ODS fine-grained access. See "Banner User ID Translations" on page 4-21 for additional information.

- Business Profiles

  Group similar users together. See “Set up Business Profiles” on page 4-24 for additional information.

3. Security Rules

   Defines the data that each user can access. See “Set up and Maintain Security Rules” on page 4-27 for additional information.

4. (optional) Security Predicates

   Review the code that generates the predicate in MGKSECR to determine if it is aligned correctly with your business rules. Also, verify the code that is generated for a security predicate. See “Security Predicates” on page 4-49 for additional information.

5. Assign Security Rules

   Enables security rules to work. Policies are either turned on or turned off. See “Policy Management” on page 4-50 for additional information.

6. Transfer Banner Fine-Grained Access process.

   This process transfers data for Finance Fund, Fund Type, and Organizations, and for Human Resources Organizations, and Employee Class from Banner to Banner ODS. To transfer additional data you need to set up additional rules. See “Transfer Banner Fine-Grained Access” on page 4-112 for additional information.

Set up and Maintain Organizational Areas

Organizational Areas are used to set up and group organizational areas together, and to help simplify the implementation of Banner ODS fine-grained access.

Example

If you have users in the Human Resources area that should have access to all of the Human Resources tables. Instead of granting access for each user to each individual Human Resources table, you can define an Organizational Area called “HR” (the name is user-defined). Then, when you create your Banner ODS Security Rules for Human Resources tables, assign those rules to the “HR” Organizational Area. Once your Organizational Areas and Human Resources Security Rules are created, go to the Assign Security Rules page. Select your Human Resources users then, check the **Access All Data In This Area** check box. This gives the user access to all tables included in the “HR” Organizational Area.

Organizational Areas can be set up in any manner you wish. In the example above, an Organizational Area was created which included all Employee tables. However, you could also set up Organizational Areas that cross Banner product groups or you could set up Organizational Areas that are subsets of a product group. The idea is that you can set up Organizational Areas in any way that makes sense for the way you divide security among your reporting users.

Banner ODS is delivered with sample Organizational Areas and sample Security Rules that are assigned to them. The sample data gives an idea of how to go about setting up your own Organizational Areas and the Security Rules that apply to them.

Create a Banner ODS Organizational Area

Use to create organizational areas.

1. Click **Preferences and Security**.
2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.
3. Click **Set up and Maintain Organizational Areas**.
4. Click **Create a Banner ODS Organizational Area**.
5. Enter the code and description.
6. Click **Save**.

Update a Banner ODS Organizational Area

Use to update organizational areas.

1. Click **Preferences and Security**.
2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.
3. Click **Set up and Maintain Organizational Areas**.
4. Click an organizational area code description.
5. Select another organizational code, or change the current description.

Note

The table at the bottom of the page indicates what rows in that table will be deleted if you delete the organizational area. ■

6. Click **Save**.

Delete a Banner ODS Organizational Area

Use to delete organizational areas.

1. Click **Preferences and Security**.
2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.
3. Click **Set up and Maintain Organizational Areas**.
4. Click an organizational area code description.
5. Select another organizational code, or change the current description.

Note

> The table at the bottom of the page indicates what rows in that table will be deleted if you delete the organizational area. ■

6. Click **Save**.

Banner User ID Translations

Use Banner User ID Translations to match Banner security user IDs with Banner ODS security IDs if they are different and you plan to run the Transfer Banner Fine-Grained Access process.

The MGBXWLK table (owned by the IA_ADMIN schema and set up through the Banner User ID Translations pages) is used to associate the two IDs. MGBXWLK contains two primary columns: the Banner User ID and the Banner ODS User ID. The Banner ODS User ID is not required, therefore you can transfer all Banner User IDs into the MGBXWLK table without triggering constraint errors. MGBXWLK has two primary purposes:

- Facilitate data transfer when user IDs are not the same
- Additional security. You may not want everyone with fine-grained access information in Banner to be able to access the data in Banner ODS. In that case, you would follow the instructions in "- Restrict the Information Transferred to a Limited Group of Users" on page 4-22. Only those users whose user IDs were added to MGBXWLK are able to access Banner ODS data after all the fine-grained access policies are enabled.

The MGBXWLK table is populated based on the scenarios below.

- Banner User IDs are the same as the Banner ODS User IDs

- The MGBXWLK table does not need to be populated

- The delivered Administrative parameter record with internal group *BANNER TO ODS FGA TRANSFER* and internal code *ODS USER ID NOT FOUND* is used to tell the transfer job what to do when a given Banner ODS user ID is not found in MGBXWLK. As delivered the value of External Code is *USE BANNER USER ID*.

- Some User IDs are the Same in Banner and Banner ODS, and Some are Not

- Enter only users with different Banner and Banner ODS user IDs into the MGBXWLK table (using the Set Up and Maintain Banner User ID Translations pages). Users with the same user ID in Banner and Banner ODS can be omitted from the table.
- The delivered Administrative parameter record with internal group *BANNER TO ODS FGA TRANSFER* and internal code *ODS USER ID NOT FOUND* is used to tell the Transfer Banner Fine-Grained Access process what to do when a given Banner ODS user ID is not found in MGBXWLK. As delivered the value of external Code is *USE BANNER USER ID*.
- If you populated the MTVPARM record with an external code of *USE BANNER USER ID*, but populated MGBXWLK with only the Banner User IDs and the Banner ODS User IDs have not yet been populated, the process "Transfer Banner Fine-Grained Access" on page 4-112, (MGKXFER.P_TransferFGA), does not read the MGBXWLK table and the Banner User ID is used.

- All Users are to have a record in MGBXWLK, regardless of whether the Banner and Banner ODS User IDs are the Same

- Add all Banner user IDs (**Banner User ID** field**)** and Banner ODS user IDs (**Banner ODS User ID** field) to MGBXWLK. This includes users with the same Banner user ID as their Banner ODS user ID.
- Enter the same MTVPARM record as "- Banner User IDs are the same as the Banner ODS User IDs" and "- Some User IDs are the Same in Banner and Banner ODS, and Some are Not", but with an external code of *DENY ACCESS*.

- Restrict the Information Transferred to a Limited Group of Users

- Add the limited set of Banner user IDs to MGBXWLK. If the Banner ODS user IDs are different, enter them in the **Banner ODS User ID** field. If the Banner User IDs are the same in Banner ODS, enter the Banner User IDs in the **Banner User ID** field and the **Banner ODS User ID** field.
- Enter the same MTVPARM record as "- Some User IDs are the Same in Banner and Banner ODS, and Some are Not" and "- All Users are to have a record in MGBXWLK, regardless of whether the Banner and Banner ODS User IDs are the Same" but with an external code of *DENY ACCESS*.

Create Banner User ID Translations

Use this to match a Banner user ID with a Banner ODS user ID.

Prerequisites

It is recommended that the Banner ODS ID is set up so that it can be selected from the drop-down list that appears when you select **Select a Banner ODS User ID** on the translation Create and Update pages.

1. Click **Preferences and Security**.
2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.
3. Click **Set up and Maintain Banner User ID Translations**.
4. Click **Create a New User ID Translation.**

   If no User ID translations exist, you are taken directly to the Create a New User ID Translation page.
5. Enter the Banner user ID, or click the Select a **Banner User ID** link to choose it from the list.

   The Banner User IDs are drawn from the Banner Finance tables FORUSFN, FORUSOR, and FOBPROF, and the Banner HR tables PSRORGN, PTRUSER, and PSRECLS.
6. Enter the Banner ODS user ID, or click the **Banner ODS User ID** link to choose it from the list.

   The Select a Banner ODS User ID list is drawn from the WAV_ALL_USERS view which contains a list of IDs for users most likely to run the reports. Your institution can change this view to include additional users (ODSMGR IA_ADMIN, for example) so that additional user IDs will appear in the list.
7. Click **Save**.

Update Banner User ID Translations

Use this to change the Banner ODS user ID matched with a Banner user ID.

1. Click **Preferences and Security**.
2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.
3. Click **Set up and Maintain Banner User ID Translations**.
4. Select the Banner user ID you want to change.
5. Enter the Banner ODS user ID, or click the link to select if from a list.
6. Click **Save**.

Delete Banner User ID Translations

Use this to delete the Banner ODS user ID matched with a Banner user ID.

1. Click **Preferences and Security**.
2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.
3. Click **Set up and Maintain Banner User ID Translations**.
4. Select the Banner user ID you want to change.
5. Enter the Banner ODS user ID, or click the link to select if from a list.
6. Click **Delete**.

Set up Business Profiles

Business Profiles enable you to easily manage groups of users by grouping similar users together. In turn, you can manage Security and Display Rule assignments as a group rather than at the individual user account level.

First you create a Business Profile, then associate one or more users with that Business Profile, or associate one or more Profiles with one or more users.

Multiple display rules can also be managed, or assigned, using business profiles.

Note

Business profiles are only used when more than one Oracle user is used to access the data from your institution supported report writer. ■

If business profiles are used, then the system pulls the appropriate values for the profile with which the user is associated, if a rule exists for that profile.

Note

If multiple profiles exist for that user, then the first profile with a matching display rule is used. If no display rules are found for any profiles assigned to the user, the display rule for the default profile (*INSTITUTION*) is used. ■

For reporting views such as the TEST_SLOT view, use business profiles to designate unique sets of test score data and the positional order of that data within the view for different business offices and users at the institution.

For hierarchical reporting views such as the PERSON_ADDRESS view, business profiles enable you to designate unique sets of (mailing) address type hierarchies for different business offices and users.

Create a Business Profile

Perform the following steps to create a business profile.

Prerequisite

Create an organizational area.

1. Click **Preferences and Security**.
2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.
3. Click **Set Up Business Profiles**.
4. Click **Create a Banner ODS Business Profile or Create a Banner EDW Business Profile**.
5. Enter a new profile code and description.
6. Click **Save**.

   See “View, Update or Delete a Business Profile” on page 4-26 for steps on updating and viewing Business Profiles.

Associate Business Profiles with a User

Perform these steps to associate a Business Profile with a user or group of users. You can also link to the Set Up Banner ODS Security Rules or Set Up Banner EDW Security Rules pages to set up security rule assignments for that profile or user.

1. Click **Preferences and Security**.
2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.
3. Click **Associate Users and Business Profiles**.
4. Choose the user to which you want to associate (or view existing) Business Profiles. If you selected the user from the user drop-down list, then click **Refresh Profile List** to redisplay the business profiles list for that user. Below the user drop-down list is an alphabetical list of all Business Profiles and the user name associated with them.
5. Check or uncheck the corresponding check boxes to associate or disassociate Business Profiles with the user.
6. To set up security rules for a user, click **Assign Security Rules**. See “Set up Fine-Grained Access Security (Banner ODS and Banner EDW)” on page 4-17 for additional information.
7. Click **Save** to update the user associations.

Associate Users with a Business Profile

Use this option to associate a user or group of users with a Business Profile. You can also link to the **Set Up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules** pages to set up security rule assignments for that profile or user.

1. Click **Preferences and Security**.

2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.

3. Click **Associate Users and Business Profiles**.

4. Choose the Business Profile to which you want to associate (or view existing) users from the **Business Profile** column.

Note

> When you select the Business Profile column or Oracle User Name column, the table toggles between associating a Business Profile with a user and associating a user with a Business Profile. ■

5. Check the corresponding check boxes to associate or disassociate users with a Business Profile.

6. Click **Save** to submit your changes.

7. To set up security rules for a Business Profile, click **Assign Security Rules**.

   See "Set up Fine-Grained Access Security (Banner ODS and Banner EDW)" on page 4-17 for instructions on assigning security rules.

View, Update or Delete a Business Profile

Use this option to change or delete a Business Profile.

1. Click **Preferences and Security**.

2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.

3. Click **Set Up Business Profiles**.

4. Click the description of the Business Profile you want to change.

   The Update a Banner ODS Business Profile or Update a Banner EDW Business Profile page opens. From this page you can change the descriptions or delete the Business Profile.

5. Make your changes to the description.

6. Click **Save** to submit your changes.

   or

   Click **Delete** to remove the displayed profile.

Note

The table at the bottom of the page indicates what rows in that table are also deleted if you delete the business profile. ■

Set up and Maintain Security Rules

The following tables (in the IA_ADMIN schema) are used to store the security rules information in Banner ODS and Banner EDW.

Table	Functional Name	Security Rules Stored
MGBFGAA	Fine Grained Access User Areas	Indicates if the user has access to all of the elements and values within an area code
MGBFGAE	Fine Grained Access User Elements	Indicates if the user has access to all of the values within an element code
MGBFGAV	Fine Grained Access User Values	If the user does not have the MGBFGAV_ALL_IND or MGBFGAA_ALL_IND for an element, area, or all of FGA, indicates which values for the element the user may access.
MGBSECR	User Security Table.	Various user security related data.
MGBFGAR	Fine Grained Access Element Rule Table	The security rules that consist of Banner ODS/ Banner EDW tables and columns that have security applied to them.
MTVFGAA	Fine Grained Access Area Validation Table	The security rules that consist of Banner ODS/ Banner EDW area that have security applied to them.

Understanding the data relationships in these tables is best explained by reviewing the Administrative UI that maintains that data.

To set up security, you need to:

- Determine the data security requirements
- Set up and maintain the security rules

Determine Data Security Requirements

Use this section to determine whether it's necessary to restrict some users' access to some of the data within Banner ODS and Banner EDW and to determine the specific security restrictions that apply to each user.

⚠ **Warning**

When deciding whether to apply fine-grained access, keep in mind that its use limits the accuracy and usefulness of data. The system does not inform users that the data they are seeing has been filtered by fine-grained access security. This can cause incorrect numerical results in some circumstances. ■

Example

If a user queries across the entire institution, and that same user has been restricted from seeing data from some departments. Although the data appears to cover the whole institution, it does in fact sum data only from those departments which the user is allowed to access. The user may draw incorrect conclusions if he or she is unaware that the data is incomplete.

If you choose to use fine-grained access, you have the following options for the level of access you can give an individual Oracle user who accesses Banner ODS:

- Full access to all data in Banner ODS.
- Full access to all data at the level of the Organizational Dimension, for example, Academic, Course and Academic, Financial, or Workforce.
- Full access to all data at an element level, for example, college, department, major, organization, or fund level.
- Restricted access to data at the element level based on a list or range of values for a specific data element, for example., allow a user to access only data related to the user’s department or a range of fund codes.

Set up a Security Rule

If you want to secure data at a granular level, you need to create the security rules that define that level of security. A security rule consists of an Organization Dimension, Table, Rule Type, and Column (you may define one or two columns).

Setting up a rule involves entering and maintaining the data that comprises a rule in the MGBFGAR table. You can use the Administrative UI to create and maintain the list of security rules that can be applied to a given user account, and to assign particular values for a given rule to a given user account. (Another method available using the Administrative UI is to assign values for a given rule using “Set up Fine-Grained Access Security (Banner ODS and Banner EDW)” on page 4-17. The Administrative UI uses the MGKFGAC package to apply the security rules you define.

Use the “Set up and Maintain Security Rules” on page 4-27 option within the Administrative UI to create, update, delete, and search for rules. (These processes are described in the next few sections.) Creating or updating rules is reflected in the MGBFGAR table. Deleting rules changes the MGBFGAR table, but in addition, any values related to a rule that are deleted are cascaded through the other fine-grained access tables. There is a list at the bottom of the security rules web pages that indicates what rows in the table are deleted if the security rule is deleted.

Sample security rules (generated from the ods\ia_admin\dbscripts\mgbfgar_data_ods.sql script) are added to MGBFGAR when ODS is installed or upgraded. The delivered sample Finance and Human Resources security rules reflect the way that security rules should be set up if you plan to use Transfer Banner Fine-Grained Access. Since those rules are added to MGBFGAR by the install or upgrade, they can be viewed through the Administrative UI Set Up and Maintain Security Rules pages.

Prerequisites

- Create organizational areas
- Create business profiles

1. Determine a Banner ODS or Banner EDW table and column value on which you want to secure information.
2. Click **Preferences & Security** from the Administrative menu.
3. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.
4. Click **Set Up and Maintain Banner ODS Security Rules** or **Set Up and Maintain Banner EDW Security Rules**.
5. Click **Create**. The Create a New Security Rule page opens.
6. Enter the values for each field as described below.

Field	Description
Organizational Areas	This attribute enables you to group similar rules together for easier maintenance/assignment. You can grant access to entire sets of columns/tables at this level using a single check-box. Rules are delivered with four groupings. You can add more groupings using the “Set up and Maintain Organizational Areas” on page 4-19.
Table	Banner EDW table on which you want to secure data, for example, the MST_TEST table, the MPT_EMPL_EARN_FY table, etc.

Field	Description
Rule Type	The type of Security Rule. There are two possibilities: *Range:* This type of rule pertains to limits, such as Financial amounts. Results in a WHERE clause predicate like: WHERE COLUMN1 > [some value1] AND < [some value2]. *List:* This type of rule pertains to lists of valid values. Results in a WHERE clause predicate that matches up the list of allowed values (from the MGBFGAV table) with the values in the source table itself. **Note:** The Transfer Banner Fine-Grained Access process only uses security rules with rule type of “List”.
Column 1	Banner ODS or Banner EDW table column to which the rule pertains.
Query for Column 1	The PL/SQL SELECT statement used to populate the list of values in the Administrative UI for the specified Column 1 when assigning values to users. Click **Generate** to automatically create the PL/SQL statement. The base rules are delivered with simple SELECT DISTINCT queries for each of the columns on the various Banner ODS or Banner EDW tables. However, if performance becomes an issue (for the SELECT DISTINCTs to return), you can create temporary tables (manually) from the results of a SELECT DISTINCT query, then change this query to have the rule point to the temporary table instead. For two-column rules, select distinct values for both columns into a temporary table and then include select distinct statements for both query for Column1 and query for Column 2. ***Example*** You have a two-column rule for MFT_GENERAL_LEDGER where Column 1 is FUND and Column 2 is CHART_OF_ACCOUNTS. First create a table: CREATE TABLE *temp_table* as SELECT DISTINCT CHART_OF_ACCOUNTS, FUND FROM MFT_GENERAL_LEDGER. Then, in Query for Column 1 enter SELECT DISTINCT FUND FROM *temp_table* and in Query for Column 2 enter SELECT DISTINCT CHART_OF_ACCOUNTS FROM *temp_table.*

Field	Description
Column 2	An optional second column on the Banner ODS or Banner EDW table to which the rule pertains. This column can be used to join AND values together from two columns. **Note:** If you are creating or modifying rules that deal with Finance such as Fund, Organization, Account, Location, or Program you *must* enter the table's Chart of Accounts column name in the **Column 2** field. This is required because Banner ODS Finance hierarchy tables check to see if there are additional permissions for a given user, and that lookup on the hierarchy table cannot occur without a value for Chart of Accounts. ***Example*** You want to set up a security rule for the FUND column on MFT_GENERAL_LEDGER. You enter *FUND* into **Column 1** and CHART_OF_ACCOUNTS into **Column 2**. If you create that rule without CHART_OF_ACCOUNTS in **Column 2**, a user's permissions for General Ledger Funds are incomplete because the Transfer Banner Fine-Grained Access process and the Fine-Grained Access Policy package, MGKSECR, are not able to read the Fund hierarchy table, MFT_FUND_HIERARCHY. If a user has access to Fund 0100 for Chart of Accounts A, the Transfer Banner Fine-Grained Access process and MGKSECR can look up the Fund hierarchy record and determine if there are additional Fund codes related to Fund 0100 that this user should also have access to. Those additional Fund codes would be stored on the hierarchy record in Fund Level 1, 2, 3, 4, and 5.
Query for Column 2	The PL/SQL SELECT statement used to populate the list of values in the Administrative UI for the optional Column 2. Click **Generate** to automatically create the PL/SQL statement.

Field	Description
FGA Transfer Type	Select a value for this field if you plan to use the Transfer Banner Fine-Grained Access process to transfer security information from Banner to Banner ODS. • Finance Organization • Finance Fund • HR Organization • HR Employee Class A rule is excluded from the Transfer Banner Fine-Grained Access process if this column is blank. **Note:** A rule *must* contain a value in this field for the Transfer Banner Fine-Grained Access process to use the rule during the transfer. ***Example*** When the Transfer Banner Fine-Grained Access process transfers Finance Fund permissions into Banner ODS, it selects the rules from MGBFGAR that apply to the Finance Fund transfer. To include a rule in the Finance Fund part of the transfer process select *Finance Fund*.

Field	Description
Column 2 Type	When you create or modify a security rule that is used to apply security to an element of Finance, that rule must have a value for the Chart of Accounts in the **Column 2** field. In addition, Column 2 Type must contain the value *Chart of Accounts* which identifies the column 2 value as a Chart of Accounts value. ***Example*** If a rule is created to limit access to the **Fund** column on the General Ledger, you would select *FUND* as the Column 1 value, *CHART_OF_ACCOUNTS* as the column 2 value, and *Chart of Accounts* as the Column 2 Type. **Note:** It is obvious that the column 2 value is a Chart of Accounts value because the name of the column is CHART_OF_ACCOUNTS. This is not obvious for all Chart of Accounts column names. Some appear as DESG_CHART_OF_ACCOUNT on the MAT_GIFT table and HOME_ORGANIZATION_CHART on MPT_EMPLOYEE. The **Column 2 Type** field explicitly identifies a column 2 value as a Chart of Accounts column. If the rule's column 2 value is a Chart of Accounts column name during the Transfer Banner Fine-Grained Access process, then the Chart of Accounts value is brought over from Banner when the data is written to MGBFGAV.

Field	Description
Predicate Code	Leave this field blank for all rules that pertain to Finance Fund, Finance Organization, and Human Resources Organization to transfer file access permissions using the Transfer Banner Fine-Grained Access process. When the field is blank, the Transfer Banner Fine-Grained Access process writes additional data to MGBFGAV from the appropriate Banner ODS Finance Hierarchy table. ***Example*** If John Smith has access to Fund 0100, when the Predicate Code is blank on all of the Finance Fund security rules, the Transfer Banner Fine-Grained Access process reads the Banner ODS Finance Fund Hierarchy table, MFT_FUND_HIERARCHY, and determines if having access to fund 0100 also gives John access to additional Fund numbers. If the Fund Hierarchy entry for Fund 0100 includes a Fund Level 1 value of 0101 and a Fund Level 2 value of 0102, additional records are written to MGBFGAV giving John access to funds 0101 and 0102 as well as fund 0100. However, if you preferred to have the Fund Hierarchy values for funds 0101 and 0102 added to the query predicate at query runtime by means of a join to the Fund Hierarchy, you can add a Predicate Code of *Fund* to each of the Finance Fund security rules. **Note:** It is not recommended that you use the Predicate Code for Finance Fund, Finance Organization, or Human Resources Organization rules because adding the join to the Banner ODS applicable Hierarchy table at runtime can significantly impact query performance. Keep in mind that Banner Finance permissions are transferred only for Fund and Organization (whether they apply to Finance or HR tables). If you want to add security rules for other portions of Finance, (namely Account, Location, or Program), those permissions are not transferred from Banner. You need to create those security rules with a predicate code of *Account*, *Location*, or *Program* so that the additional values from the appropriate hierarchy table are included in the query predicate at runtime. However, the same potential performance warning applies for using predicate codes for those security rules. To resolve the performance issue you might consider adding records to MGBFGAV for rules pertaining to Account, Location, or Program.

7. Click **Save**.

Update or Delete a Security Rule

Perform all of these steps for each Security Rule you want to set up. Use the following steps to update an existing Security Rule.

1. Click **Preferences & Security** from the Administrative menu.
2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.
3. Click **Set Up and Maintain Banner ODS Security Rules** or **Set Up and Maintain Banner EDW Security Rules**.
4. From the drop-down list, choose the **organizational area**, **table**, and/or **column** for the rule you want to edit.
5. Click **Search**. The list of related Security Rules displays.
6. Click the link in the **Column 1** column for the rule you want to edit. The Update an Existing Rule page opens.
7. Edit **Query for Column 1, Query for Column 2**, **FGA Transfer Type**, **Column 2 Type**, and/or **Predicate Code**, then click **Save**.

   or

   Click **Delete** to remove the displayed security rule.

Note

The table at the bottom of the page indicates what rows in that table will be deleted if you delete the security rule. ■

Assign Security Rules

After security rules are created, you must determine what level of security each user requires. This is also where the rules are turned on and off.

Next, set up the security rules for users. You can use the Administrative UI to maintain the list of rules in the MGBFGAR table.

Note

The administrator account that you use to set up fine-grained access control needs to have unrestricted access to all data, or the list of values the administrator can grant to others is limited to what the administrator can access. ■

Use any of the following methods to secure user access:

- user name
- organizational area
- business profile
- element

Secure Access by User Name

Use the following steps to assign security by user name. This method also enables you to grant a user access to all data in the entire solution by checking a single checkbox.

1. Click **Preferences & Security** from the Administrative menu.

2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.

3. Click **Assign Security Rules**.

   The list of User IDs is determined by the IA_ADMIN.WAV_ALL_USERS view. This view contains a list of IDs for users most likely to run the reports. Your institution can change this view to include additional users (ODSMGR IA_ADMIN, for example) so that additional user IDs will appear in the list.

4. Check the **Access to all Banner EDW Data** check box for each user in which you want to assign access to all data from the Secure Banner EDW Access by User Name page.

   Each column is described below.

   Click the individual user’s name to restrict access to specific areas for that user. The link opens the Secure by Organizational Dimension page. See “Secure Access by User ID” on page 4-38 for steps on restricting a user’s access by Organizational Dimension.

Field	Description
Oracle User Name	Grouping of similar rules for easier maintenance/assignment. Rules are delivered with four groupings, but more groupings can be added in the MTVFGAA validation table, and can be used for new or existing rules. To restrict access for a specific user, click that user's user name. Organizational Dimension restrictions are made on the Secure Banner EDW Access by Organizational Dimension page. See "Secure Access by User ID" on page 4-38 to restrict a user's access.
Profiles	Set up an existing business profiles on the Create a Business Profile page. Click **Assign Profiles** to open the View Business Profiles and User Association page.
Access Level	The current level of access the user has to areas of information. To grant full access, check the checkbox in the **Access to all Banner ODS Data** or **Access to all Banner EDW Data** column. Possible values: *All* Green. Full access. *Partial* Yellow. Access to specified areas only. *None* Red. No access.
Access to all Banner ODS or Banner EDW Data	Check the checkbox to give the user unrestricted access to all areas and information. If the checkbox is checked, a Y is stored in the MGBSECR_FGA_ALL_IND column in the MGBSECR table. When the MGKSECR package is called from the policy, no predicate is returned. This allows access to all data.

5. Click **Save** to update the Administrative UI.

Secure Access by User ID

Use the following steps to assign security to an individual user.

1. Click **Preferences & Security** from the Administrative menu.
2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.
3. Click **Assign Security Rules**.
4. Check the **Access to all Banner ODS Data** or **Access to all Banner EDW Data** check box to grant the user unrestricted access to all information.
5. Click the user name to which you want to assign access.

   This page displays the security rules defined on the Set Up Banner ODS Security Rule or Set Up Banner EDW Security Rule page. The rules are grouped alphabetically by Organizational Dimension.

   Each column is described below:

Column	Description
Oracle User Name	Click **Select Another User** to open the Secure Access by User Name page.
Profiles	Existing Business Profiles set up on the Create a Business Profile page. Click **Assign Profiles** to open the View Banner ODS Business Profile and User Associations page
Access to All Banner ODS or Banner EDW Data	Check the checkbox to give the user unrestricted access to all areas and information. Click **Duplicate User** to open the Duplicate User Security Rules window.
Organizational Area	Area within the institution set up within the `IA_ADMIN.MTVFGAA` table.

Column	Description
Access All Data in this Area	Select the checkbox to grant the user security access to information within the corresponding organizational area. The list of areas is stored in the MTVFGAA table. You may change this list as desired. Rules can be grouped differently, for example. The All Data indicator for an area is stored in the MGBFGAA_ALL_IND in the MGBFGAA table. If the indicator is *Y* for a given table you are accessing, no predicate is returned from MGKSECR and you have full access.
Table	Banner ODS or Banner EDW table on which you want to secure data, for example, the MST_TEST table, the MPT_EMPL_EARN_FY table, etc. Click the link to enable or disable the security policies for that organizational area.

Column	Description
Element	Click an element to open the Secure Access by Element page. Elements can be set up as double or single column rule elements on the Create Security Rules page. ***Double Column Rules*** If a single rule was created that applies to two element columns, then both of the column names appear together in the **Element** column on the Secure Access by Organizational Dimension page, and are connected by an & (ampersand). This may be done when the user needs to see both pieces of the data in order to accurately understand the data. ***Single Column Rules*** A single column rule is when an element column was set up with a single column and a single rule.

Column	Description
Rule Type	The type of Security Rule. There are two possibilities: *Range:* This type of rule pertains to limits, such as Financial amounts. Results in a WHERE clause predicate like: WHERE COLUMN1 > [some value1] AND < [some value2] *List:* This type of rule pertains to lists of valid values. Results in a WHERE clause predicate that matches up the list of allowed values (from the MGBFGAV table) with the values in the source table itself.
Access Level	The level of security access assigned to the user. *All* Green. Full access. *Partial* Yellow. Access to specified areas only. *None* Red. No access.

6. To copy security access settings from one user or Business Profile to another, click **Duplicate User**. The Duplicate User Security Rules window opens.

   6.1. Choose the user(s) and Business Profiles(s) whose setting you want to merge, or duplicate. To choose more than one user or profile, hold down the Ctrl key while you continue to choose users or profiles.

   6.2. Use the radio buttons to indicate whether to merge current settings together, or replace one set of settings with another.

   6.3. Click **Duplicate** to save your settings, or **Cancel** to close the page.

7. Click **Save** at the bottom of the page to update the Administrative UI.

Secure Access by Business Profile

Use the following steps to assign security by Business Profiles.

1. Click **Preferences & Security** from the Administrative menu.

2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.

3. Click **Assign Security Rules**.

   A description of each field on the page appears below:

Field	Description
Business Profile	Existing Business Profiles set up on the Create a Business Profile page. Click a Business Profiles to open the Set Up Security Rules page.
Access Level	The level of security access assigned to the business profile. *All* Green. Full access. *Partial* Yellow. Access to specified areas only. *None* Red. No access.
Access to All Banner ODS or Banner EDW Data	Check the check box to give the business profile unrestricted access to all information.

4. Click the Business Profile to which you want to assign access from the Secure Banner ODS Access by Profile page. The Set Up Security Rules page opens.

   This page displays the security rules defined on the Set Up Banner ODS Security Rule page. The rules are grouped alphabetically by Organizational Area. Each column is described below:

Column	Description
Profile	Click **Select Another Profile** to open the Secure Access by Profile page.
Users	The Users associated with this Business Profile. Click **Assign Users** to open the View Banner ODS Business Profile and User Associations page.
Access to All Banner ODS Data	Check the checkbox to give the Business profile unrestricted access to all areas and information. Click **Duplicate User** to open the Duplicate User Security Rules window.
Organizational Area	Area within the institution set up within the `IA_ADMIN.MTVFGAA` table.

Column	Description
Access All Data in this Area	Select the checkbox to grant the Business Profile security access to information within the corresponding organizational area. The list of areas is stored in the `MTVFGAA` table. You may change this list as desired. Rules can be grouped differently, for example. The All Data indicator for an area is stored in the `MGBFGAA_ALL_IND` in the `MGBFGAA` table. If the indicator is *Y* for a given table you are accessing, no predicate is returned from MGKSECR and you have full access.
Table	Banner ODS table on which you want to secure data, for example, the `MST_TEST` table, the `MPT_EMPL_EARN_FY` table, etc. Click the link to enable or disable the security policies for that organizational area.
Element	Click an element to open the Secure Access by Element page. Elements can be set up as double or single column rule elements on the Create Security Rules page. ***Double Column Rules*** If a single rule was created that applies to two element columns, then both of the column names appear together in the **Element** column on the Secure Access by Organizational Dimension page, and are connected by an & (ampersand). This may be done when the user needs to see both pieces of the data in order to accurately understand the data. ***Single Column Rules*** A single column rule is when an element column was set up with a single column and a single rule.

Column	Description
Rule Type	The type of Security Rule. There are two possibilities: *Range:* This type of rule pertains to limits, such as Financial amounts. Results in a WHERE clause predicate like: `WHERE COLUMN1 > [some value1] AND < [some value2]` *List:* This type of rule pertains to lists of valid values. Results in a WHERE clause predicate that matches up the list of allowed values (from the `MGBFGAV` table) with the values in the source table itself
Access Level	The level of security access assigned to the user. *All* Green. Full access. *Partial* Yellow. Access to specified areas only. *None* Red. No access.

5. To copy security access settings from one user or Business Profile to another, click **Duplicate User**. The Duplicate User Security Rules window opens.

   5.1. Choose the user(s) and Business Profiles(s) whose setting you want to merge, or duplicate. To choose more than one user or profile, hold down the Ctrl key while you continue to choose users or profiles.

   5.2. Use the radio buttons to indicate whether to merge current settings together, or replace one set of settings with another.

   5.3. Click **Duplicate** to save your settings, or **Cancel** to close the page.

6. Click **Save** at the bottom of the page to update the Administrative UI.

Secure Access by Element

Use the following steps to assign security by element.

1. Click **Preferences & Security** from the Administrative menu.

2. Click **Set up Banner ODS Security Rules** or **Set Up Banner EDW Security Rules**.

3. Click **Assign Security Rules**.

4. If you wish to secure by element for a Business Profile, select **Secure By Profile.**

5. Depending on whether you are securing by User ID or by Business Profile, choose a name for Oracle User Name or Business Profile column. The Set Up Security Rules page opens.

   This page displays the security rules defined on the Set Up Banner ODS or Set Up Banner EDW Security Rules page. The rules are grouped alphabetically by Organizational Dimension.

6. Elements can be set up as double or single column rule elements on the Create Security Rules page.

Double Column Rules

If a single rule was created that applies to two element columns, then both of the column names appear together in the **Element** column on the Secure Access by Organizational Dimension page, and are connected by an & (ampersand). This is often done when the user needs to see both pieces of the data in order to accurately understand the data.

Single Column Rules

A single column rule is when an element column is set up with a single column and a single rule.

7. Choose the element to which you want to assign security for the user.

   From this page you can:

   - choose another element
   - assign profiles to the user/business profile to access all values for the element
   - copy user access to another user

   A description of each field appears below:

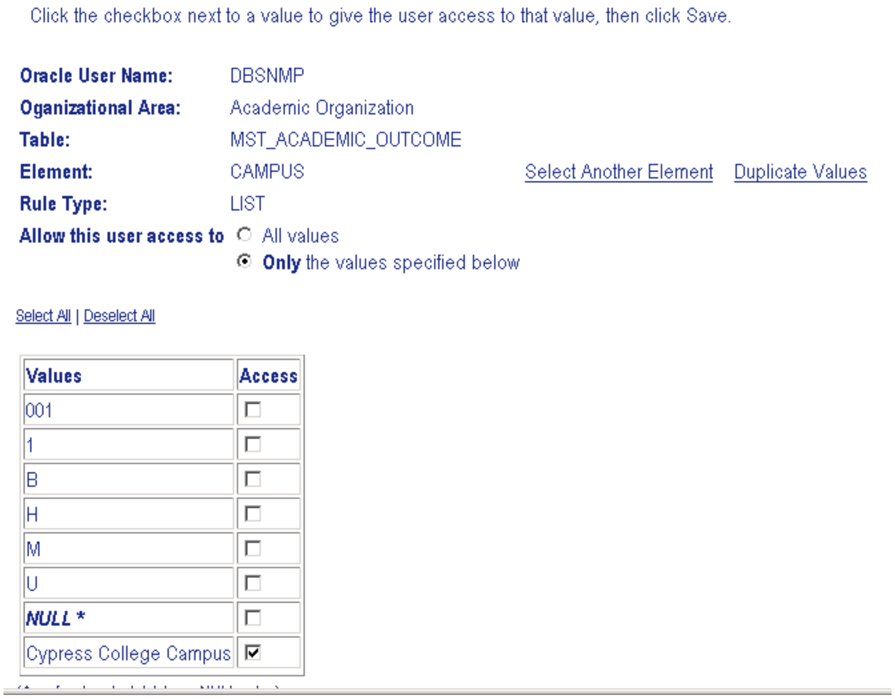
Field	Description
Oracle User Name	The user’s Oracle User ID. Grouping of similar rules for easier maintenance/assignment. Rules are delivered with four groupings, but more groupings can be added in the `MTVFGAA` validation table, and can be used for new or existing rules.
Organizational Area	Area within the institution set up within the `IA_ADMIN.MTVFGAA` table.

Field	Description
Table	Banner ODS or Banner EDW table on which you want to secure data, for example, the `MST_TEST` table, the `MPT_EMPL_EARN_FY` table, etc.
Element	Elements can be set up as double or single column rule elements on the Create Security Rules page. *Double Column Rules* If a single rule was created that applies to two element columns, then both of the column names appear together in the **Element** column on the Secure Access by Organizational Dimension page, and are connected by an & (ampersand). This is often done when the user needs to see both pieces of the data in order to accurately understand the data. *Single Column Rules* A single column rule is when an element column is set up with a single column and a single rule.

Field	Description
Rule Type	The type of Security Rule. There are two possibilities: *Range:* This type of rule pertains to limits, such as Financial amounts. Results in a WHERE clause predicate like: `WHERE COLUMN1 > [some value1] AND < [some value2]` *List:* This type of rule pertains to lists of valid values. Results in a WHERE clause predicate that matches up the list of allowed values (from the `MGBFGAV` table) with the values in the source table itself.
Allow this user/ profile Access to	Click the appropriate button: *All values:* The user is granted access to all values for this element, and is stored in the `MGBFGAE_ALL_IND` column as a *Y*. If new values are add, they will be considered accessible after the next refresh. *Only the values specified below:* Specify which values the user can access. If new values are added then they will not be considered accessible after the next refresh. Each new value needs to be checked individually. You can click **All Values**, which is then stored in the `MGBFGAE_ALL_IND` column as a *Y*, then the user or business profile is granted access to all values for this element. If you can choose **Only the values specified below,** then you can choose the specific values to which the user will have access (a la carte style). Those selected values are then stored in the `MGBFGAV` table.

8. Indicate whether you want to allow the user or business profile access to all values, or only the values that appear in the Values table below the **Allow this user (or profile) Access to** radio group.

   8.1. If you selected a single column rule element, then refer to the sample screen for a single column rule element below:



Click the checkbox next to a value to give the user access to that value, then click Save.

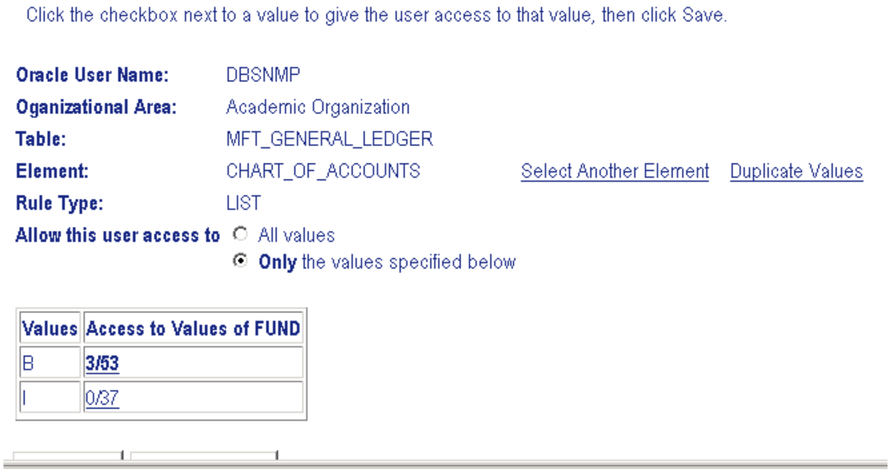
Oracle User Name:	DBSNMP
Oganizational Area:	Academic Organization
Table:	MST_ACADEMIC_OUTCOME
Element:	CAMPUS
Rule Type:	LIST
Allow this user access to	○ All values ● **Only** the values specified below

Values	Access
001	☐
1	☐
B	☐
H	☐
M	☐
U	☐
NULL *	☐
Cypress College Campus	☑

A description of each single column rule element column appears below:

Field	Description
Value	These values are set up in the validation tables in your source system. NULL indicates missing codes in your source system.
Access	Check the checkbox of the values to which you want to assign security for the selected user.

8.2. If you selected a double column rule element, then refer to the sample screen for a double column rule element below.



Click the checkbox next to a value to give the user access to that value, then click Save.

Oracle User Name: DBSNMP

Oganizational Area: Academic Organization

Table: MFT_GENERAL_LEDGER

Element: CHART_OF_ACCOUNTS

Rule Type: LIST

Allow this user access to ○ All values
● **Only** the values specified below

Values	Access to Values of FUND
B	3/53
I	0/37

A description of each column appears below

Column	Description
Value	These values are set up in the validation tables in your source system. NULL indicates missing codes in your source system.
Access to Values of (column name)	These values have one rule for two columns. The number to the right of the slash indicates the number of values in the column that have been assigned to the user. The second number indicates the total number of possible values available for that column. In the sample screen above, 2 out of 53 possible values have been assigned for the FUND column.

9. Click **Save** to keep your settings.

Security Predicates

An important, but optional, step in your implementation of Banner ODS Fine-grained Access is to review the delivered code in the MGKSECR PL/SQL package. This way you can review the delivered business logic, and determine if it is appropriate for your institution. You can also determine if there is any business logic that you might want to add.

If you encounter issues using the Security system, you might examine the security predicates that are generated. Enter the following query:

```
select mgksecr.f_check_ODS_fga('ODSMGR','MST_TEST') from dual;
```

```
Returns:
exists (select 'x' from mgbfgav where mgbfgav_username =
sys_context( 'userenv','session_user') and
mgbfgav_fgaa_code='ACAORG'and mgbfgav_column_name = 'TEST' and
NVL(mgbfgav_value,1) = NVL(TEST,1)) and exists(select 'x' from
mgbfgav where mgbfgav_username = sys_context(
'userenv','session_user') and mgbfgav_fgaa_code = 'ACAORG' and
mgbfgav_column_name = 'TEST_TYPE' and NVL(mgbfgav_value,1) =
NVL(TEST_TYPE,1))
```

Oracle produces a JOIN to the security tables for any columns that do not have the **All Data** indicator set. This allows the Oracle query optimizer to determine the fastest way to retrieve the data.

Policy Management

Typically, policies (and hence security) are either completely on or off. Two scripts are delivered with the Administrative UI to help manage the policies.

Prerequisites

- Create organizational areas
- Create user ID translations
- Create business profiles
- Create security rules

Policies for all Tables

To set up policies for all the tables that have security rules defined for them, run the following script:

```
sqlplus IA_ADMIN/<password> @create_all_fga_policies
```

Note

These scripts are delivered in the `dbscripts/utility_scripts` directory.

To remove all the policies from Banner ODS or Banner EDW tables, run:

```
sqlplus IA_ADMIN/<password> @drop_all_fga_policies
```

Note

These scripts add or drop Policies only for those tables with defined security rules. However, by default, security rules are not defined for all Banner ODS or Banner EDW tables. You should review the list of security rules in the Administrative UI to verify that all tables that you want to secure have rules defined. Since you only set up Policies for the tables with rules, any other tables remain unsecured. Remember, however, you can always update the security rules later, and then rerun the “drop” and “create” scripts to establish Policies as well. ■

Policies for a Single Table

Banner ODS is delivered with a script that can create a policy for a single table. This script enables you to independently test security access. Edit the script to supply the name of the table for which you want to create a policy, and then run the following:

```
sqlplus IA_ADMIN/<password> @create_fga_policy
```

Another way to enable a policy for a single table is available on the Assign Security Rules/set Up Security Rules pages of the Administrative UI. In the **Table** column is a link that is either set to **Policy Enabled**, or **Policy NOT Enabled**. Click the link to toggle between enabling or disabling the policy for a single table.

Example:

1. Create a new user to access Banner ODS - call the account BRUCE.

2. Use the `MST_TEST` table, and add nine rows using the following commands:

```
TRUNCATE TABLE ODSMGR.MST_TEST;
INSERT INTO ODSMGR.MST_TEST (TEST,TEST_TYPE) VALUES
('Test1','Type A');
INSERT INTO ODSMGR.MST_TEST (TEST,TEST_TYPE) VALUES
('Test1','Type A');
INSERT INTO ODSMGR.MST_TEST (TEST,TEST_TYPE) VALUES
('Test1','Type A');
INSERT INTO ODSMGR.MST_TEST (TEST,TEST_TYPE) VALUES
('Test2','Type A');
INSERT INTO ODSMGR.MST_TEST (TEST,TEST_TYPE) VALUES
('Test2','Type A');
INSERT INTO ODSMGR.MST_TEST (TEST,TEST_TYPE) VALUES
('Test2','Type A');
INSERT INTO ODSMGR.MST_TEST (TEST,TEST_TYPE) VALUES
('Test3','Type A');
INSERT INTO ODSMGR.MST_TEST (TEST) VALUES ('Test3');
INSERT INTO ODSMGR.MST_TEST (TEST) VALUES ('Test3');
COMMIT;
```

Note

The last two rows have a NULL value for TEST_TYPE. ■

Banner ODS does not have any policies in place when it is delivered. If the user BRUCE has been granted SELECT access to the `MST_TEST` table, you can execute the following query:

```
SQL> select count(*) from odsmgr.mst_test;
COUNT(*)
________
      9
```

3. Apply the policy to this table (from the `IA_ADMIN` user account):

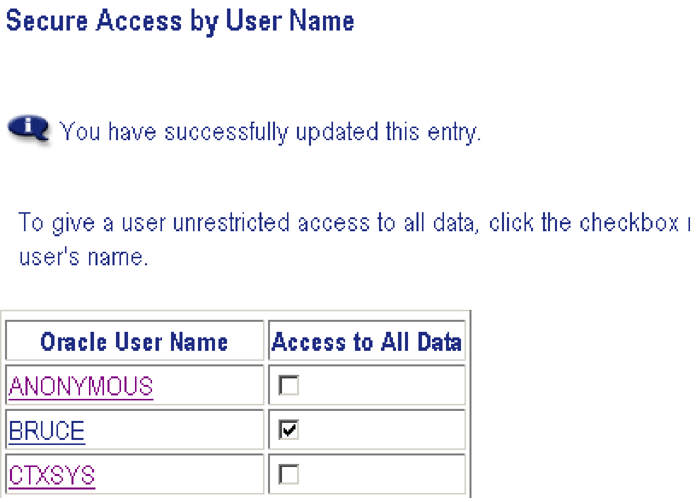
```
SQL> set serveroutput on size 50000;
SQL> exec mgkutil.p_createFGAPolicy('ODSMGR','MST_TEST',1);
Policy added to table: MST_TEST
PL/SQL procedure successfully completed.
```

4. Run the BRUCE query again.The following appears:

```
SQL> select count(*) from odsmgr.mst_test;
COUNT(*)
----------------------
      0
```

Look in the Administrative UI Security. The BRUCE account is displayed with no global access.

5. Select the **All Data** checkbox, and rerun the query. The following appears:



Secure Access by User Name

You have successfully updated this entry.

To give a user unrestricted access to all data, click the checkbox ı
user's name.

Oracle User Name	Access to All Data
ANONYMOUS	☐
BRUCE	☑
CTXSYS	☐

```
SQL> select count(*) from odsmgr.mst_test;
```

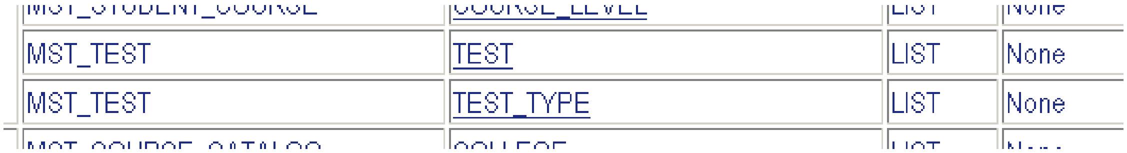
```
COUNT(*)
----------------------
9
```

6. Clear the **All Values**.

7. Click the **Save**.

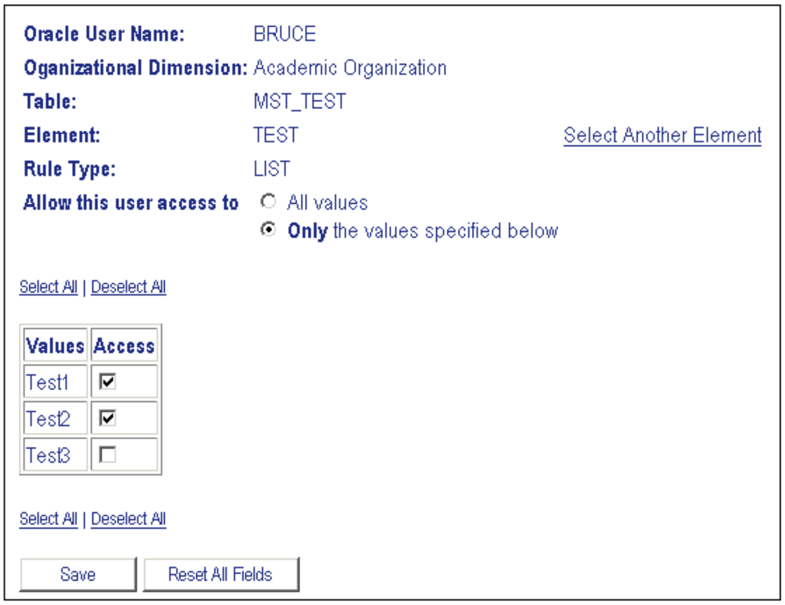
8. Choose the BRUCE account.

   To duplicate these results check/uncheck the **Access All Data in This Area** checkbox for the Academic Organization. To continue to test this, choose a combination of values for the two columns in the `MST_TEST` table, namely:



MST_STUDENT_COURSE	COURSE_LEVEL	LIST	None
MST_TEST	TEST	LIST	None
MST_TEST	TEST_TYPE	LIST	None
MST_COURSE_CATALOG	COLLEGE	LIST	None

9. Enable the first two values of the TEST element as follows:



Oracle User Name: BRUCE
Oganizational Dimension: Academic Organization
Table: MST_TEST
Element: TEST
Rule Type: LIST
Allow this user access to ○ All values
● **Only** the values specified below

Values	Access
Test1	☑
Test2	☑
Test3	□

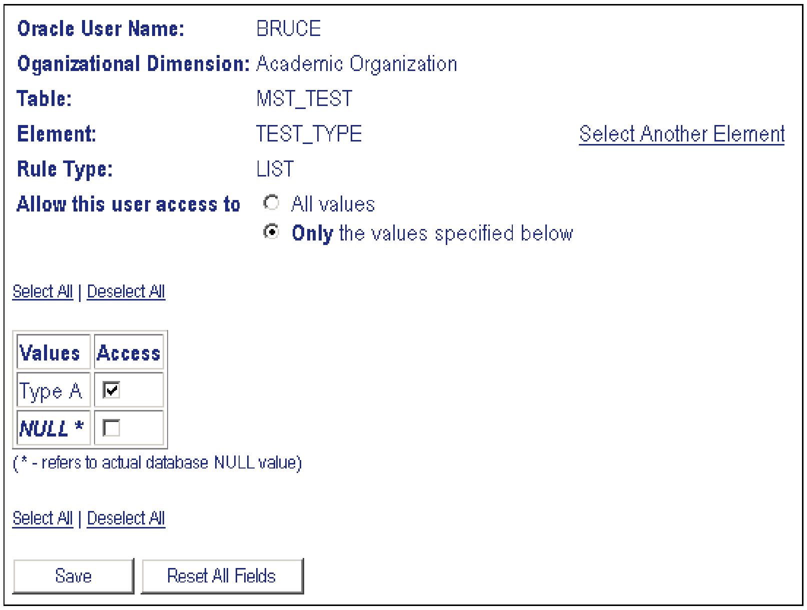
And yet:

```
SQL> select count(*) from odsmgr.mst_test;

COUNT(*)
------------------------
 0
```

Security rules are cumulative. Users must have access to values across *all* columns/ rules for a given table in order to access the data.

10. Update the TEST_TYPE element as follows:



Oracle User Name:	BRUCE
Oganizational Dimension:	Academic Organization
Table:	MST_TEST
Element:	TEST_TYPE
Rule Type:	LIST
Allow this user access to	○ All values ● **Only** the values specified below

Values	Access
Type A	☑
NULL *	□

(* - refers to actual database NULL value)

The expected results are:

```
SQL> select count(*) from odsmgr.mst_test;

COUNT(*)
--------------------
6
```

You can continue to test security using the Administrative UI, and see the results from queries that are run against the system.

Administrative User Interface Data Access

Once policies are in place, you control all access to tables using the information in the security (MGBFGA*) tables. You might wonder how can the Administrative UI issue the SELECT DISTINCT queries to retrieve the list of values? Shouldn't they need to be configured in the Security Tables also? Does the user account used by the web or application server have some kind of back door around the security system? The answer is, yes and no. As part of the Policy/FGA security system, Oracle provides a way to selectively bypass security using application context variables. You can create a context that is associated with a particular package that has permission to set application context values. This can then be retrieved by other parts of the application.

In practice, this means you can create a context called IA_FGA and associate it with the Administrative UI (MGKFGAC) package. In that package, you can set a context variable prior to making queries to the tables. Then, when Oracle calls the MGKSECR package to enforce the policy, it checks that the context variable exists, and sensing it, returns no predicate. This allows full access to the data in that table. The context variable only exists for the life of the package (in the application server memory) and can be accessed only by that package. So, no other attempts to access the context are allowed. This allows the Administrative UI to maintain complete access to administer security while keeping security in place for all other access attempts. (For more information on using Application Context for security, see the Oracle Database Security Guide)

Set up and Synchronize Data (Banner ODS)

Maintaining current data in Banner ODS is key to producing accurate reports. Banner ODS uses programs—Oracle Warehouse Builder (OWB) mappings—to associate elements in the administrative system with their corresponding elements in Banner ODS. When you run a job (schedule a process via the Administrative UI), it calls the related mappings and loads or updates the data defined by them.

Banner ODS includes two main categories of mappings:

- LOAD mappings—load data from the administrative system into Banner ODS. These mapping names include a “LOAD_” prefix.
- REFRESH mappings—update Banner ODS with data that has changed in the administrative system. Mappings in this category have an “UPDATE_” or “DELETE_” prefix. Typically, these mappings exist in pairs. To perform a complete refresh, you run the DELETE mapping followed by its associated UPDATE mapping.

Banner ODS is delivered with hundreds of mappings already defined. LOAD and REFRESH mappings exist for each composite table in Banner ODS. To make it easier to work with the mappings, they are organized into groups by product area. This gives you the ability to run one job that includes a group of mappings at one time. (For example, Finance-related mappings.) Or, you can run a single mapping.

Banner ODS exists in a self-contained environment separate from your source system. You synchronize data between the two systems using the processes that load and refresh data in Banner ODS. Even with daily synchronization, you can expect minor differences between the two systems. Three main reasons that differences exist are:

- Data currency in Banner ODS is dependent on the timing of a query against Banner ODS, and when Banner ODS was last refreshed. Changes that occur in the administrative system after the last refresh are not reflected until the next refresh occurs. This causes a variance between the two systems until Banner ODS is refreshed again.
- Display rules may differ between the two systems. Some display rules are loaded form the Banner GTVSDAX table into the ODS MGRSDAX table. If these rules are not maintained in both places or updated in the ODS using the script that loads data from GTVSDAX to MGRSDAX, they will be out of sync.
- drive Banner ODS views created to support existing Object:Access functionality. Differences may occur based on which rules are applied to each system.
- Security rules may also cause differences between the two systems. Your source system allows you to set up fine-grained access security at the element level as does Banner ODS. Rules in both systems are discrete, so there may be differences in the data a user can view based on the security rules defined within each system.

It is important to keep in mind these possible differences while reporting against Banner ODS.

- When you first install Banner ODS, populate it with data from your source system by running the “Load All Banner ODS Products” job
- Refresh data in Banner ODS on a regular basis by scheduling jobs that update Banner ODS each night
- Update specific areas of Banner ODS as needed by scheduling that job when data is changed in the source system

Set up and Synchronize Data (Banner EDW)

The Banner EDW stores data that is fed to it from the Banner ODS. Use the Administrative UI to schedule jobs, which run OWB mappings that take information from Banner ODS and load it into the warehouse.

After cleansing and MTVPARM records have been finalized, you can populate the Banner EDW using the Banner EDW Load Processes. To populate the Banner EDW, you run ETL jobs from the Administrative UI. There are two menu groups for scheduling EDW process/jobs - ‘Schedule Banner EDW Snapshot Mappings’ and ‘Schedule Banner EDW Operational Mappings’. The EDW operational mappings will store all data relative to the selected star. The EDW snapshot stars are event based and can be optionally loaded if it suits your business needs

Each job requires you to select certain runtime parameters. Once the job is submitted, Oracle Warehouse Builder (OWB) mappings take information from Banner ODS to Banner EDW using the following process.

1. The mappings extract the appropriate Banner ODS data based on the parameters selected.

2. The data runs through cleansing setting the Banner EDW codes and short and long descriptions.

3. The dimension and fact tables are loaded.

You can monitor the progress of the jobs from the View Control Reports page under the Options menu in the Administrative UI. If there are cleansing errors, the codes which could not be translated will be listed in the control report. Modify the cleansing code translation and description records to accommodate for those new values and then either re-run the job, replacing the event or run the Fix process for that star.

A number of pieces of information are defined to perform internal functions. These pieces of information are called parameters, and are stored in the Parameter Table (MTVPARM) in the Banner EDW. If Banner EDW requires one of these pieces of information while running a process or displaying an administrative web page, it goes to MTVPARM and gets the needed information. Before you populate the Banner EDW, you need to review and edit the parameters related to these tasks.

Follow the steps below to populate the Banner EDW:

1. Click **Options** from the Administrative UI menu.

2. Click **Schedule a Process**. The Select a Process page opens.

3. Click **Schedule Banner EDW Snapshot Mappings** or **Schedule Banner EDW Operational Mappings** from the Select a Process page. The Select a Subprocess page opens.

4. Click the mappings group that corresponds to the star(s) you want to load. The Schedule a Process page opens.

5. Enter values for other Process Parameters, if any exist, for the selected process.

For instructions on how to set up process parameters, refer to the “Schedule a Process Parameter” section.

Set up Parameters (Banner ODS and Banner EDW)

Parameters that are delivered with your solution are stored in a table called MTVPARM. You can use the Administrative UI to view and modify the entries in MTVPARM, and to customize Banner ODS and Banner EDW, and the Administrative UI. (Example customizations: Schedule a process, define mappings that move data from the source system, define data cleansing, freeze data, publishing meta data, etc. See "Set up Customized Scheduled Processes" on page 4-77 for additional information.)

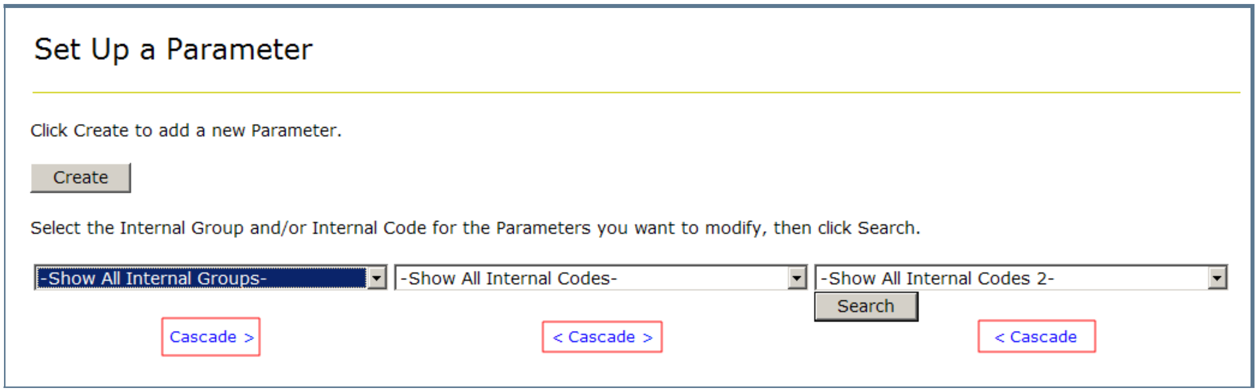
Note

These parameters are different from the actual runtime parameters that you supply when you schedule a process (run the mappings). (See "Schedule a Process Parameters (Banner ODS and Banner EDW)" on page 4-84.) The parameters discussed in this section are internal parameters that are used in internal processing.

A parameter can include multiple values. The values for a single parameter all use the same Internal Code. You use the Internal Code to choose a parameter to edit. Parameters are edited on the Set Up a Parameter page of the Administrative UI.

Cascade filter

The Cascade links on the Set Up a Parameter page let you filter the related field values when you select them. When you make a selection from any of the dropdown lists on the page then click the related Cascade link, the other lists will filter to display only related values.



Set Up a Parameter

Click Create to add a new Parameter.

Select the Internal Group and/or Internal Code for the Parameters you want to modify, then click Search.

-Show All Internal Groups- | -Show All Internal Codes- | -Show All Internal Codes 2-

For example, select a value in the Internal Groups dropdown list, then click the Cascade > link to filter the Internal Code and Internal Code 2 dropdown lists to display only the values related to the select Internal Group.

Create a parameter

Use the following steps to create a parameter entry.

1. Click **Options** from the Administrative UI menu. The Options menu opens.
2. Click **Set Up Parameters**. The Set Up a Parameter page opens.
3. Click **Create** from the Set Up a Parameter page, or click **Duplicate** from the Update an Existing Parameter page. The Create a New Parameter page opens.
4. Enter the information for the new parameter. A description of each field, followed by an example, appears below:

Field	Description
Internal Group	Rows of data with varying Internal Codes that are categorized together to provide multiple entries for one parameter.
Internal Code 1	Parameter values. Related values have the same Internal Code 1.
Internal Code 2	Used in combination with Internal Code 1 to further define the parameter values when the values in Internal Code 1 are not unique. Often this field is not used.
Internal Code Sequence Number	Order in which multiple rows of data appear within their parameter group. For parameters that are used to create a list, it specifies the order in which the values will appear in that list.
External Code	Short description of the parameter value for the related Internal Code. Also used as a Yes/No value indicator in some parameters.
Description	Long description of the parameter value for the related Internal Code.
System Required?	Yes or No. Indicates whether the field is required for production processing.

5. Click **Save** to create the new parameter.

Example: Event parameter

When you freeze data, you must specify an event so that the process knows where to load the new information. The Event parameter is used to define EVENT codes that are used for freezing data.

The Internal Group value is EVENT. It's used to identify all of the values for the Event parameter.

Internal Code 1 defines the various areas within Banner ODS that require different event definitions. It includes all the Subprocess values used to freeze data

Internal Code 2 defines each different event related to the areas defined by Internal Code 1. The values in this field are the valid values you can enter in the Event Code field.

The Internal Code Sequence is used to order parameter values that fall within the same area defined by Internal Code 1.

Update or Delete a Parameter

Use the following steps to change or delete an existing parameter.

1. Click **Options** from the Administrative UI menu. The Options menu opens.
2. Click **Set Up Parameters**. The Set Up a Parameter page opens.
3. From the **Show All Internal Groups** drop-down list on the Set Up a Parameter page, choose the Internal Group and Internal Code name of the parameter you want to access. Or, keep the default setting to show all Internal Groups or Internal Codes.

Tip

If you know the first letter of the Internal Group or Code you want to choose, open the **Show all Internal Groups** (or **Codes**) drop-down list then type the first letter of the group or code. Your cursor will move to the first group or code in the list that begins with that letter. This saves you from scrolling through the entire list. ■

4. Click **Search**. The Select an Existing Parameter page opens.
5. Click the description link that corresponds to the parameter entry you want to update or delete. The Update an Existing Parameter page opens.
6. Change the information as needed.

Note

Only External Codes less than 80 characters in length display in the drop-down list. You can create entries that are longer than 80 characters, and they will exist in the system, but do not appear in the list. ■

7. Click **Save**, to save the parameter, or **Delete** to completely remove the parameter.

System Parameters

Your solution is delivered with values that define aspects of your solution. Below are the delivered system parameters, and how they are used. Additional information can be found in the section “Schedule a Process (Banner ODS and Banner EDW)” on page 4-65.

Note

The parameters listed below are delivered with Banner ODS, followed by an additional list of parameters that are also delivered with Banner EDW. For a list of parameters used only to schedule a process, see “Schedule a Process Parameters (Banner ODS and Banner EDW)” on page 4-84. ■

Parameters *(Banner ODS and Banner EDW)*

This Parameter . . .	Used for this Task and Solution . . .	Does This . . .
ADMIN_PREFERENCES	Administrative UI for Banner ODS and Banner EDW	Optional parameters. These are various settings used to control aspects of the Administrative UI. Currently can be used to control the number of Control Reports that are displayed on the main selection page.
BANNER TO ODS FGA TRANSFER	“Schedule a Process (Banner ODS and Banner EDW)” on page 4-65	Transfers security for Banner Finance Fund, Fund Type, and Organizations, and Banner Human Resources Organizations and Employee Class.
ETL CONTROL GROUP	“Schedule a Process (Banner ODS and Banner EDW)” on page 4-65	Groups together ETL MAP PACKAGE and/or ETL SLOT PACKAGE jobs as one job.
ETL MAP PACKAGE	“Schedule a Process (Banner ODS and Banner EDW)” on page 4-65	Groups related jobs (OWB mappings) as one job.
ETL MAP PACKAGE LOAD PURGE	“Schedule a Process (Banner ODS and Banner EDW)” on page 4-65	Identifies the required crosswalk DELETE mappings for the Load Purge Process.

This Parameter . . .	Used for this Task and Solution . . .	Does This . . .
ETL MAP PACKAGE LOGIC	"Schedule a Process (Banner ODS and Banner EDW)" on page 4-65	Allows you to specify job termination logic for a mapping within a job stream. By default, all mappings in a job run in sequence regardless of whether they have errors or not. By defining an ETL Map Package record for a given mapping in a job, you can have the job stop if that mapping encounters errors. This parameter is used primarily with Banner EDW jobs as they have dependencies from one step (or mapping) to another, while Banner ODS mappings are independent of each other.
ETL MAP PACKAGE RECONCILE LOGIC	"Reconcile a Single Table" on page 3-21 and "Reconcile Multiple Tables" on page 3-22	Provides a list of mappings that are exceptions in the reconcile Banner ODS tables process. In this list are the mappings that are ignored in the reconcile process because of the complexity of the mapping or other factors outside the scope of reconciling that Banner ODS table. This list also includes mappings that require either multiple source composite views or mappings in order to reconcile a Banner ODS table.
ETL SLOT PACKAGE	"Schedule a Process (Banner ODS and Banner EDW)" on page 4-65	Groups together related slot jobs (SQL packages) as one job.
EVENT	"Freeze Data Maintenance (Banner ODS and Banner EDW)" on page 4-135	Defines EVENT codes used for freezing Banner ODS data.
EVENT-EDW	"Freeze Data Maintenance (Banner ODS and Banner EDW)" on page 4-135	Defines the Event parameter for freezing EDW business concepts.

This Parameter . . .	Used for this Task and Solution . . .	Does This . . .
INSTALLED PROCESS	“Schedule a Process (Banner ODS and Banner EDW)” on page 4-65	Populates a list of processes displayed on the Select a Process page.
JOB	“Schedule a Process (Banner ODS and Banner EDW)” on page 4-65	Defines the actual name of the job (program) to run when you schedule a process.
JOB INTERVAL	“Schedule a Process (Banner ODS and Banner EDW)” on page 4-65	Defines the list of sample Job Interval settings displayed in the Select an Interval window on the Schedule a Process page.
JOB_KILLER	“Kill a Running Job/Process” on page 4-74	Defines which administrative accounts have the ability to stop a process that is running.
JOB_NOTIFICATION	“Set up E-mail Notification (Banner ODS and Banner EDW)” on page 4-133	Defines a list of process parameters you need to set up e-mail notification.
METADATA	Meta Data publishing. “Meta Data (Banner ODS and Banner EDW)” on page 4-148	Defines meta data related settings. Currently there is one for where to publish Meta Data pages, and another for where to view them.
ODS FINANCE TEXT	Finance Reporting Text Views	Defines different types of text for Finance Reporting Text Views. For example, Encumbrance Text, Grant Text, Fund Text, Fixed Asset Text, etc.
OWB_SYSTEM_PARAMETER	“Schedule a Process (Banner ODS and Banner EDW)” on page 4-65	Defines the list of known OWB system – used when running mappings, to differentiate which mapping parameters are passed to OWB specifically, and which are passed to the mapping itself.
PARAMETER	“Schedule a Process (Banner ODS and Banner EDW)” on page 4-65	Defines a list of a job’s input parameters you need to supply when you schedule a process.

This Parameter . . .	Used for this Task and Solution . . .	Does This . . .
PUBL_CATE_CODE	“Schedule a Process (Banner ODS and Banner EDW)” on page 4-65	Used during meta data publishing to differentiate the source from the target types.
SUBPROCESS	“Schedule a Process (Banner ODS and Banner EDW)” on page 4-65	Populates a list of processes displayed on the Select a Subprocess page.

Parameters (Banner EDW)

This Parameter . . .	Used for this . . .	Does This . . .
CLEANSING DATA ELEMENT	“Set Up Cleansing” on page 2-66	The list of data elements used in Cleansing. The elements listed here are what show up in the drop-down lists for cleansing description and code value screens.
CLEANSING DEFAULT VALUES	“Set Up Cleansing” on page 2-66	Define the value, and long and short descriptions, used in Banner EDW for NULL and BAD (i.e. value not found in cleansing values list) cleansing values.
CLEANSING SOURCES	“Set Up Cleansing” on page 2-66	Defines the sources used for cleansing (Banner EDW) and for Freezing data (Banner ODS).
EDW EXTRACT PARAMETERS	“Define EDW Extract Parameters” on page 2-16	Controls how certain Banner EDW extracts operate when moving information from Banner ODS to Banner EDW.
EVENT-EDW	“List Events for a Banner EDW Star (Banner EDW)” on page 4-108	Determines which columns to exclude from the List Events for a Banner EDW Star page.
PARAMETER MAP	“Set up Parameter Maps” on page 2-102	Defines the values for parameter maps used to identify particular pieces of data in Cognos.

Schedule a Process (Banner ODS and Banner EDW)

You can schedule a job to run at a specific time. To run load and refresh (update) jobs, select the **Schedule a Process** option on the Options menu of the Administrative UI.

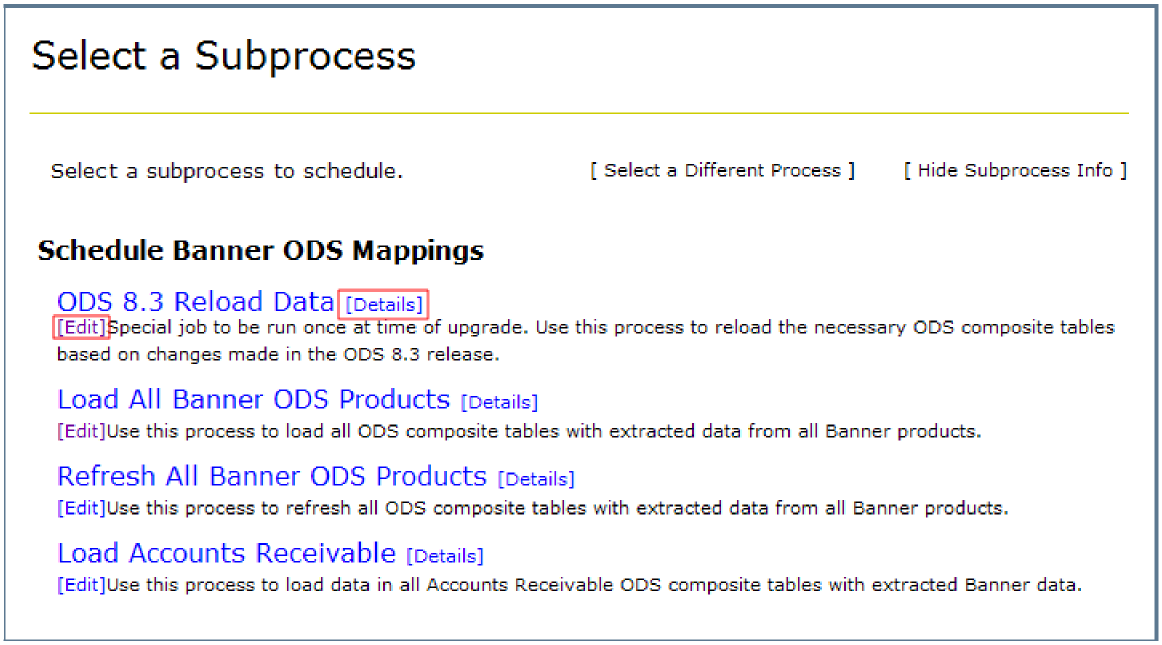
Before you schedule any jobs to run, you *must* review and set up parameters associated with scheduling a process. See "Set up Parameters (Banner ODS and Banner EDW)" on page 4-58 for more details.

Process descriptions and details

Click the [Show Process Info] or [Show Subprocess Info] links to view a description of each job on the Select a Process or Select a Subprocess pages. Similarly, click the [Hide Process Info] or [Hide Subprocess Info] links to view only the job names.

Refer to the "PROCESS INFO Parameter" on page 4-87 and "SUBPROCESS INFO Parameter" on page 4-91 sections for details on how to define the descriptions that display for a process or subprocess.

If you choose to Show Process Info on the Select a Subprocess page, [Details] and [Edit] links display for each process in the list as illustrated in the following picture.



Select a Subprocess

Select a subprocess to schedule.

Schedule Banner ODS Mappings

ODS 8.3 Reload Data [Details]
[Edit]Special job to be run once at time of upgrade. Use this process to reload the necessary ODS composite tables based on changes made in the ODS 8.3 release.

Load All Banner ODS Products [Details]
[Edit]Use this process to load all ODS composite tables with extracted data from all Banner products.

Refresh All Banner ODS Products [Details]
[Edit]Use this process to refresh all ODS composite tables with extracted data from all Banner products.

Load Accounts Receivable [Details]
[Edit]Use this process to load data in all Accounts Receivable ODS composite tables with extracted Banner data.

Details

Click the [Details] link next to a process name to view the job details including the title, code, description, list of all processes included in that job, and the Banner ODS tables or Banner EDW Fact tables that get loaded by the job.

The ADMIN_PREFERENCES parameter lets you define whether the details of a process, when you select to view them on the Schedule a Subprocess page, display in:

- One popup window which refreshes each time you display the details of a process
- Multiple popup windows, one for each process whose details you want to display

The Parameter record with the following combination of values:

- **Internal Group** = `ADMIN_PREFERENCES`
- **Internal Code 1** = `SCHEDULE_UI`
- **Internal Code 2** = `DETAILS_DIALOG`

controls whether to use one or multiple popup windows when displaying process details. Set the **External Code** = *0* for this Parameter record to allow multiple popup windows or set the **External Code** = *1* to use a single popup window.

Edit

Click the [Edit] link next to a process description to go to the Update a Parameter page and edit the process description.

The ADMIN_PREFERENCES parameter lets you define whether to display the [Edit] link next to each process or subprocess description.

The Parameter record with the following combination of values:

- **Internal Group** = `ADMIN_PREFERENCES`
- **Internal Code 1** = `SCHEDULE_UI`
- **Internal Code 2** = `DISPLAY_EDIT_LINK`

controls whether to display the [Edit] link. Set the **External Code** = *1* for this Parameter record to display the [Edit] link next to each description or set the **External Code** = *0* to not display the link.

Banner ODS Processes

The following sections detail the Banner ODS processes that you can run.

Schedule Banner ODS Mappings

Use the options on this menu to load or update the corresponding data into all Banner ODS composite and slotted tables.

Banner ODS Utilities

Use the options on this menu to report source change table counts, reconcile tables, add comments to reporting views, and run checks and balances.

Banner EDW Processes

The following sections detail the Banner EDW processes that you can run.

Schedule Banner EDW Snapshot Mappings

Use the options on this menu to schedule the mappings that load the corresponding snapshot star. Use the Fix options to run all mappings that load the corresponding star, but use the cleansing error table as input.

Schedule Banner EDW Operational Mappings

Use the options on this menu to run the mappings that load or refresh the corresponding operational star.

Banner EDW Utilities

Use the options on this menu to run a checks and balance report to assess the state of Banner EDW and to load default cleansing values into the original and enhanced warehouse structures.

Schedule Banner EDW Complete Load/Refresh

Use this option to load or refresh the enhanced warehouse structures, specifically all of the advancement-related data, from your source system into the fact, dimension, group, and aggregate tables. This job only displays if you have installed the enhanced warehouse structures. If Banner Advancement is your source system, the option name includes “Banner EDW”. If Advance is your source system, the option name includes “Advance EDW”.

Schedule [Banner EDW/Advance EDW] Dimensional Components

Use the options on this menu to run the mappings that load or refresh the corresponding dimension tables of the enhanced warehouse structures, specifically the advancement-related data dimension tables. This option only displays if you have installed the enhanced warehouse structures. If Banner Advancement is your source system, the option name includes “Banner EDW”. If Advance is your source system, the option name includes “Advance EDW”.

Schedule [Banner EDW/Advance EDW] Facts

Use the options on this menu to run the mappings that load or refresh the corresponding fact tables of the enhanced warehouse structures, specifically the advancement-related fact tables. This option only displays if you have installed the enhanced warehouse structures. If Banner Advancement is your source system, the option name includes “Banner EDW”. If Advance is your source system, the option name includes “Advance EDW”.

Schedule [Banner EDW/Advance EDW] Groups and Aggregates

Use the options on this menu to run the mappings that allow you to load the group and aggregate tables of the enhanced warehouse structures, specifically the advancement-related data group and aggregate tables. This option only displays if you have installed the enhanced warehouse structures. If Banner Advancement is your source system, the option name includes “Banner EDW”. If Advance is your source system, the option name includes “Advance EDW”.

Schedule [Banner EDW/Advance EDW] Freeze Aggregates

Use the options on this menu to run the mappings that allow you to freeze aggregate tables into their associated freeze tables.This option only displays if you have installed the enhanced warehouse structures. If Banner Advancement is your source system, the option name includes “Banner EDW”. If Advance is your source system, the option name includes “Advance EDW”.

Schedule Enrollment Management Mappings

Use the options on this menu to run all mappings that let you load and refresh components and stars of the warehouse that support the data stored for enrollment management.

Freeze Multiple Banner ODS Tables/Views

Use this option to freeze multiple Banner ODS composite tables or reporting views that have been defined by your institution.

Freeze a Single Banner ODS Table/View

Use this option to freeze a single Banner ODS composite table or reporting view.

Publish Meta Data Reports

Use this option to create the meta data html pages for end user access to meta data information specified in the institutional preferences.

Schedule a Single Process

Use the following steps to schedule when you want a single process to run:

Note

You can only run a single Banner ODS mapping,-not a single Banner EDW mapping. ■

1. Click **Options** from the Administrative UI menu.

2. Click **Schedule a Process**. The Select a Process page opens.

3. Choose the type of process you want to schedule to run from the Select a Process page.

   If you chose Schedule Banner ODS Mappings or Freeze Multiple Banner ODS Tables/ Views, then the Select a Subprocess page opens. Continue to the next step.

   All other selections open the Schedule a Process page. Skip to step #4.

4. Choose the subprocess you want to run. The Schedule a Process page opens.

5. If you selected the subprocess **Run a Single** Banner ODS **Mapping**, choose the mapping from the **Mapping to Run** drop-down list.

6. Enter values for other Process Parameters for the selected process, if any exist.

7. Enter the required Scheduling Parameters information.

   7.1. Enter a **Run Date** (format dd-mon-yyyy) and **Runtime** (format hh24:mi:ss).

   7.2. If you want to run the process on a recurring basis, enter an **Interval**.

   Click the link next to the **Interval** field. A sample Interval window opens. Click the link under the **Interval Expression** column for the interval in which you want to schedule a process. For example, to run a process every day at the same time select *SYSDATE+1*.

8. Click **Save** to save the information about this job. The job is entered into the job queue to run at the specified day and time.

Schedule Multiple Processes

You can schedule and list multiple processes with different parameters as a group. For example, if you want to run multiple Banner ODS Freeze Tables.

To create a multiple process schedule, you must export the definition of each desired single process (including all related parameters) to a comma separated values (.csv) file. You can then use that information to define/copy multiple job definitions in that file into a single master schedule which is then re-imported into the job queue.

To schedule multiple processes:

1. From the Administrative UI menu, click **Options**.

2. Click **Schedule a Process**. The Select a Process page opens.

3. From the Select a Process page, choose the type of process you want to schedule.

   If you chose Schedule Banner ODS Mappings, Banner ODS Utilities, Banner EDW Utilities, or Freeze Multiple Banner ODS Tables/Views, then the Select a Subprocess page opens. Continue to the next step below.

   For all other selections, the Schedule a Process page opens. Skip to step 5.

4. Choose a subprocess. The Schedule a Process page opens.

5. If you selected the subprocess **Run a Single** Banner ODS or Banner EDW **Mapping**, choose the mapping from the **Mapping to Run** drop-down list.

6. To open the.csv file, click **Export**.

   You can either open the file directly, or save it to another directory and open it from there.

   The columns names in the .csv file are described below:

Column	Description
JOBDEF	A constant for parsing the input data.
DATE	Date the job should run. Use MON-DD-YYYY format.
TIME	Time the job should run. Use H:MM:SS format.
PROCESS and SUBPROCESS	Internal identifiers for the job.
(Additional job specific parameters)	Any job-specific parameters such as Event, Source Institution, etc. For job-specific parameters that use drop-down lists of allowable values, all possible values for those fields are provided in the export download so that they can be copied when setting up the job records to import.

⚠ **Warning**

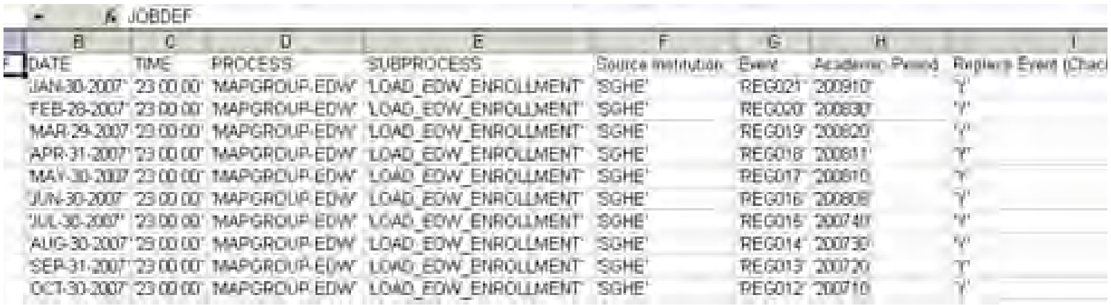
You *must* retain the formatting of each field in the .csv file. Each field is surrounded by single quotes. These must be retained for the import to parse the data correctly. Microsoft Excel sometimes strips a leading single quote from the contents of a cell, so you *must* be sure it is retained in the .csv output. You may want to use an alternate editing application, although Microsoft Excel works fine as long as you are careful. ■

7. Duplicate the **JOB** line once for each run desired.

8. Enter the date and time you want the process to run.

9. Enter the desired parameter values for each line.

10. Remove extra values in the additional lines. An example resulting .csv file is displayed below:



JOBDEF

B	C	D	E	F	G	H	I
DATE	TIME	PROCESS	SUBPROCESS	Source Institution	Event	Academic Period	Replace Event (Chec
'JAN-30-2007'	'23:00:00'	'MAPGROUP-EDW'	'LOAD_EDW_ENROLLMENT'	'SGHE'	'REG021'	'200910'	'Y'
'FEB-28-2007'	'23:00:00'	'MAPGROUP-EDW'	'LOAD_EDW_ENROLLMENT'	'SGHE'	'REG020'	'200830'	'Y'
'MAR-29-2007'	'23:00:00'	'MAPGROUP-EDW'	'LOAD_EDW_ENROLLMENT'	'SGHE'	'REG019'	'200820'	'Y'
'APR-31-2007'	'23:00:00'	'MAPGROUP-EDW'	'LOAD_EDW_ENROLLMENT'	'SGHE'	'REG018'	'200811'	'Y'
'MAY-30-2007'	'23:00:00'	'MAPGROUP-EDW'	'LOAD_EDW_ENROLLMENT'	'SGHE'	'REG017'	'200810'	'Y'
'JUN-30-2007'	'23:00:00'	'MAPGROUP-EDW'	'LOAD_EDW_ENROLLMENT'	'SGHE'	'REG016'	'200808'	'Y'
'JUL-30-2007'	'23:00:00'	'MAPGROUP-EDW'	'LOAD_EDW_ENROLLMENT'	'SGHE'	'REG015'	'200740'	'Y'
'AUG-30-2007'	'23:00:00'	'MAPGROUP-EDW'	'LOAD_EDW_ENROLLMENT'	'SGHE'	'REG014'	'200730'	'Y'
'SEP-31-2007'	'23:00:00'	'MAPGROUP-EDW'	'LOAD_EDW_ENROLLMENT'	'SGHE'	'REG013'	'200720'	'Y'
'OCT-30-2007'	'23:00:00'	'MAPGROUP-EDW'	'LOAD_EDW_ENROLLMENT'	'SGHE'	'REG012'	'200710'	'Y'

11. Click **Import** on the Schedule a Process page to re-import the .csv file into the Administrative UI.

12. Enter the name of the exported job into the subwindow, or search for it using **Browse**.

13. Click **Import Jobs**.

    The Select and View Scheduled Processes window opens in the background listing the new jobs.

View and Remove a Scheduled Process

You can schedule to run a process/job immediately, or at a future date/time. Processes scheduled to run at a future time remain in the job queue until runtime. Processes already in the queue can be edited as long as they have not run.

Use the steps below to access the queue and review which processes are scheduled, or to edit or delete a job from the queue.

1. Click **Options** from the Administrative menu.

2. Click **View and/or Remove Scheduled Processes**. The Select and View Scheduled Processes page opens.

3. Choose the date from which you would like to view scheduled processes from the Select and View Scheduled Processes page.

   Click **Select a Date** to open a calendar window. The default date is *Today*. When you Choose a date on the calendar, that date appears in the date field.

4. Click **Display Jobs**. The processes scheduled for the selected date display.

   To sort the columns in ascending or descending order, click the corresponding column header.

To edit a process/job

4.1. Click **Edit** next to the job number. The Schedule a Process page opens.

4.2. Make your changes.

4.3. Check the **Overwrite Existing Job in Queue** checkbox at the bottom of the page to overwrite the existing process.

Or, leave the box unchecked to create a duplicate process with the information.

4.4. Click **Submit**.

Or to delete a process/job

4.1. To delete processes, check the checkbox in the **Delete** column for the process you want to delete.

4.2. Click **Delete Jobs**.

Configure an Account and Stop a Running Job/Process

Sometimes jobs/processes run for too long, or are run by accident and you want to stop the job and maybe restart it later. A running job/process can be stopped from the job's control report if the user's account is configured to allow this feature.

Configure a User Account to Kill a Job/Process

A user account name must be configured before that user has the ability to stop a job.

Prerequisite

Set up the Administrative user name accounts (See "Set up Users and PINS (Banner ODS and Banner EDW)" on page 4-3.)

1. Click **Options** from the Administrative UI menu.

2. Click **Set up Parameters**.

3. Click **Create**.

4. In the **Internal Group Code** field type *JOB KILLER*, or select it from the drop-down list.

5. In the **Internal Code 1** field type *ACCOUNT NAME*, or select it from the drop-down list.

6. In the **External Code** field, type the administrative user name (account log in name).

If the user name was entered as an External Code when the parameter was created, you can select the name from the drop-down list.

7. Enter a description into the description field. The description is usually the same as what appears in the External Code field.

8. Click **Save**.

Kill a Running Job/Process

A running job/process can be stopped from within the job's/processes control report.

Prerequisite

The administrative account user name must be set up with this ability. See "Configure a User Account to Kill a Job/Process" on page 4-73

1. Click **Options** from the Administrative UI menu.

2. Click **View Control Reports**.

3. Click the link in the **Process** column for the job/process you want to stop.

   The Control Report for that process opens.

4. Click **Kill Job** located in the **Status** column.

Note

> This link only appears for jobs that are currently running, and if the user's account is properly configured to kill jobs. ■

   The Process Termination Wizard window opens and displays the process attributes.

5. Choose to either kill the process (at the operating system level), or to have the wizard display a list of Oracle commands needed to kill the process manually from the command line outside Banner ODS.

   Killing the process at the operating system level immediately stops the process, refreshes the Control Report, and displays a *Terminated* status for the process.

Note

> Killing a running process could leave the affected parts of Banner ODS in an undefined state, depending on the process that was stopped. Be sure to clean up data as necessary. Rerun the process to overwrite existing data. ■

Run a Process from Outside the Administrative UI

All Banner ODS and Banner EDW processes can be run from outside the Administrative UI. The processes are defined in the database as PL/SQL packaged procedures, therefore they can be run from outside the Administrative UI using any application that executes Oracle commands (typically Oracle's sqlplus utility). The name of the (packaged) procedure to run for a given process is defined using the JOB parameter (See “JOB Parameter” on page 4-94 in the Banner ODS Handbook for additional information.). However, an easy way to determine the name of the procedure used to execute a process is to submit that process to run at a future date, then view the process definition in the Job Queue using the steps below.

1. Schedule the Load Student process to run from the Schedule Banner ODS Mappings menu.

2. Select a date or time in the future.

3. Click **Submit**.

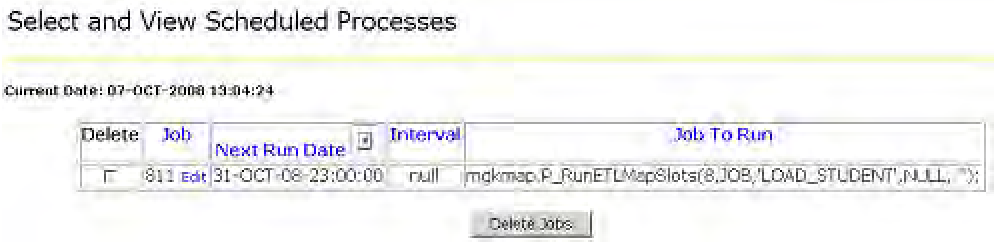
4. Return to the Options menu.

5. Select **View and/or Remove Scheduled Processes**.

6. Enter the date you scheduled the process to run.

7. Click **Display Jobs**.

   Below is an example page that might display:



Select and View Scheduled Processes

Current Date: 07-OCT-2008 13:04:24

Delete	Job	Next Run Date	Interval	Job To Run
□	811 Edit	31-OCT-08-23:00:00	null	mgkmap.P_RunETLMapSlots(8,JOB,'LOAD_STUDENT',NULL, '');

This page indicates in the **Job To Run** field that the Load Student process calls the (IA_ADMIN.) MGKMAP.p_runETLMapSlots, and takes the following parameters:

```
PROCEDURE P_RunETLMapSlots(userID IN VARCHAR2 DEFAULT NULL,

jobNumber IN BINARY_INTEGER DEFAULT NULL,

process IN VARCHAR2 DEFAULT NULL,

subProcess IN VARCHAR2 DEFAULT NULL,

parms IN VARCHAR2 DEFAULT NULL);
```

Parameter	Description
userID	Name/ID of the user associated with the job. The 8 in the example is the Administrative UI user account for "BILL" (for example, select `mgbuser_id` from `mgbuser` where `mgbuser_pidm=&userID;`)
jobNumber	Number of the job, and of the corresponding Control Report created when the job runs. When these jobs are run using the Administrative UI, using the DBMS_JOBS queue to run them, Oracle takes the *JOB* keyword and substitutes in the actual job number in the queue for this value. (811 in the example above). When the job is run outside the Administrative UI, you can give the job any number you want to. *Tip*: Do not use a number that is currently used by the Control Report or you'll have duplicate numbers. Begin numbering with high numbers so that the jobs are easy to find.
process	Name of the PROCESS to schedule (see Administrative UI PROCESS parameter description). In the example it is LOAD_STUDENT.
subprocess	Name of the SUBPROCESS to schedule (see Administrative UI PROCESS parameter description). In the example it is NULL (or empty)
parms	Any process-specific parameters needed. In the example there are none. Typicaljobs in the Banner ODS do not take parameters. When scheduling the job through the Administrative UI, you specify these parameters on the Submit page. See "PARAMETER Parameter" on page 4-105 in the Banner ODS Handbook for additional information.)

8. Issue the following command to run LOAD_STUDENT job:

```
EXEC mgkmap.P_RunETLMapSlots(8,811,'LOAD_STUDENT',NULL, '');
```

In the example, this would run the LOAD_STUDENT job as the userID 8 and the job number 811.

Note

This executes the job synchronously, outside of the DBMS_JOBS queue, meaning the job actually runs to completion and the above call does not return until the job completes. This is usually desired when calling jobs outside the Administrative UI.

It is also possible to submit jobs to the DBMS_JOBS queue externally as well to run jobs asynchronously. See the DBMS_JOBS package documentation for more details. ■

9. Remove the job from the queue when you are finished.

You can also externally execute all other Administrative UI processes, for example, Metadata Publishing and the Utilities, following similar steps.

Set up Customized Scheduled Processes

A scheduled process can be set up to run one or more customized mappings, and to have the new, customized process appear in the list of scheduled processes on the Select a Subprocess page.

For example, you want to bring in additional data and you don’t want to modify an existing mapping. You can create your own mapping(s) then run it either as part of one of the existing processes, like LOAD_STUDENT, REFRESH_ALL, etc., or create your own process, like LOAD_MY_DATA, etc.

The way mappings are organized can also be changed. Delivered mappings are grouped into processes. LOAD_STUDENT runs all the Student LOAD mappings, REFRESH_HR runs all the HR REFRESH mappings, etc. However, you can combine the groups differently to improve performance, to run them simultaneously in separate job processes, etc.

Banner EDW processes are set up like Banner ODS processes, in that they use the same ETL MAP PACKAGE, JOB, and SUBPROCESS parameter entries. Banner EDW processes also use the ETL MAP PACKAGE LOGIC parameter to indicate a termination point for the process. By default, all mappings within a process execute regardless of whether they error. Setting up an ETL MAP PACKAGE LOGIC allows a job to terminate if a mapping errors, and if the delivered processes to load the cubes use these parameters.

If you wanted to add new star or cube, you can duplicate one of the existing processes (complete with PROCESS, SUBPROCESS, ETL MAP PACKAGE entries).

To set up a scheduled mappings process, you need to:

- create a parameter record with an internal group code using the ETL MAP PACKAGE parameter set up for each new OWB mapping to be scheduled
- use the `SUBPROCESS` parameter to create a new group containing one or more customized mappings (`MAPGROUP-EDW`) to appear on the Select a Subprocess web page, and on the Schedule Banner ODS Mappings menu. It is also possible to add the new OWB mapping to an already existing group, by selecting one of the entries in the pull-down list.

- link the JOB parameter record to the process. This tells Banner ODS and Banner EDW which item in the Schedule Banner ODS or Banner EDW Mappings list (MAPGROUP-EDW) to run.

Follow the steps below. Examples appear after the steps.

1. Click **Set Up Parameters** from the Options menu. The Set Up a Parameter page opens.

2. Click **Create** from the Set Up a Parameter page. The Create a New Parameter page opens. Enter the information for the new process, or select it from the drop-down lists.

3. Click **Save**.

   Repeat these steps once for each mapping in the group to set up the ETL MAP PACKAGE parameter, once to set up the SUBPROCESS (or PROCESS) parameter, and once to set up the JOB parameter. They can be set up in any order.

4. To run the newly created process, click **Schedule a Process** from the Options menu. The Select a Process page opens.

5. Click **Schedule** Banner ODS or Banner EDW **Mappings**. The Select a Subprocess page opens.

6. Choose your new process.

Banner ODS Example:

The example below walks you through how to create a scheduled process called TEST_LOAD_STUDENT_COURSE. This group will have one mapping called TEST_LOAD_STUDENT_COURSE_1

First, create an internal group record using the ETL MAP PACKAGE parameter.

1. Click **Set Up Parameters** from the Options menu. The Set up a Parameter page opens.

2. Open the Create a New Parameter page.

3. Enter the information below into the fields.

In This Field ...	Enter This ...	Here's Why ...
Internal Group	`ETL MAP PACKAGE`	*Must* be `ETL MAP PACKAGE`.
Internal Code 1	`TEST_LOAD_STUDENT_COURSE`	Mapping group name. Create your own name, or specify an existing group if you want to add this mapping to an existing group.
Internal Code 2	`TEST_LOAD_STUDENT_COURSE_1`	Mapping name in OWB and the package name in Banner ODS.
Internal Code Sequence Number	1	Order of the mappings within the Mapping group (Internal Code 1). Controls the order in which multiple mappings are executed within that group. If you add more mappings then the code should on number up such as 2, 3, 4, 5, etc.
External Code	`ODS_TARGET_STUDENT`	Location/project in the OWB repository. These locations pertain to the schema containing the target table(s).
Description	`TEST_LOAD_STUDENT_COURSE_1`	Actual name of the mapping. *Must* be the exact same entry as entered into the **Internal Code 2** field.
System Required	No	Parameter records entered through the Administrative UI are marked as *No* to differentiate those delivered with the product. Display only.

4. Click **Save**.

Second, set up the `SUBPROCESS` parameter so that you can create and name a new group of one or more customized mappings. This tells Banner ODS that you want this new process(es) to appear on the Select a Subprocess page, and on the Schedule Banner ODS Mappings menu (`MAPGROUP`) on that page.

1. Click **Create a New Parameter** at the bottom of the page. The fields on the page reset.

2. Enter the following information.

In This Field ...	Enter This ...	Here's Why ...
Internal Group	`SUBPROCESS`	*Must* be `SUBPROCESS`. This tells Banner ODS to display this group on the Select a Subprocess menu.
Internal Code 1	`MAPGROUP`	Must be MAPGROUP in order to display this group on the Schedule Banner ODS Mappings menu. You can enter a different SUBPROCESS name if you want to create or use additional process listings.
Internal Code 2		This field remains blank.
Internal Code Sequence Number	1	Order of the entries on the Select a Subprocess menu. Entries with the same number are sorted by group name. If you add more mappings then the code should on number up such as 2, 3, 4, 5, etc.
External Code	`TEST_LOAD_MST_STUDENT`	Group name. Must be the same as what was entered into the **Internal Code 1** field when you set up the `ETL MAP PACKAGE` parameter.
Description	`TEST Load MST Student`	Actual text you want to display on the Schedule Banner ODS Mappings list on the Schedule a Subprocess page.
System Required	No	Parameter records entered through the Administrative UI are marked as *No* to differentiate those delivered with the product. Display only.

3. Click **Save**.

Third, link the `JOB` parameter to the new group of mappings. This tells Banner ODS which item in the Schedule Banner ODS Mappings list (`MAPGROUP`) to run.

1. Click **Create a New Parameter** at the bottom of the page. The fields on the page reset.

2. Enter the following information.

In This Field ...	Enter This ...	Here's Why ...
Internal Group	`JOB`	`JOB` *must* be entered.
Internal Code 1	`MAPGROUP`	Must match the **Internal Code 1** field of the `SUBPROCESS` record.
Internal Code 2	`TEST_LOAD_STUDENT`	*Must* match the **Internal Code 1** field when you set up the `ETL MAP PACKAGE`.
Internal Code Sequence Number	1	Leave as *is*.
External Code	0	Leave as *0*.
Description	`mgkmap.P_RunETLMapSlots`	Name of the PL/SQL procedure executed by the process. For mapping scheduled processes, use the standard procedure P_RunETLMapSlots
System Required	No	Parameter records entered through the Administrative UI are marked as *No* to differentiate those delivered with the product. Display only.

3. Click **Save**.

Banner EDW Example:

The example below walks you through how to create a scheduled process called TEST_LOAD_EDW_EMPLOYEE. This group will have one mapping called TEST_LOAD_EDW_EMPLOYEE_1

First, create an internal group record using the ETL MAP PACKAGE parameter.

1. Open the Create a New Parameter page.

2. Enter the information below into the fields.

In This Field . . .	Enter This . . .	Here’s Why . . .
Internal Group	ETL MAP PACKAGE	*Must* be `ETL MAP PACKAGE`.
Internal Code 1	`TEST_LOAD_EDW_EMPLOYEE`	Mapping group name. Create your own name, or specify an existing group if you want to add this mapping to an existing group.
Internal Code 2	`TEST_LOAD_EDW_EMPLOYEE_1`	Mapping name in OWB and the package name in Banner EDW.
Internal Code Sequence Number	1	Order of the mappings within the Mapping group (Internal Code 1). Controls the order in which multiple mappings are executed within that group. If you add more mappings then the code should on number up such as 2, 3, 4, 5, etc.

External Code	`EDW_GENERATE_LEDGER`
	OR
	`STAGE_GENERAL_LEDGER`
		Location/project in the OWB repository. These locations pertain to the schema containing the target table(s).

Description	`TEST_LOAD_EDW_EMPLOYEE_1`	Actual name of the mapping. *Must* be the exact same entry as entered into the **Internal Code 2** field.
System Required	No	Parameter records entered through the Administrative UI are marked as *No* to differentiate those delivered with the product. Display only.

3. Select **Save**.

Second, set up the SUBPROCESS parameter so that you can create and name a new group of one or more customized mappings. This tells Banner EDW that you want this new process(es) to appear on the Select a Subprocess page, and on the Schedule Banner EDW Mappings menu (`MAPGROUP-EDW`) on that page.

1. Click **Create a New Parameter** at the bottom of the page. The fields on the page reset.

2. Enter the following information.

In This Field . . .	Enter This . . .	Here's Why . . .
Internal Group	`SUBPROCESS`	*Must* be `SUBPROCESS`. This tells Banner EDW to display this group on the Select a Subprocess menu.
Internal Code 1	`MAPGROUP-EDW`	Must be MAPGROUP in order to display this group on the Schedule Banner EDW Mappings menu. You can enter a different SUBPROCESS name if you want to create or use additional Process listings.
Internal Code 2		This field remains blank.
Internal Code Sequence Number	1	Order of the entries on the Select a Subprocess menu. Entries with the same number are sorted by group name. If you add more mappings then the code should on number up such as 2, 3, 4, 5, etc.
External Code	`TEST_LOAD_EDW_EMPLOYEE`	Group name. Must be the same as what was entered into the **Internal Code 1** field when you set up the `ETL MAP PACKAGE` parameter.
Description	`TEST Load EDW Employee`	Actual text you want to display on the Schedule Banner EDW Mappings list on the Schedule a Subprocess page.
System Required	No	Parameter records entered through the Administrative UI are marked as *No* to differentiate those delivered with the product. Display only.

3. Click **Save**.

Third, link the `JOB` parameter to the new group of mappings. This tells Banner EDW which item in the Schedule Banner EDW Mappings list (`MAPGROUP-EDW`) to run.

1. Click **Create a New Parameter** at the bottom of the page. The fields on the page reset.

2. Enter the following information.

In This Field . . .	Enter This . . .	Here's Why . . .
Internal Group	`JOB`	`JOB` *must* be entered.
Internal Code 1	`MAPGROUP-EDW`	Must match the **Internal Code 1** field of the `SUBPROCESS` record.
Internal Code 2	`TEST_LOAD_STUDENT`	*Must* match the **Internal Code 1** field when you set up the `ETL MAP PACKAGE`.
Internal Code Sequence Number	1	Leave as *is*.
External Code	0	Leave as *0*.
Description	mgkmap_P_RunETLMaps	Name of the PL/SQL procedure executed by the process. For mapping scheduled processes, use the standard procedure mgkmap_P_RunETLMaps.
System Required	No	Parameter records entered through the Administrative UI are marked as *No* to differentiate those initially delivered with the product. Display only.

3. Click **Save**.

Schedule a Process Parameters (Banner ODS and Banner EDW)

The Administrative UI uses several system parameters to create the web pages associated with scheduling a process (running the mappings). The next sections describe these parameters, their purpose, and their role in scheduling a process.

Note

These runtime parameters are different from the set up parameters stored in MTVPARM (See “Set up Parameters (Banner ODS and Banner EDW)” on page 4-58.) ■

Parameters are maintained on the Set Up a Parameter page of the Administrative UI. See “Update or Delete a Parameter” on page 4-60 for additional information on updating parameters. Each parameter and its purpose appear below:

Parameter	Definition
“INSTALLED PROCESS Parameter” on page 4-86	Populates a list of processes displayed on the Select a Process page.
“PROCESS INFO Parameter” on page 4-87	Defines process descriptions that optionally display on the Select a Process page.
“SUBPROCESS Parameter” on page 4-89	Populates a list of processes displayed on the Select a Subprocess page.
“SUBPROCESS INFO Parameter” on page 4-91	Defines subprocess descriptions that optionally display on the Select a Subprocess page and defines information specific to a subprocess that displays on the Schedule a Process page.
“JOB Parameter” on page 4-94	Defines the actual name of job (program) to run when you schedule a process.
“ETL MAP PACKAGE Parameter” on page 4-96	Groups related jobs (OWB mappings) as one job.
“ETL MAP PACKAGE LOAD PURGE Parameter” on page 4-98	Identifies DELETE mappings for the Load Purge Process.
“ETL MAP PACKAGE LOGIC Parameter” on page 4-99	Allows you to specify job termination logic for a mapping within a job stream. By default, all mappings in a job run in sequence regardless of whether they have errors or not. By defining an ETL Map Package record for a given mapping in a job, you can have the job stop if that mapping encounters errors. This parameter is used primarily with Banner EDW jobs as they have dependencies from one step (or mapping) to another, while Banner ODS mappings are independent of each other.
“ETL MAP PACKAGE RECONCILE LOGIC Parameter” on page 4-99	Provides a list of mappings that are exceptions in the reconcile Banner ODS tables process. In this list are the mappings that are ignored in the reconcile process because of the complexity of the mapping or other factors outside the scope of reconciling that Banner ODS table. This list also includes mappings that require either multiple source composite views or mappings in order to reconcile a Banner ODS table.

Parameter	Definition
"ETL SLOT PACKAGE Parameter" on page 4-100	Groups together related slot jobs (SQL packages) as one job.
"ETL CONTROL GROUP Parameter" on page 4-101	Groups together ETL MAP PACKAGE and/or ETL SLOT PACKAGE jobs as one job.
EDW EXTRACT	Controls how certain Banner EDW extracts operate when moving information from Banner ODS to Banner EDW. Refer to the "Set Up Data" chapter for more information about defining EDW Extract parameters.
"EVENT parameter" on page 4-103	Defines Events for data being loaded in Banner ODS. These Events let you freeze or take a snapshot of data at a point in time.
"EVENT-EDW parameter" on page 4-104	Defines Events for data being loaded in Banner EDW. These Events let you freeze or take a snapshot of data at a point in time.
"PARAMETER Parameter" on page 4-105	Defines a list of a job's input parameters you need to supply when you schedule a process.

INSTALLED PROCESS Parameter

The Description field for this parameter defines the process names that display on the Select a Process page of the Administrative UI. You can choose from that list to schedule a process.

This parameter is delivered with one entry for each type of process (job) that you can run. The processes defined by this parameter have 'children' defined by the SUBPROCESS and JOB parameters. To designate the parent/child relationship, match the External Code of the INSTALLED PROCESS to the Internal Code 1 of the SUBPROCESS and the Internal Code 1 of the JOB.

The following table illustrates a sample of the values as delivered. This is just a sample. The second row gives a definition of each field.

Internal Group: `INSTALLED PROCESS`

Internal Code 1	Internal Code 2	Internal Code Seq.	External Code	Description
Can be a short description of the process. This field is not used in processing, but it requires a value.	N/A	Order for entries on Select a Process page.	Short description of the process. Map values of this field to the **Internal Code 1** values of Subprocesses and related Jobs to define them as its children.	Actual process name that appears on the Select a Process administrative page.
ADHOC_FREEZE	N/A	6	ADHOC_FREEZE	Freeze A Single Banner ODS Table/View
FREEZE_TABLE	N/A	5	FREEZE_TABLE	Freeze Multiple Banner ODS Tables/Views
MAPGROUP	N/A	1	MAPGROUP	Schedule Banner ODS Mappings
MAPGROUP-EDW		2	MAPGROUP-EDW	Schedule Banner EDW Mappings

Setting up the INSTALLED PROCESS parameter

The only field you should change for the delivered values of this parameter is the **Description** field. If you want to change the name of a process that appears on the Select a Process page, change its description.

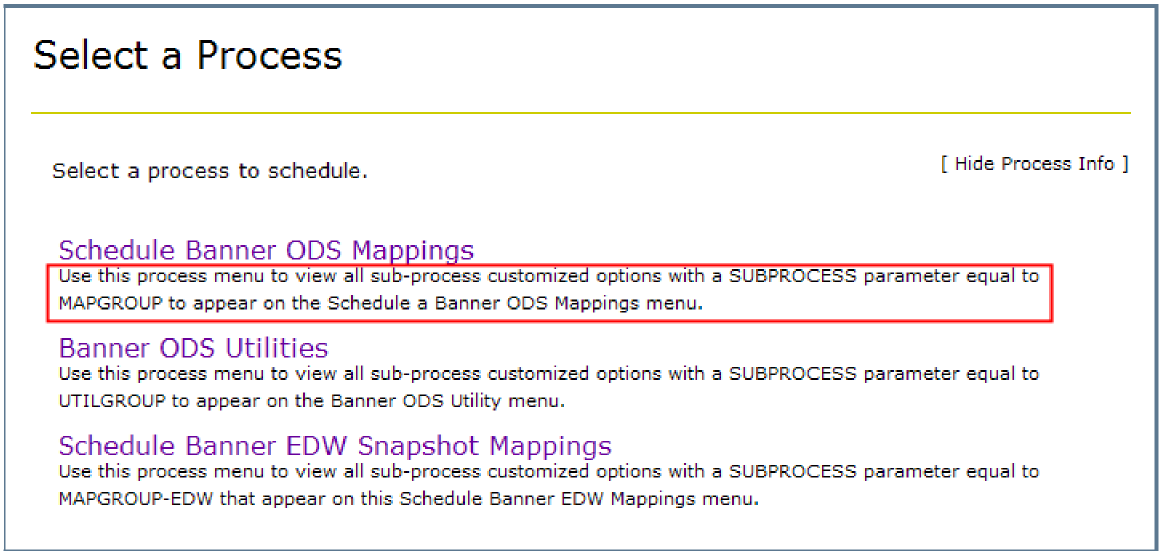
Create a new Installed Process parameter value

If you want to add a process developed by your institution, create the process and add it as a new record for this parameter.

PROCESS INFO Parameter

The Process Info Parameter records define process descriptions that optionally display on the Select a Process page. Process Info parameter records match to the related Installed Process parameter records based on sharing the same **Internal Code 1** and **External Code** values.

Parameter records with an **Internal Code Group** = *PROCESS INFO* and **Internal Code 2** = *PROCESS_LIST* use the **Description** field to define the process descriptions that a user can optionally display when they select the [Show Process Info] link on the Select a Process page as illustrated in the following picture.



Select a Process

Select a process to schedule.

[Hide Process Info]

Schedule Banner ODS Mappings
Use this process menu to view all sub-process customized options with a SUBPROCESS parameter equal to MAPGROUP to appear on the Schedule a Banner ODS Mappings menu.

Banner ODS Utilities
Use this process menu to view all sub-process customized options with a SUBPROCESS parameter equal to UTILGROUP to appear on the Banner ODS Utility menu.

Schedule Banner EDW Snapshot Mappings
Use this process menu to view all sub-process customized options with a SUBPROCESS parameter equal to MAPGROUP-EDW that appear on this Schedule Banner EDW Mappings menu.

Descriptions are included for each process that is delivered with the warehouse. To define a processes description for a new job, create a Parameter record with values defined as follows:

Field	Value
Internal Group	PROCESS INFO
Internal Code 1	<Installed Process Name> Enter the name of the installed process. This value needs to match the Internal Code 1 value of the related Installed Process parameter.
Internal Code 2	PROCESS_LIST
Internal Code Sequence Number	Not used
External Code	<Installed Process Name> Enter the name of the installed process. This value needs to match the Internal Code 1 value of the related Installed Process parameter.

Field	Value
Description	<Description of Process> Enter the description of the process that will display on the Select a Process page when a user selects the [Show Process Info] link.
System Required	No

SUBPROCESS Parameter

The **Description** field of this parameter defines the subprocess names that display on the Select a Subprocess of the Administrative UI.

This parameter is delivered with one entry for each subprocess, which are processes grouped under one of the main processes—Schedule Banner ODS or Banner EDW Mappings, Freeze Multiple Banner ODS Tables/Views, or Freeze A Single Banner ODS Table/View.

Subprocesses are related to `JOB` parameter values and both are “children” of one of the processes defined by the `INSTALLED PROCESS` parameter. To designate the parent/child relationship, match the **External Code** of the `INSTALLED PROCESS` to the **Internal Code 1** of the `SUBPROCESS` and the **Internal Code 1** of the `JOB`.

The following table illustrates a sample of the values as delivered. This is just a sample. The second row gives a definition of each field.

Internal Group: `SUBPROCESS`

Internal Code 1	Internal Code 2	Internal Code Seq.	External Code	Description
Map to **External Code** of the `INSTALLED PROCESS` that is parent to this subprocess.	N/A	Order for entries with same **Internal Code 1**.	Short description of the subprocess. Use values of this field in the **Internal Code 2** of its related `Job`.	Actual process name that appears on the Select a Subprocess administrative page.
`MAPGROUP`	N/A	1	`LOAD_ALL`	Load all Banner ODS products
`MAPGROUP`	N/A	2	`LOAD_ALL_SLOTS`	Load all Banner ODS slotted tables
`MAPGROUP`	N/A	3	`LOAD_FINANCE`	Load finance

Internal Code 1	Internal Code 2	Internal Code Seq.	External Code	Description
MAPGROUP	N/A	8	REFRESH_ALL	Refresh all Banner ODS products
MAPGROUP	N/A	9	REFRESH_FINANCE	Refresh finance
MAPGROUP	N/A	14	RUN_SINGLE_MAP	Run a single mapping

Set up the SUBPROCESS Parameter

The only field you should change for the delivered values of this parameter is the Description. If you want to change the name of a subprocess that appears on the Select a Subprocess page, change its Description.

Create a SUBPROCESS Parameter

You can add to the subprocess list jobs developed by your institution that you can then run via the Administrative UI. Use the following steps to do this.

1. Create the job.

2. Add the job to the list of subprocesses you can schedule by creating a new Subprocess parameter with the following values:

   2.1. **Internal Group**: Subprocess

   2.2. **Internal Code 1**: The External Code value of the INSTALLED PROCESS you want the subprocess to be listed under. Existing values include:

   - MAPGROUP to list under the Schedule OWB Mappings process.
   - FREEZE_TABLE to list under the Freeze Multiple Banner ODS Tables process.
   - ADHOC_FREEZE to list under the Freeze A Single Banner ODS Table process.

   2.3. **Internal Code 2**: blank

   2.4. **Internal Code Sequence Number**: Number indicating the order in which to run this subprocess.

   2.5. **External Code**: The External Code value of the INSTALLED PROCESS you want the subprocess to be listed under. See existing values listed above with the **Internal Code 1** field.

   2.6. **Description**: The name of the subprocess that will display on the Select a Subprocess page in the Administrative UI.

SUBPROCESS INFO Parameter

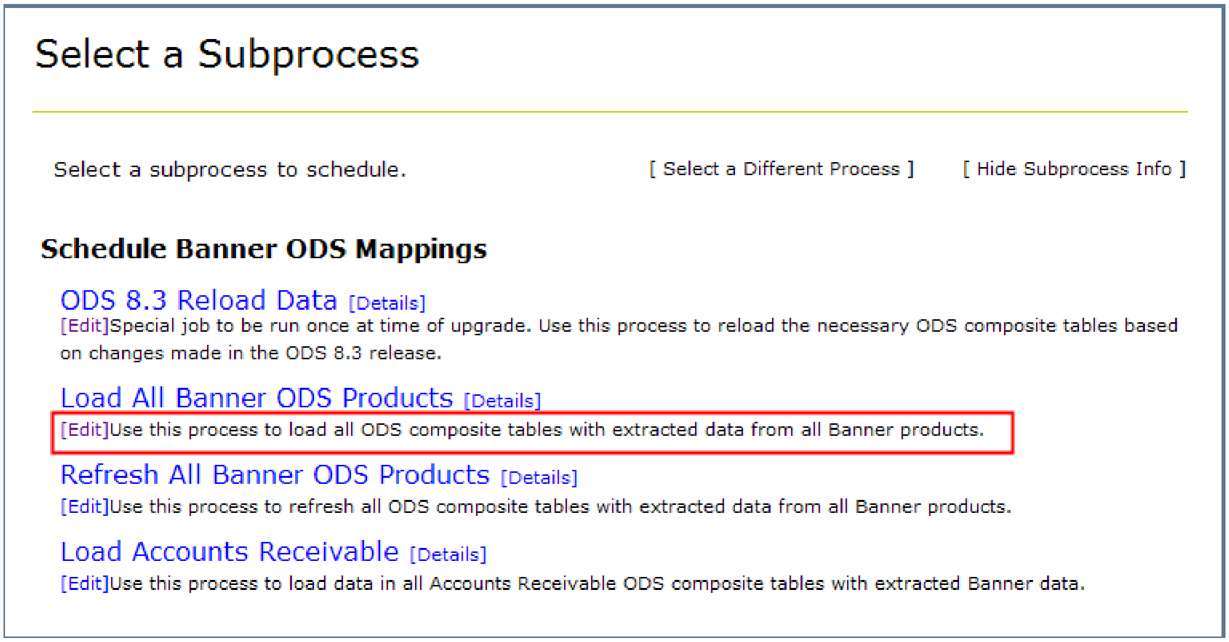
The Subprocess Info Parameter records define one of the following pieces of information:

- "PROCESS_LIST records" - Define subprocess descriptions that optionally display on the Select a Subprocess page
- "SCHEDULING_PAGE records" - Define special instructions or information that will display for a process on the Schedule a Process page

Subprocess Info parameter records match to the related Subprocess parameter records based on sharing the same **Internal Code 1** and **External Code** values.

PROCESS_LIST records

Parameter records with an **Internal Code Group** = *SUBPROCESS INFO* and **Internal Code 2** = *PROCESS_LIST* use the **Description** field to define the subprocess descriptions that a user can optionally display when they select the [Show Subprocess Info] link on the Select a Subprocess page illustrated in the following picture.



Select a Subprocess

Select a subprocess to schedule.

Schedule Banner ODS Mappings

ODS 8.3 Reload Data [Details]
[Edit]Special job to be run once at time of upgrade. Use this process to reload the necessary ODS composite tables based on changes made in the ODS 8.3 release.

Load All Banner ODS Products [Details]
[Edit]Use this process to load all ODS composite tables with extracted data from all Banner products.

Refresh All Banner ODS Products [Details]
[Edit]Use this process to refresh all ODS composite tables with extracted data from all Banner products.

Load Accounts Receivable [Details]
[Edit]Use this process to load data in all Accounts Receivable ODS composite tables with extracted Banner data.

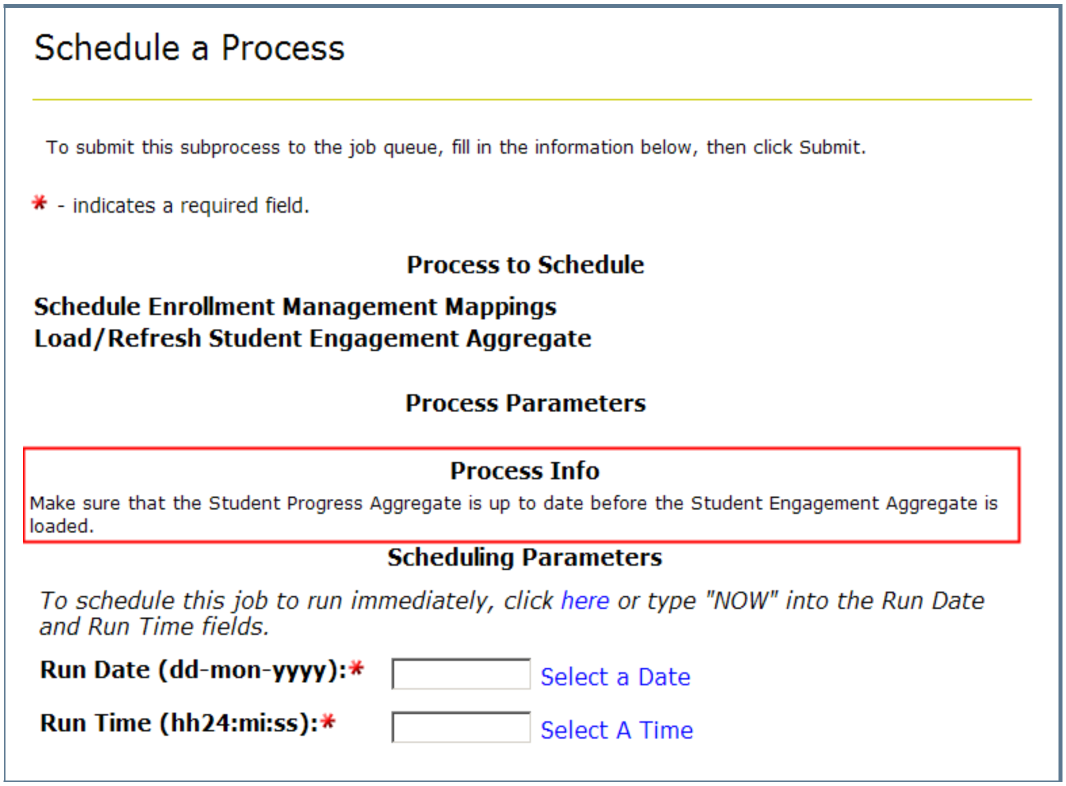
Create Subprocess Info/PROCESS_LIST records

Descriptions are included for each subprocess that is delivered with the warehouse. To define a Subprocesses description for a new job, create a Parameter record with values defined as follows:

Field	Value
Internal Group	SUBPROCESS INFO
Internal Code 1	<Mapping Group Name> Enter the name of the mapping group for the subprocess. This value needs to match the Internal Code 1 value of the related Subprocess parameter.
Internal Code 2	SUBPROCESS_LIST
Internal Code Sequence Number	Not used
External Code	<Subprocess Name> Enter the subprocess name. This value needs to match the External Code value of the related Subprocess parameter.
Description	<Description of Subprocess> Enter the description of the subprocess that will display on the Select a Subprocess page when a user selects the [Show Subprocess Info] link.
System Required	No

SCHEDULING_PAGE records

Parameter records with an **Internal Code Group** = *SUBPROCESS INFO* and **Internal Code 2** = *SCHEDULING_PAGE* use the **Description** field to define information specific to the subprocess that will display in the Process Info section of the Schedule a Process page illustrated in the following picture.



Schedule a Process

To submit this subprocess to the job queue, fill in the information below, then click Submit.

* - indicates a required field.

Process to Schedule

Schedule Enrollment Management Mappings
Load/Refresh Student Engagement Aggregate

Process Parameters

Process Info

Make sure that the Student Progress Aggregate is up to date before the Student Engagement Aggregate is loaded.

Scheduling Parameters

To schedule this job to run immediately, click here or type "NOW" into the Run Date and Run Time fields.

Run Date (dd-mon-yyyy):* Select a Date

Run Time (hh24:mi:ss):* Select A Time

Create Subprocess Info/SCHEDULING_PAGE records

To define information that will display for Subprocesses in the Process Info section of the Schedule a Process page, create a Parameter record with values defined as follows:

Field	Value
Internal Group	SUBPROCESS INFO
Internal Code 1	<Mapping Group Name> Enter the name of the mapping group for the subprocess. This value needs to match the Internal Code 1 value of the related Subprocess parameter.
Internal Code 2	SCHEDULING_PAGE
Internal Code Sequence Number	Not used
External Code	<Subprocess Name> Enter the subprocess name. This value needs to match the External Code value of the related Subprocess parameter.

Field	Value
Description	<Process Info Description> Enter the information or instructions specific to the subprocess that will display in the Process Info section of the Schedule a Process page for the related subprocess.
System Required	No

JOB Parameter

This parameter defines the actual program name of a job that gets sent to the job queue via the **Schedule a Process** administrative page in the Options menu.

This parameter is delivered with one entry for each process (job) that you can schedule. A Job is related to a SUBPROCESS and a “child” of one of the processes defined by the INSTALLED PROCESS parameter. To designate the parent/child relationship, match the **External Code** of the INSTALLED PROCESS to the **Internal Code 1** of the SUBPROCESS and the **Internal Code 1** of the JOB.

The following table illustrates a sample of the values as delivered. This is just a sample. The second row gives a definition of each field.

Internal Group: JOB

Internal Code 1	Internal Code 2	Internal Code Seq.	External Code	Description
Map to **External Code** of the INSTALLED PROCESS that is parent to this job.	Map to **External Code** of the SUBPROCESS related to this job.	N/A	The number of parameters that get passed to the mapping.	Actual program name (package.procedure) for the job. Refer to the mgkproc package for more information about submitting jobs.
MAPGROUP	LOAD_ALL	1	0	mgkmap.P_RunETLMapSlots
MAPGROUP	LOAD_ALL_SLOTS	1	0	mgkmap.P_RunETLMapSlots
MAPGROUP	LOAD_AR	1	0	mgkmap.P_RunETLMapSlots
MAPGROUP	LOAD_FINANCE	1	0	mgkmap.P_RunETLMapSlots

Set up the JOB Parameter

You should not edit any of these entries. If your institution doesn’t maintain one of the areas of Banner ODS or Banner EDW data, you can delete all of the entries for that area.

Create a Job Parameter Value

You can add a program developed by your institution to the Schedule a Process page. Create the program and define it by adding a new record for this parameter with the program name in the **Description** field.

OWB Mappings and Slot Packages

OWB mappings are executed from the Administrative UI via the MGKMAP package. This package provides routines for running both OWB mappings and slotted table LOAD and UPDATE jobs. Refer to the MGKMAP package for more details.

The primary APIs used in the MGKMAP package are:

P_RunETLMapSlots:
When a process/subprocess pair is passed to the procedure, it runs all mappings and slot package records associated with that process/subprocess combination. Specifically, if any **ETL Control Group** records are defined for the process/subprocess pair and the **Description** value is *Y*, then the procedure runs all Mapping and Slot Package records associated with those Control Group areas. If there are no **ETL Control Group** records associated the process/subprocess pair but there are individual mapping records associated with the pair, the procedure runs those Mapping and Slot Package records.

Example: P_RunETLMapSlots

> As delivered, the ETL Control Group parameter records for all baseline systems have an External Code value of *Y*. This means data for all systems is loaded into Banner ODS when you submit the Load All Banner ODS Products and Refresh All Banner ODS Products processes. If you want to load only Student and Finance data into Banner ODS, set the **External Code** field to *N* for the ETL Control Group record for each of the other systems. The Mappings and Slot Packages will only run for Student and Finance when you submit the Load All Banner ODS Products and Refresh All Banner ODS Products processes.

P_RunETLMaps:
When a process/subprocess pair is passed to the procedure, it runs all mappings associated with that process/subprocess combination. This API follows the same processing rules as `P_RunETLMapSlots`, except that it only runs mappings; it does *not* run Slot Packages.

P_RunETLSlots:
When a process/subprocess pair is passed to the procedure, it runs all Slot Packages associated with that process/subprocess combination. This API follows the same processing rules as `P_RunETLMapSlots`, except that it only runs Slot Packages; it does *not* run mappings.

P_RunSingleMap:
When a process/subprocess pair and mapping name are passed to the procedure, it runs that single mapping.

ETL MAP PACKAGE Parameter

Hundreds of OWB mappings are used to load and refresh Banner ODS and Banner EDW data, and to cleanse Banner EDW data. The ETL Map Package parameter defines groups of related mappings as one job. This allows you to quickly run just one job that, for example, loads all of the AR mappings.

This parameter is delivered with one entry for each mapping. The actual program name for the mapping occupies the Internal Code 2 and Description fields and is associated with an ETL group name in the Internal Code 1 field.

Example

> When you run the LOAD_AR job using the Schedule a Process option in the Administrative UI, the mappings associated with each ETL Map Package entry that has an Internal Code 1 of LOAD_AR is run. The External Code field contains the Location value defined for the mappings in OWB. These values are defined at mapping deployment time (usually at install) and are generally not modified.

The following table shows the entries for ETL Map Package entries that have an **I**nternal Code 1 value of LOAD_AR. The following table illustrates a sample of the values as delivered. This is just a sample. The second row gives a definition of each field.

Internal Group: ETL MAP PACKAGE

Internal Code 1	Internal Code 2	Internal Code Seq.	External Code	Description
The ETL group to which the mapping is assigned.	Mapping name.	Order to run the mapping within the group.	OWB locations of the mapping.	Mapping name.
LOAD_AR	LOAD_MTT_ACCOUNT	1	ODS_TARGET_AR	LOAD_MTT_ACCOUNT
LOAD_AR	LOAD_MTT_ACCOUNT_ DETAIL	2	ODS_TARGET_AR	LOAD_MTT_ACCOUNT_ DETAIL
LOAD_AR	LOAD_MTT_ APPLICATION_OF_PAYM	3	ODS_TARGET_AR	LOAD_MTT_ APPLICATION_OF_PAYM
LOAD_AR	LOAD_MTT_AR_ DEPOSITS	4	ODS_TARGET_AR	LOAD_MTT_AR_ DEPOSITS
LOAD_AR	LOAD_MTT_CONTRACT	5	ODS_TARGET_AR	LOAD_MTT_CONTRACT
LOAD_AR	LOAD_MTT_EXEMPTION	6	ODS_TARGET_AR	LOAD_MTT_EXEMPTION

Internal Code 1	Internal Code 2	Internal Code Seq.	External Code	Description
LOAD_AR	LOAD_MTT_ INSTALLMENT_PLAN	7	ODS_TARGET_AR	LOAD_MTT_ INSTALLMENT_PLAN
LOAD_AR	LOAD_MTT_LEDGER_ ACCOUNTING	8	ODS_TARGET_AR	LOAD_MTT_LEDGER_ ACCOUNTING

Set up the ETL MAP PACKAGE Parameter

You shouldn't change any of the delivered values for this parameter.

Create an ETL Map Package

You can define a new ETL group of mappings by creating a new set of related ETL Map Package parameter entries. Create one new entry for each mapping in the group using the following steps.

1. Create new entries with a new ETL Group name in the **Internal Code 1** field.
2. Specify in the **Internal Code 2** and **Description** fields for the mappings you want to include in the group.
3. Specify the location of each mapping in the **External Code** field.
4. Specify the order in which to run the mappings in the **Internal Code Sequence Number** field.
5. Add the new ETL group to the list of subprocesses you can schedule by creating a new Subprocess parameter with the following values:

    5.1. **Internal Group**: Subprocess

    5.2. **Internal Code 1**: MAPGROUP

    5.3. **Internal Code 2**: blank

    5.4. **Internal Code Sequence Number**: Number indicating the order in which to run this subprocess.

    5.5. **External Code**: the new ETL Group name you created. This is the value in **Internal Code 1** in the ETL Map Package entries created above.

    5.6. **Description**: Name of the subprocess (ETL Group job) as it appears on the Select a Subprocess administrative page.

ETL MAP PACKAGE LOAD PURGE Parameter

As part of the LOAD mapping Change Table Purge process, use this parameter to define the appropriate DELETE mapping for those LOAD_x mappings that do not have an equivalent DELETE_x counterpart, or where no Change table purge is required.

The MGKMAP package in Banner ODS (in the IA_ADMIN schema which executes the various OWB mappings that make up a job) automatically runs the Purge process for each change table that is related to a particular Load mapping. The name of the change table and the PROCESS_ID (a key field in the change table that identifies which records relate to a given mapping) are retrieved from the corresponding Delete mapping of the same name where LOAD_x = DELETE_x. For example, for the LOAD mapping LOAD_MST_STUDENT, the DELETE_MST_STUDENT mapping is used to identify the change table and process ID. However, occasionally there is no direct equivalent DELETE mapping for the LOAD mapping in context, or no change table purge is required. For example:

- Sometimes the mapping names do not match exactly (for example, LOAD_MAT_ORGANIZATION_CONTACT and DELETE_MAT_ORGANIZATION_CONT).
- LOAD mappings that require multiple DELETE mappings.
- LOAD mappings where change tables do not exist (such as the VALIDATION mappings) and subsequently no purge process is required.
- LOAD mappings are broken up across several sequential mappings (such as LOAD_MFT_TRANS_HISTORY_1, LOAD_MFT_TRANS_HISTORY_2, LOAD_MFT_TRANS_HISTORY_3, and so on) and the change table purge process is only required to run once (DELETE_MFT_TRANS_HISTORY).

In these cases, a Load Purge parameter is required to provide the MGKMAP package with the appropriate crosswalk information to designate what DELETE mapping(s) are required to run the Change Table Purge process, or when the Change Table Purge process should be ignored.

Any errors encountered when running the purge appear in the Load Control Report.

Use the following codes:

- Group Code: ETL_MAP_PACKAGE_LOAD_PURGE
- Internal Code: Enter the designated LOAD mapping
- External Code: Enter the designated DELETE mapping(s) or, enter NA to disable the Purge process for a given LOAD mapping.

The following table illustrates a sample of the values as delivered. This is just a sample. The first row gives a definition of each field.

Internal Group: ETL MAP PACKAGE LOAD PURGE

Internal Code 1	Internal Code 2	Internal Code Seq.	External Code	Description
Name of LOAD mapping		Ignored	Name of DELETE mapping (s – note comma-separated if >1) or “NA”	Simple text to explain record’s purpose.
LOAD_MAT_ ORGANIZATION_CONTACT		1	DELETE_MAT_ ORGANIZATION_CONT	Load Purge Record
LOAD_MGT_VALIDATION _GENERAL		1	NA	Load Purge Record
LOAD_MFT_TRANS_ HISTORY_1		1	DELETE_MFT_TRANS_ HISTORY	Load Purge Record
LOAD_MFT_TRANS_ HISTORY_2		1	NA	Load Purge Record
LOAD_MFT_TRANS_ HISTORY_3		1	NA	Load Purge Record

ETL MAP PACKAGE LOGIC Parameter

This parameter controls job processing if an error occurs during one of the mappings. By default, the MGKMAP package, which executes mappings for a job, runs all mappings for the job, regardless of whether they complete successfully. This assumes that there are no dependencies between mappings. However, within Banner EDW, there are such dependencies (such as when the Time Key does not process correctly).

Use this parameter to override processing logic. Specifically, if a parameter record exists with the ETL Map Package Logic group code, and the same Internal Code 1 (the job name) and Internal Sequence Number as the ETL Map Package record for the mapping for the job in question, and the External Code is set to “Terminate Job,” then the job stops if there is an error in that particular mapping.

ETL MAP PACKAGE RECONCILE LOGIC Parameter

This parameter controls how the reconciliation process identifies LOAD mappings which do not follow the standard pattern (of one source Composite view equating to one Banner ODS Composite table). Those exceptions are notes by the External Code, being either:

- IGNORE: used to identify mappings *not* to try to reconcile
- IGNORE COLUMN: used to identify specific columns (1 parameter record per column) not to try to reconcile, where Int Code2 stores the column name

- UNION: used to identify Composite tables populated by multiple Composite views, in which case the name(s) of the related mappings are stored in the **Description** field.

Internal Group: ETL MAP PACKAGE RECONCILE LOGIC

Internal Code 1	Internal Code 2	Internal Code Seq.	External Code	Description
Name of LOAD mapping	Ignored	1	Action to take	Either simple description or name of related LOAD mappings
LOAD_MAT_CONSTIT_STAFF_ASSIGN	Ignored	1	IGNORE	Do not reconcile this mapping
LOAD_MST_STDNT_CRSE_ATT_STEP_1	Ignored	1	UNION	LOAD_MST_STDNT_CRSE_ATT_STEP_2
LOAD_MTT_ACCOUNT_DETAIL	OPERATING_DATE	1	IGNORE COLUMN	Do not reconcile this column

ETL SLOT PACKAGE Parameter

The ETL Slot Package parameter is similar to the ETL Map Package parameter; it defines groups of related Slot Packages as one job. The difference is that the groups defined by the Slot Package parameter use the slot packages to load data into the slotted tables within Banner ODS. The groups of jobs defined by this parameter let you easily run one job that, for example, loads all of the Financial Aid slot slotted tables.

This parameter is delivered with one entry for each package that loads or updates data in a slotted table in Banner ODS. The actual program name for the slot package occupies the **Description** field and is associated with an ETL group name in the **Internal Code 1** field. For example, when you run the LOAD_FINANCIAL_AID job from the Schedule a Process option in the Administrative UI, the slot packages associated with each ETL SLOT PACKAGE entry that has an **Internal Code 1** of LOAD_FINANCIAL_AID is run.

The following table shows the entries for ETL Slot Package entries that have an **Internal Code 1** value of LOAD_FINANCIAL_AID. The table illustrates a sample of the values as delivered. This is just a sample. The second row gives a definition of each field.

Internal Group: ETL SLOT PACKAGE

Internal Code 1	Internal Code 2	Internal Code Seq.	External Code	Description
The ETL group to which the mapping is assigned.	This field is not used.	Order to run the mapping within job group.	Package.procedure	Slot Package name.
LOAD_FINANCIAL_AID		1	MRKBCMP.P_POPULATE('L')	MRT_FINAID_BUDGET_COMP_SLOT
LOAD_FINANCIAL_AID		2	MRKTREQ.P_POPULATE('L')	MRT_TRACKING_REQUIREMENT_SLOT

Set up the ETL SLOT PACKAGE Parameter

You should not change any delivered values for this parameter. If you want to define a new ETL group of slot packages, you can create new entries with a new ETL group name in the **Internal Code 1** field. Then specify the slot packages that you want to include in the group. Create one new entry for each package in the group.

ETL CONTROL GROUP Parameter

This parameter gives you the ability to load or refresh all the data in your Banner ODS by running one job. The parameter is used in conjunction with the ETL Map Package parameter to further combine groups of jobs into one job.

As delivered, the ETL Control Group parameter defines which groups of job mappings, defined by the ETL Map Package parameter, to run when you run the LOAD_ALL and REFRESH_ALL jobs.

This parameter is delivered with one entry for each ETL group defined by the ETL Map Package parameter. The actual ETL group name (e.g., LOAD_AR, LOAD_FINANCE, LOAD_GENERAL, etc.) occupies the Internal Code 2 field. Each entry is associated with either the LOAD_ALL or REFRESH_ALL control group job in the **Internal Code 1** field. The External Code field for each record has the value *Y*, which means that all jobs (mappings) defined by the group are run when you run the LOAD_ALL job.

The following table shows the entries for the ETL Control Group when the value of **Internal Code 1** is LOAD_ALL. The table illustrates a sample of the values as delivered. This is just a sample. The second row gives a definition of each field.

Internal Group: ETL CONTROL GROUP

Internal Code 1	Internal Code 2	Internal Code Seq.	External Code	Description
Control Group to which the ETL group is assigned.	ETL Group Name Map to **Internal Code 1** of the `ETL MAP PACKAGE` related to this ETL group.	Order to run the ETL group within the control group.	Controls whether to run the group. *Y* = run ETL group *N*=do not run ETL group	A description of the record. This values is not used in processing.
`LOAD_ALL`	`LOAD_ADVANCEMENT`	1	Y	Advancement Load ETL Control Record
`LOAD_ALL`	`LOAD_AR`	2	Y	AR Load ETL Control Record
`LOAD_ALL`	`LOAD_FINANCIAL_AID`	3	Y	FinAid Load ETL Control Record
`LOAD_ALL`	`LOAD_FINANCE`	4	Y	Finance Load ETL Control Record
`LOAD_ALL`	`LOAD_GENERAL`	5	Y	General Load ETL Control Record
`LOAD_ALL`	`LOAD_HUMAN_RESOURCES`	6	Y	HR Load ETL Control Record
`LOAD_ALL`	`LOAD_STUDENT`	7	Y	Student Load ETL Control Record

Set up the ETL CONTROL GROUP Parameter

Review all of the entries delivered for this parameter. If your institution doesn’t maintain some of the areas of Banner ODS data, change the **External Code** value to *N* for those areas. For example, if your institution doesn’t use Advancement and Human Resources, change the **External Code** value to *N* for entries that have **Internal Code 2** values of `LOAD_ADVANCEMENT` or `LOAD_HUMAN_RESOURCES`.

Create an ETL Control Group

You can define a new ETL Control Group by creating a new set of related ETL Control Group parameter entries. Create one new entry for each ETL Group you want to include in the Control Group using the following steps.

1. Create new entries with a new Control Group name in the **Internal Code 1** field.

2. Specify in the **Internal Code 2** field the ETL Groups that you want to include in the Control Group.

3. Specify that you want to run each ETL Group by entering a *Y* in the **External Code** field.

4. Specify the order in which to run the ETL Groups in the **Internal Code Sequence Number** field.

5. Enter a **description** for the new Control Group entry.

6. Add the new ETL Control Group to the list of subprocesses you can schedule by creating a new Subprocess parameter with the following values:

   6.1. **Internal Group**: Subprocess

   6.2. **Internal Code 1**: MAPGROUP

   6.3. **Internal Code 2**: blank

   6.4. **Internal Code Sequence Number**: Number indicating the order in which to run this subprocess.

   6.5. **External Code**: the new ETL Control Group name you created. This is the value in **Internal Code 1** in the ETL Control Group entries created above.

   6.6. **Description**: Name of the subprocess (ETL Control Group job) as it will appear on the Select a Subprocess administrative page.

EVENT parameter

An Event is a logical point in time when you extract information from the source system and load it into the Banner ODS, essentially freezing the data and giving you a snapshot of the data at that point in time. You can then use these data snapshots for reporting purposes. A logical point in time refers to a conceptual time, not an actual calendar date. For example, a logical time to extract financial information may be at the end of each month or for student registration data on the census date for the academic period.

Before you run the processes that load data into Banner ODS, you need to define events that are relevant for your institution’s business needs. This parameter will create the Event parameter list when you Freeze Multiple Banner ODS Tables/Views.

EVENT-EDW parameter

An Event is a logical point in time when you extract information from the source system and load it into the Banner EDW, or a point in time when you snapshot a set of Banner EDW operational star data to be used for reporting purposes essentially freezing the data and giving you a snapshot of the data at that point in time. A logical point in time refers to a conceptual time, not an actual calendar date. For example, a logical time to extract financial information may be at the end of each month or for student registration data on the census date for the academic period.

Create an EVENT-EDW parameter record

Before you run the processes that load data into Banner EDW, you need to define events that are relevant for your institution's business needs. Within the Banner EDW, stars are frozen by Business Concept, meaning all stars for a Business Concept are frozen with the same event.

Create EDW-related events by creating EVENT-EDW Parameter records as follows:

1. Click **Options** from the Administrative UI menu.
2. Click **Set Up Parameters**.
3. Click **Create**.
4. Enter the information for an EVENT-EDW parameter as follows.
   Refer to the section "Considerations when creating Events" when defining the Category, Type, and Description information for events.

Field	Value
Internal Group	`EVENT-EDW` The Internal Group value must always be `EVENT-EDW` for every EDW-related event parameter you define.
Internal Code 1	The Category for the EDW Event
Internal Code 2	The Type for the EDW Event
Internal Code Seq Number	Sequence Number for the event within the Category/Type
External Code	The Name for the Event
Description	Description of the event

5. Click **Save**.

Load Event dimension

When you create new EDW Events, they need to be loaded into the Event dimension.To do this, run theEvent Dimension job using the following menu options.

1. Click **Options** from the Administrative UI menu.
2. Click **Schedule a Process**.
3. Click **Schedule Banner EDW Operational Mappings**.
4. Click**Event Dimension**.

Considerations when creating Events

When defining your institution’s sets of Category, Type and Description information related to Events, you need to consider the following aspects of the data and underlying structures.

Business Concepts share supporting operational stars

If you use the same Event for more than one Business Concept, you could possibly overwrite data. For example, if you were to use the same Event and freeze one Business Concept on Monday then freeze another Business Concept on Tuesday, the second freeze would overwrite the first freeze for any shared stars. For this reason, you may want to consider creating events with unique names by Business Concept.

Can different users or departments share same Events?

You should consider whether different users, departments, or other groups need unique event names or can share event names. This means if different departments will be freezing the same Business Concept, you would not want departments to use the same event name for different data and overwrite each other’s data.

Performance client freezing scorecard data?

If a you use the Performance products, consider whether you’ll use frozen data within the scorecard and what events you’ll need for this.

PARAMETER Parameter

The Parameter parameter is a processing parameter named “Parameter.” This parameter defines the parameters that you must enter at runtime when you Schedule a Process. Basically, all values set up with the Internal Group of “Parameter” and the same Internal Code 2, display on the Schedule a Process page as the runtime parameters for the job defined by that Internal Code 2 value. The values of this parameter are stored in the MTVPARM table.

For example, when you freeze data in a Banner ODS table, you need to specify which table to freeze and the name you want to give the frozen table. Those two parameters are defined by the first two rows in the table below. When you run any of Banner EDW load jobs, you must supply runtime parameters. Which parameters you need to supply are defined here.

Example.

The Load Banner EDW Employee job uses five runtime parameters—Source Institution, Event, Calendar Year, Calendar MOnth and Replace Event. The last four rows in the table below define some of these parameters. The **Description** field supplies the actual prompt that appears for the parameter on the Schedule a Process page. The **Internal Code 2** defines the job with which the parameter is associated, in this case the `LOAD_EDW_EMPLOYEE` job. The **Internal Code 1** defines the parent menu option (on the Select A Process menu) from which you chose the Load Banner EDW Employee job.

This parameter is delivered with one entry for every process parameter. The following table illustrates a sample of the values as delivered. This is just a sample. The second row gives a definition of each field.

Internal Group: `PARAMETER`

Internal Code 1	Internal Code 2	Internal Code Seq.	External Code	Description
External Code of the process that is parent to this subprocess.	**External Code** of subprocess for this parameter.	Order for entries with same **Internal Code 2**.	Short description of parameter. This value is used in the processing code. Short description of parameter. This value is used in the processing code.	Actual parameter field prompt that appears on **Schedule a Process** administrative page.
`ADHOC_FREEZE`		1	`TABLE_NAME`	Enter Table to Freeze
`ADHOC_FREEZE`		2	`TABLE_HISTORY`	Enter Table Name to Freeze to
`MAPGROUP-EDW`	`LOAD_EDW_EMPLOYEE`	1	`EVENT_IN`	Event:
`MAPGROUP-EDW`	`LOAD_EDW_EMPLOYEE`	2	`CALENDAR_YEAR_IN`	Select the calendar year

MAPGROUP-EDW	LOAD_EDW_EMPLOYEE	3	CALENDAR_MONTH_IN	Select the calendar month
MAPGROUP-EDW	LOAD_EDW_EMPLOYEE	4	REPLACE_IND_IN	Replace Event (Check for 'Yes'):

Note

If the Internal Code 2 field is left blank, the parameter appears for all subprocesses under the parent process in Internal Code 1. For example, the Enter Table to Freeze parameter does not have an entry for Internal Code 2. This parameter value appears for all subprocesses under the `ADHOC_FREEZE` (Internal Code 1) area ■

Set up the PARAMETER Parameter

The only existing values you should change for this parameter are the descriptions. If you want to change the name of a parameter that appears on the Schedule a Process page, change its description.

If you want to add a process to the Schedule a Process page and it requires input parameters, you need to define the parameters by adding new values for this Parameter parameter.

Create Runtime Parameters to Scheduled Processes

You may add new processes to the Administrative UI that require runtime parameters, or you may want to add runtime parameters to existing processes, for example, a defined Freeze Data list. Create a new record for this Parameter to define a runtime parameter.

The easiest way to understand how to create a new runtime parameter is to review the existing ones. In the previous table, there are two records whose Internal Code 1 = FREEZE_TABLE and the Internal Code 2 field is blank. Each of these records defines a runtime parameter that appears on the Schedule a Process page when Freeze Multiple Banner ODS Tables/Views is selected. The Internal Code 1 field of FREEZE_TABLE on the Parameter record here matches to the External Code of FREEZE_TABLE on the INSTALLED PROCESS parameter.

Example

If you add a runtime parameter to a freeze data list called TEST1, the following steps show the field values needed to create this new parameter.

1. Enter **Internal Group** = *PARAMETER*.

2. Enter **Internal Code 1** = *FREEZE_TABLE*. The parent process for the TEST1 freeze data list.

3. Enter **Internal Code 2** = *TEST1*. The actual name of the freeze data list to associate the parameter.

4. Enter **Internal Code Sequence Number** = *2*. The order that parameters are listed at runtime. You can add up to two parameters to a freeze data list.

5. Enter **External Code** = *ACADEMIC_PERIOD*. The actual field value that you want the user to supply at runtime.

6. Enter **Description** = *Enter Term Code*. The prompt that a user needs to supply at runtime.

7. Choose **PARAMETER Type** = *SELECT*. Identifies how the user enters the runtime parameter. The field accepts four values:
   - SELECT = User must supply a valid PL/SQL statement.
   - DATE = User must supply a valid date.
   - EDIT = User can supply a text string.
   - CHECKBOX = User must check or uncheck an option.

8. Enter **PARAMETER SQL**. This field is only required when the **PARAMETER Type** is *SELECT*. Enter a valid PL/SQL statement, which is used to populate the valid field values to display in the drop-down list of the runtime prompt.

9. Enter **PARAMETER SQL Delimiter**. This field is only required when the **PARAMETER Type** is *SELECT* and you use a delimiter in the **PARAMETER SQL** field. Specify the delimiter used in the **PARAMETER SQL** field.

List Events for a Banner EDW Star (Banner EDW)

Use this option to view a list of events that were loaded for each Banner EDW Star. All the time dimension information displays for job events which have been run.

The EDW_EVENT_LISTING parameter determines which columns you want to display on the List Events for a Banner EDW Star page.

1. Click **Options** from the Administrative UI menu. The Options menu opens.

2. Click **List Events for a Banner EDW Star**. The List Events for a Banner EDW Star page opens.

3. Choose the star you want to review from **Banner EDW Star** drop-down list on the List Events for a Banner EDW Star page.

4. Select Show SQL if you want to view the SQL for the query when you view the list of events.

5. Click **List Events**.

Banner ODS Utilities

The Utilities process contains utility jobs or processes that perform various administrative tasks, and provide ongoing maintenance of the Banner ODS. For example, the Utilities option enables you to compare the number of rows in one table or multiple tables in the source system with the number of rows in the composite tables in Banner ODS. You can also check for potential problems that may cause performance issues.

Once a job is completed, a control report is created. When discrepancies are found, the control report indicates the number of records found in each object, as well as the key values for the records that are not synchronized.

See “Schedule a Process (Banner ODS and Banner EDW)” on page 4-65 for instructions on how to schedule processes.

Report Banner ODS Source Change Table Counts

The Report Banner ODS Source Change Table Counts produces a control report that calculates how many rows are in each of the source system Change tables for each Banner ODS Composite table. This enables you to monitor the accumulation of Change records for a particular Composite table.

Depending on how many rows are in a Change table for a given Banner ODS Composite table, it may be more efficient to run a Load process instead of the Refresh process. Determining which process to run is a matter of individual experience withtimes for various Composite Tables.

Change Tables and Control Reports

The row count totals that are reported in a Control Report for a given mapping reflect all rows that OWB accesses during the processing of that mapping. So, for a given DELETE mapping, OWB may SELECT 500 records, and then DELETE 500 records. Alternatively, if there is a filter condition applied in the mapping, the number SELECTED and DELETED (and/or UPDATED) may vary, based on the filter condition. Typically, these filter conditions occur to “filter” out change records associated with other processes. As an example, the SPRPCHG table processes many PIDM based tables, including SPRIDEN, SPBPERS, SPRADDR and SPRTELE. If the DELETE and UPDATE mappings for the Address process were running, and the SPRPCHG table had 100 records in it for different processes, the DELETE_MST_ADDRESS mapping would “select” all 100 records, but then it would only filter out and pass along those records where the field “SPRPCHG_TABLE_NAME” was equal to “ADDRESS”. Therefore, the delete would only “delete” those records, which would be the subset of the overall “selected” count.

The Change Tables contain the identifying data (sometimes the primary key value) about the records that are modified in the Banner system. Hence, if a record in a Banner table is edited 5 times, only one record will appear in the Change Table for it, which the Refresh process uses to know to bring the entire row of data over to the ODS for that record. Further, some of the Change Tables are used by multiple mappings, so the mappings will typically apply a filter when accessing the Change Table. This means that the row count totals in the Control Reports will usually not match up directly with the Change Table counts, though there are some cases of Changes Tables that are used only by a single mapping and that mapping also doesn't apply filter conditions.

Example

Addresses are managed through the ETL process by PIDM. If a person has 10 addresses and 3 of them change, then there is one record in the SPRPCHG table with the PIDM of the changed record, it has in it the most recent date/time of the last DML activity, and it also reflects the most recent DML action (C- change, D- Delete, or U-update). The delete mapping deletes all 10 addresses for that PIDM from the ODS even though there is only one record in the change table. The update mapping adds all the address records using the AS_ADDRESS composite view where the PERSON_UID field in the view matches the PIDM in the SPRPCHG table (ultimately re-adding all 10 addresses). Thus, a single change table entry results in 10 records being deleted and 10 records being inserted. The Change Table Counter process attempts to handle the change tables that are used by multiple mappings. That is, for a given DELETE mapping, it parses the mapping (PL/SQL) code for the default value of the second parameter (P_TABLE) which it then uses to construct the following SQL: code:

= 'SELECT COUNT(*) FROM ' || parm1Value || '@' || linkName || ' WHERE ' || parm1Value || '_TABLE_NAME=''' || parm2Value || ''''; where parm1Value is the CHG_TABLE value and parm2Value is the P_TABLE value.

See “Schedule a Process (Banner ODS and Banner EDW)” on page 4-65 for instructions on how to schedule processes.

Synchronize Comments for Multiple Reporting Views

Run this process to generate comments on multiple reporting views. The meta data Business Definitions for each reporting view and the meta data business definitions for each of the columns is copied from the meta data into the database **Comments** field.

This process is scheduled from Banner ODS Utilities menu. (See “Schedule a Process (Banner ODS and Banner EDW)” on page 4-65 for instructions on how to schedule a process.) The same functionality is available by selecting the **Sync Comments** link on the View Target Report page. (See “Synchronize Meta Data Comments with Reporting Views” on page 4-158 for instructions.)

Synchronize Comments for a Single Reporting View

Run this process to generate comments on a reporting view. The meta data Business Definition for the reporting view and the meta data business definition for each of the columns is copied from the meta data into the database **Comments** field.

This process is scheduled from Banner ODS Utilities menu. (See “Schedule a Process (Banner ODS and Banner EDW)” on page 4-65 for instructions on how to schedule a process.) The same functionality is also available by selecting the **Sync Comments** link on the View Target Report page. (See “Synchronize Meta Data Comments with Reporting Views” on page 4-158 for instructions.)

Banner ODS Checks and Balances

Banner ODS Checks and Balances utility processes can be run after an upgrade or intermittently to perform the following information checks.

- Check Mappings and Parameters

  Verifies that all Banner ODS Mapping packages have been created in the database and are valid. This process also confirms that all ETL MAP PACKAGE parameters have a corresponding DELETE*, LOAD* and UPDATE* package (for example, LOAD_MAT_GIFT, UPDATE_MAT_GIFT, DELETE_MAT_GIFT).

- Check Metadata

  Compares the defined total of Banner ODS baseline delivered meta data records to a count of the records in Banner ODS to determine if the meta data records have loaded successfully

- Miscellaneous Checks

  Verifies that the database link to the source system exists and is working.

- Check Indexes

  Identifies the following:

  - Baseline Banner ODS indexes that are missing from the staged tables replicated from the source database.
  - Baseline Banner ODS indexes that are missing from change tables used to incrementally update the ODS composite tables.
  - Baseline Banner ODS indexes that are missing from the target database composite tables.

  If there are local indexes you would like verified by this process, insert the appropriate data into the MGBINDX table.

- Freeze table changes

  As new versions of Banner ODS are released, Reporting views may have new columns added and, in some cases, existing column names changed. Therefore, if you have created freeze table data in earlier versions of Banner ODS, those table structures may become out of sync with newer versions of Reporting views, causing subsequent freeze processes to fail.

This process compares the table structure of any existing freeze table data against the current Reporting view, and any column discrepancies are reported. In addition, the appropriate Oracle 'ALTER TABLE' statement is also provided in the control report so you can resynchronize your freeze tables with the Reporting views.

- Check Triggers

  Identifies any baseline table triggers that are missing from the staged tables replicated from the source database. The purpose of these triggers is to track changes that are needed to incrementally refresh the target database composite tables

See “Schedule a Process (Banner ODS and Banner EDW)” on page 4-65 for instructions on how to schedule processes.

Transfer Banner Fine-Grained Access

Use Transfer Banner Fine-Grained Access to transfer data for Banner Finance Fund, Fund Type, and Organizations, and for Banner Human Resources Organizations and Employee Class from Banner to Banner ODS.

Prerequisites

- “Set up and Maintain Organizational Areas” on page 4-19
- “Banner User ID Translations” on page 4-21
- “Set up Business Profiles” on page 4-24
- “Set up and Maintain Security Rules” on page 4-27
- “Policy Management” on page 4-50
- “Transfer Banner Fine-Grained Access” on page 4-112

1. Click **Options**.
2. Click **Schedule a Process**.
3. Click **Banner ODS Utilities**.
4. Click **Transfer Banner Fine-Grained Access**.
5. Check the boxes that correspond to the fine-grained access security permissions you want to transfer.

6. Choose the **Transfer Mode** to use for the transfer.
   You can choose from the following modes:

   - REPLACE - Replaces all FGA rules of Banner in Banner ODS with new Banner FGA rules. Existing FGA rules in Banner ODS, created using Administrative UI, are not affected.

   - REPLACE_ALL - Replaces FGA rules of both Banner and Banner ODS data with new Banner FGA rules. Existing FGA rules in Banner ODS that were not part of the current Banner data load are not affected.

   - TRUNCATE - Replaces all FGA rules, both Banner data and Banner ODS data, and refreshes Banner ODS with new Banner FGA rules.

7. Click **Submit** to schedule the process to run.

The transfer checks GUBINST to see if Banner Finance or Banner Human Resources is installed. If it is not, then a warning message displays.

Also, if Banner Finance is not installed, then the Banner Finance Fund/Organization transfer is bypassed. If Banner Human Resources is not installed, then both the Human Resources Organizations and Employee Class transfers are bypassed.

The data is transferred using the ODSMGR XXXX@SOURCE_DB database link. Whether the data transfers to Banner ODS or not is based on whether security is turned on in Banner in the following areas:

- Finance: FOBSYSC_FUND_ORG_SECURITY_IND

- Human Resources Organization: PTRINST_ORGN_SECURITY_IND

- Human Resources Employee Class: PTRINST_ECLS_SECURITY_IND

In the Banner ODS Administrative UI (**Options** tab, **Set up Parameters** link) there are three process/job parameters under the Internal Group of BANNER TO ODS FGA TRANSFER which indicate whether or not Banner security settings should affect the job. By default, all three parameters are set to *N* (maintained on the Update a Parameter Administrative UI web page).

In Banner, FOBSYSC_FUND_ORG_SECURITY_IND, from the Banner FOBSYSC table, indicates whether or not Banner Finance Fund and Organizations security is active. For the Banner Finance Fund and Organizations transfer, the BANNER TO ODS FGA TRANSFER parameter with a value for Internal Code 1 of FINANCE FUND/ORG SECURITY ACTIVE determines whether not to consider the value of FOBSYSC_FUND_ORG_SECURITY_IND. If the job parameter has an external Code of *Y*, then this indicates that Banner Finance Fund and Organizations security must be turned on for the Fund/Org transfer to occur. If it is not turned on, a warning message is displayed and the Banner Finance Fund and Organizations transfer is bypassed. If the external Code is *N*, then this indicates to go ahead and run the Finance Fund/Org transfer, regardless of whether Banner Fund/Org security is active.

The BANNER TO ODS FGA TRANSFER parameters with internal Codes of HR ORG SECURITY MUST BE ACTIVE and HR ECLS SECURITY MUST BE ACTIVE perform the same function against PTRINST_ORGN_SECURITY_IND and PTRINST_ECLS_SECURITY_IND respectively. After these parameters have been evaluated, the transfer begins.

At this point the data transfer begins and MGBXWLK is checked. The way that MGBXWLK has been configured, i.e., which of the four set up options you have chosen, determines how value-level data is written to MGBFGAV and column-level data is written to MGBFGAE.

The Finance Fund/Org transfer reads data from FORUSFN and FORUSOR and transfers user permissions for individual Funds and Organizations to IA_ADMIN.MGBFGAV based on the fine-grained access rules in IA_ADMIN.MGBFGAR. For users who have access to either all Finance Funds or all Finance Organizations, data is read from FOBPROF and is written to IA_ADMIN.MGBFGAE. For users who have Fund Type permissions on FORUSFN, the fund numbers associated with each Fund permission, as listed by fund in the Banner ODS Fund Hierarchy table MFT_FUND_HIERACHY, are also written to MGBFGAV.

The Banner Human Resources Organizations transfer reads data from PSRORGN and transfers user permissions for individual Organizations to IA_ADMIN.MGBFGAV based on the fine-grained access rules in IA_ADMIN.MGBFGAR. For users who have access to all Banner Human Resources Organizations, data is read from PTRUSER and written to IA_ADMIN.MGBFGAE.

The Banner Human Resources Employee Class transfer reads data from PSRECLS and transfers user permissions for individual Employee Classes to IA_ADMIN.MGBFGAV based on the fine-grained access rules in IA_ADMIN.MGBFGAR. For users who have access to all Banner Human Resources Organizations, data is read from PTRUSER and written to IA_ADMIN.MGBFGAE.

See “Schedule a Process (Banner ODS and Banner EDW)” on page 4-65 for instructions on how to schedule processes.

Materialized Views

A materialized view (MVIEW) is the actual physical implementation of the logic within a reporting view. Within the Administrative User Interface, you can create a materialized view based on its underlying reporting view. The system performs all the calculations and business logic of the reporting view and stores the resulting data in a Banner ODS database table.

A materialized view is a snapshot of the data from the source database at the time the view was materialized or later refreshed. The nightly refresh process will update the data in a materialized view to make it current as of that last build.

The benefit of creating materialized views is that queries run against a materialized view can perform significantly faster than queries run against the underlying reporting view because the calculations required by the base reporting view have already been performed and stored in the database.

By contrast, some reporting views are very complex. They access multiple tables joining data from other views and using business logic like database functions that require many computations. Each time you execute a query against a complex reporting view the system needs to perform the joins and the business logic of that view, which can result in longer run and response times for the query.

Each table associated with a materialized view is a simple table like one you would create from performing a “CREATE TABLE AS SELECT * FROM <view>” operation. However, the MVIEW table has various internal database attributes, such as indexes, that are hidden. The MVIEW also allows you to index the columns most likely used for reporting and to use query rewrite in the database.

Materialize a View

Materializing a reporting view simply means creating a materialized version of the view and replacing the existing reporting view with the materialized view. The materialized view keeps the same name as the reporting view so that all existing reporting accesses, such as reporting tool meta data and reports, continue to function correctly regardless of whether or not the view is materialized.

The reporting view and its associated materialized view cannot exist in the database at the same time. For this reason, when you create a materialized view, the source code for the associated reporting view is stored in the IA_ADMIN.MGBMVEW table. This table also stores specific information, such as elapsed time and row count, about the materialized view. This information is used to create and refresh the materialized version of a view. When you delete a materialized view, the system uses the information about that view stored in the MGBMVEW table to recreate the reporting view. Then the row related to that view is deleted from the MGBMVEW table.

You perform the tasks of creating, refreshing, and deleting materialized views using the Administrative User Interface (UI). It is possible to use the API calls defined in the MGKODSU package to manipulate the materialized views outside of the Administrative UI. However, any changes you make directly to the package or the APIs are not supported.

Materialized View Considerations

Though you can create a materialized version of any existing reporting view, there are a number of factors to consider when deciding whether to convert a view from a reporting view to its materialized version. Converting a large or complex reporting view that performs slowly to a materialized view can improve reporting performance. However, the improved performance will introduce additional processing time to create and refresh the materialized view and it will require extra disk space to store the materialized data.

You'll want to consider these factors when determining which reporting views to materialize. In addition, you need to weigh these considerations against other standard performance tuning techniques (like creating additional indexes on underlying tables, specifying Oracle database parameter settings, and performing hardware upgrades) when deciding which reporting views to materialize.

Materialized Views and Patch Releases

There will be times when you need to apply a patch release which includes changes that affect the reporting views. If a reporting view has been materialized in the system and you apply a patch release, application of the patch may fail since Oracle won't overwrite an existing materialized view with a reporting view of the same name. In these cases, you will need to delete the materialized view before you apply the patch release then recreate the materialized view.

Refer to the documentation delivered with a particular patch release for information about dependent objects. That documentation will direct you on how to determine which materialized views you need to delete and then recreate before and after applying the patch.

Create an Index of Materialized View

The ODS Materialized View Create process can create indices on the materialized view. The value set for the Recommended Search Columns in the ODS Meta Data determines whether to create a materialized view index. Generally, the Recommended Search Column value(s) are set in the meta data for that reporting view before it gets materialized, so that when it get materialized (during the CREATE process), indexes are added appropriately.

If you want to create an index of a materialized view, where no Recommended Search Columns are defined, you can add them prior to creating the materialized view.

To create an index of the materialized view, perform the following steps:

1. Log on to the Administrative User Interface.

2. Click the **Meta Data** tab.

3. Click **Maintain Banner ODS Meta Data** link. The View Target Report List page opens.

4. Choose the reporting view for which you want to create an Index.

    4.1. Click the **Select** link next to the Subject Area.

    4.2. Choose a Report Type and Subject Area.

    4.3. Select the view for which you want to add an index.

For example, to choose the ACADEMIC_STUDY reporting view, click the **Select** link next to the Subject Area, choose Student, and select the ACADEMIC_STUDY reporting view.

5. Click the **Properties** link.
6. In the **Recommended Search Columns** field, enter the columns you want to be in the index. The columns list must be separated by a comma and then a space. If you want more than one index, you can do that too, by entering <BR> and then the next column list.
7. Click **Save Changes.** This will save a local version of the metadata and not affect any baseline data. The Admin UI will show you have local changes by displaying the headers in a different color. (You can refer to other definitions to see examples.)
8. Create the Materialized View. It will generate an index for the columns that you specified. For more information on creating materialized view, see "Create a Materialized View".

Create a Materialized View

A materialized view (MVIEW) is the actual physical implementation of the logic within a reporting view. Within the Administrative User Interface, you can create a materialized view based on its underlying reporting view. The system performs all the calculations and business logic of the reporting view and stores the resulting data in a Banner ODS database table. You may want to create a materialized view because queries typically run faster against a materialized view. Perform the following steps to create one or more materialized views using the Administrative User Interface (UI).

1. Click **Options** from the Administrative UI menu.
2. Click **Schedule a Process**.
3. Click **Banner ODS Utilities**.
4. Click **Create Materialized Views**.
5. Select the **Reporting Views** that you want to materialize. Use Shift-click to select a contiguous range of views or Ctrl-click to select noncontiguous views.

   There is a 3800 character limit on the values you can select. As you select values, you can see the total "Selected size" in the status bar at the bottom of the window. If you want to choose values that total more than 3800 characters, break the selections into multiple groups and run the jobs separately.
6. Enter a **Run Date** (format dd-mon-yyyy) and **Run Time** (format hh24:mi:ss). Enter NOW in each field to run the job immediately.

7. Click **Submit**.

When the job completes running, you can review its activity by clicking **View Control Reports** at the bottom of the Select a Process page.

Maintain a Materialized View

Typically, your institution implements a process each night that updates the Banner ODS data to synchronize it with the current source Banner data. This process runs the refresh jobs that update any existing materialized views that have been created, changed or deleted since the last refresh. You may want to refresh a materialized view on a more timely basis. Refreshing the view repopulates the associated database table with current data. Perform the following steps to refresh a materialized view on an as needed basis using the Administrative User Interface (UI).

1. Click **Options** from the Administrative UI menu.
2. Click **Schedule a Process**.
3. Click **Banner ODS Utilities**.
4. Click **Maintain Materialized Views**.
5. Select the **Materialized Views** that you want to refresh. Use Shift-click to select a contiguous range of views or Ctrl-click to select noncontiguous views.

   There is a 3800 character limit on the values you can select. As you select values, you can see the total “Selected size” in the status bar at the bottom of the window. If you want to choose values that total more than 3800 characters, break the selections into multiple groups and run the jobs separately.
6. Select Refresh from the **Materialized Views Action** drop-down list.
7. Enter a **Run Date** (format dd-mon-yyyy) and **Run Time** (format hh24:mi:ss). Enter NOW in each field to run the job immediately.
8. Click **Submit**.

When the job completes running, you can review its activity by clicking **View Control Reports** at the bottom of the Select a Process page.

Delete a Materialized View

Deleting a materialized view removes it from the database and replaces it with the original reporting view. Perform the following steps to delete a materialized view using the Administrative User Interface (UI).

1. Click **Options** from the Administrative UI menu.
2. Click **Schedule a Process**.
3. Click **Banner ODS Utilities**.
4. Click **Maintain Materialized Views**.
5. Select the **Materialized Views** that you want to delete. Use Shift-click to select a contiguous range of views or Ctrl-click to select noncontiguous views.

   There is a 3800 character limit on the values you can select. As you select values, you can see the total “Selected size” in the status bar at the bottom of the window. If you want to choose values that total more than 3800 characters, break the selections into multiple groups and run the jobs separately.
6. Select Drop from the **Materialized Views Action** drop-down list.
7. Enter a **Run Date** (format dd-mon-yyyy) and **Run Time** (format hh24:mi:ss). Enter NOW in each field to run the job immediately.
8. Click **Submit**.

When the job completes running, you can review its activity by clicking **View Control Reports** at the bottom of the Select a Process page.

Materialized View Status Report

The Materialized View Status report (REPORT_MVIEW_STATUS) is a process that you can run to view basic information about each materialized view currently defined in the system. The status report includes the following information about each materialized view: creation date and time, elapsed creation time, and the number of rows created. If the view was refreshed, the report also includes the latest refresh date time, elapsed refresh time, and the number of rows refreshed.

Report Materialized View Status

The Materialized View Status report includes information about each materialized view currently defined in the system. Run the report if you want to know when a materialized view was created or refreshed, how long it took to create or refresh a view, or how many rows were created or refreshed in a view. Perform the following steps to run the Materialized View Status report.

1. Click **Options** from the Administrative UI menu.
2. Click **Schedule a Process**.
3. Click **Banner ODS Utilities**.
4. Click **Report Materialized View Status**.
5. Enter *NOW* in the **Run Date** and **Run Time** fields.
6. Click **Submit**.

When the job completes running, you can review its activity by clicking **View Control Reports** at the bottom of the Select a Process page.

Materialized View Control Reports

Each time you create, refresh, or delete a materialized view, the CREATE_MVIEWS or MAINTAIN_MVIEWS process creates a control report. The report includes information about creating, refreshing or dropping the materialized view and creating or dropping the related reporting view as appropriate. The report also includes the date and time that each action within the job completed.

You can save the control report as a .csv (comma separated values) file. You can open and review the .csv file in a spreadsheet application like Microsoft Excel. This option is especially useful for viewing large control reports. Click the CSV Summary button on the Display a Control Report page to view or save the report in .csv format.

For most reporting views, the following ORACLE command is used to create the associated materialized view:

```
create materialized view <view> BUILD IMMEDIATE REFRESH
COMPLETE ON DEMAND;
```

Some reporting views use different Oracle syntax to create the associated materialized view. The control report notes these view exceptions when the materialized view is created. Although a view may be created using alternate syntax, the resulting materialized view is functionally the same as the reporting view.

Banner EDW Utilities

The Banner EDW Utilities item includes utility jobs or processes that perform administrative tasks and provide ongoing maintenance of the Banner EDW. Once a job is completed, a control report is created. When discrepancies are found, the control report indicates the number of records found in each object, as well as the key values for the records that are not synchronized.

See “Schedule a Process (Banner ODS and Banner EDW)” on page 4-65 for instructions on how to schedule processes.

Banner EDW Checks and Balances

Banner EDW Checks and Balances reports on the state of Banner EDW. This process could be run after an upgrade or intermittently if you suspect an error. Options are available to ensure the following:

- Mapping packages exist and are valid in the database
- Mapping parameters exist for each mapping package
- Delivered indexes exist on Banner EDW dimension and fact tables
- Meta data records have been created

See “Schedule a Process (Banner ODS and Banner EDW)” on page 4-65 for instructions on how to schedule processes.

Load Banner EDW Data Default Cleansing Values

The Load Banner EDW Data Default Cleansing Values job loads information that is used to cleanse the data associated with the operational and snapshot stars that are part of the original warehouse architecture (stars that existed before Banner EDW 8.2/8.2a.) You should run this job before you load the these stars. You can load a single data element’s default cleansing values, or load all data elements’ default cleansing values.

See “Schedule a Process (Banner ODS and Banner EDW)” on page 4-65 for instructions on how to schedule processes.

Load Banner EDW Dimensional Data Default Cleansing Values

The Load Banner EDW Dimensional Data Default Cleansing Values job loads information that is used to cleanse the data associated with the dimensional stars that are part of the enhanced warehouse architecture (stars delivered new in Banner EDW 8.2/8.2a or later releases.) You should run this job before you load the dimensional stars into the warehouse. You can load a single data element's default cleansing values, or load all data elements' default cleansing values.

See “Schedule a Process (Banner ODS and Banner EDW)” on page 4-65 for instructions on how to schedule processes.

View Control Reports (Banner ODS and Banner EDW)

When a process runs, it creates a control report that details the progress, status, and errors in the process. Each control report highlights items like run time errors, record counts, and the job status for the process submitted.

Follow the steps below to review the control reports to determine whether a process ran successfully and to view errors.

1. From the Administrative menu, click **Options**.
2. Click **View Control Reports**. The Select a Control Report page opens.
3. On the Select a Control Report page, find the process you want to review in the list. Check the Status column to see if the process ran successfully. If the status is ERROR, there was a problem with the process.
   - To sort the list of control reports click one of the column headings - Run Date, Job Number, Process, User ID, or Status.
   - To filter the list of control reports, select the filter button next to one of the column headings (Run Date, Process, User ID, or Status), select the filter values and click **Select**. You can only apply one filter at a time.
4. Click **Refresh Job Status Codes** to see the most current job status. Often a job status will change from Running to Completed.

   The **Refresh Job Status Codes** button is helpful with jobs that have been terminated in the database (due to a shutdown, or other error, etc.). If a job is terminated in the database, it locks the status as Running on the View Control Reports page. Therefore, if you click this button you not only refresh all status codes, but also ensure that any Terminated status codes display correctly.

   To delete a control report, select the corresponding checkbox in the Delete column. To select or clear all the control reports, click **Select All** or **Deselect All**.

5. To review additional information on how a process ran, click the link for that process from the Process column. The Display a Control Report page opens.

   5.1. Click **View error message(s)** to view the first error message.

   5.2. Click **Next error** to browse all errors for the job.

A description of each button on the Display a Control Report page appears below:

Button	Description
Filter Report	To view only selected lines of a report. The Select Report Filters window opens. Check the box next to the filter phrases that you want to see in the report. (Select All and Deselect All links select or clear all lines on the report.) Click **Filter Report**. To redisplay the information, click **Show Detail**. To filter the report for certain leading words/phrases, enter one or more delimiter characters and click **Scan**. (The default delimiter is a colon.) The page then displays all the unique occurrences of text up to the first occurrence of that delimiter, which you can use to filter the Control Report.
CSV Summary	Click to save a control report to a .csv (comma separated values) file that you can open and review in a spreadsheet application like Microsoft Excel. This option may be especially helpful for viewing large control reports. **Note:** This option is primarily intended to use with output from LOAD and REFRESH jobs, reporting the number of rows processed for each mapping. There are also CSV outputs specific to the Reconcile utilities, as well as the Change Table Counts utility process.
Reschedule Process	Click to open the Schedule a Process page and reschedule a process.

Error Messages

This section lists some of the error messages you may encounter on the control report for any process. Not all error messages are documented, so this is not a complete list.

Some error messages identify data cleansing Warning messages and they show the number of rows written to the `WTT_<star schema name>_ERROR` table. The line under the

Warning identifies the `DATA_ELEMENT` and data value that caused the cleansing error. Correct the cleansing error and run the `FIX EDW <star name>`.

Banner ODS Checks and Balances Process

Warning: Obsolete sequence numbers in MGBPSQL

Reason: Each row in the IA_ADMIN.MGBPSQL table should have a corresponding row in the IA_ADMIN.MTVPARM table matching on the sequence number. Any unmatched rows in MGBPSQL are reported

Action: Sequence numbers that exist in MGBPSQL but not in MTVPARM should be deleted from the table

Error: <mapname> is INVALID in the database

Reason: A delivered ETL mapping (PL/SQL) package currently has an INVALID status, and will not run during any of thejobs.

Action: Recreate the mapping package in the ODSMGR schema of the database.

Warning: <mapname> parameter does not have corresponding MAPPING

Reason: Baseline ETL mapping packages, that have been created with Oracle Warehouse Builder, exist in the ODSMGR schema with a name starting with “LOAD_”, “DELETE_”, “UPDATE_”. Each package has a corresponding Parameter record with the same name. This warning indicates that a parameter exists for the specified mapping, but the actual package does not exist in the ODSMGR schema of the database.

Action: For baseline packages, create the ETL mapping package in the ODSMGR schema. If the mapping package does not exist in the database, use the Administrative UI to remove the parameter.

Warning: <mapname> mapping package does not have corresponding parameter record

Reason: The baseline Banner ODS ETL mapping packages, that have been created with Oracle Warehouse Builder, exist in the ODSMGR schema with a name starting with “LOAD_”, “DELETE_”, “UPDATE_”. Each package has a corresponding parameter with the same name. Without this parameter record, the mapping will not be run during any of thejobs.

Action: Create the parameter record for the <mapname> package, similar to the other ETL MAP PACKAGE parameter records. Note: If the mapping is a locally developed package, consider using a different naming standard (ex: 'MY_LOAD_%', 'MY_DELETE_%'), OR create a different schema for local modifications.

ERROR: Parameters not loaded for Banner ODS mappings (ETL MAP PACKAGE)

Reason: The mapping parameters for Banner ODS have not been created in MTVPARM.

Action: Check with technical staff to create the missing entries.

Warning: ---> <view name> is documented but does not exist

Reason: This check will verify that all reporting views documented in the metadata actually exist in the database. The warning message reports views that do not exist in the database.

Action: Check with technical staff to create the missing view in the ODSMGR schema of the database.

Warning for REPORTING View: <view name> WARNING: ---> MetaData column missing in view: <column name>

Reason: Baseline reporting views are delivered with corresponding metadata for each view column. The column that is documented does not exist within the view.

Action: Check with technical staff to determine why the column is missing from the view, and recreate the view if necessary.

Note

Client developed reporting views can be imported into the metadata using the Administrative User Interface. If the column should not be documented for a locally developed view, use the Administrative UI to remove the metadata. ■

Warning for REPORTING View: <view name> WARNING: ---> View column missing in MetaData: <column name>

Reason: Baseline reporting views are delivered with corresponding metadata for each view column. The column exists n the view, but is not documented in the metadata

Action: Check with technical staff to determine why the column is not documented in the metadata. Document the missing column with the Administrative UI.

Note

Client developed reporting views can be imported into the metadata using the Administrative User Interface. If the column should not be documented for a locally developed view, use the Administrative UI to create the metadata. ■

Baseline index <index_name> is missing from table

Reason: Delivered index names are stored in the IA_ADMIN.MGBINDX table. Any missing indexes may impact Banner ODS performance and are reported

Action: Create the missing index to ensure optimum system performance.

Additional index (index_name) found for table

Reason: Local indexes that do not exist in the IA_ADMIN.MGBINDX table are reported.

Action: To eliminate the warning message from the control report, insert the index information into MGBINDX with local = YES.

Warning: More than one database link found as source location for OWB

Reason: Verify that only one source database is identified for the OWB.

Action: Remove or rename incorrect database links from Banner ODS database. (Search DBA_DB_LINKS where LINK_NAME like '%SOURCE_DB%' to identify these).

WARNING: Use the Freeze Data Maintenance page to remove these columns from the freeze table <freeze_table>

Reason: It is possible to select columns to include in the freeze data. If a column that has been used in a freeze table is no longer valid in the source, a warning message is provided

Action: Use the Freeze Data Maintenance page in the Administrative UI to locate the freeze table and review the selected columns. Remove the obsolete columns from the selected columns list.

Freeze Table <freeze_table> does not exist. Used in Freeze Data List <freeze_data_list>

Reason: Freeze data lists are created to freeze multiple tables

Action: Review tables in the Freeze Data Lists reported to determine why the freeze data has not been generated.

ERROR: AR dblink test failed

Reason: A query from AT_AR_DEPOSIT view in Banner database failed.

Action: If the database link is valid, verify that the listed view exists in Banner database

ERROR: ADVANCEMENT dblink test failed

Reason: A query from a single Advancement view in Banner database (AA_CONSTITUENT) is done as a check that the system configuration is correct for Advancement ETL mapping packages to execute. The error message would indicate that the database link is incorrect or the view is not valid.

Action: Verify that Advancement views have been created in Banner database

ERROR: FINANCE dblink test failed

Reason: A query from a single Finance view in Banner database (AF_PURCHASE_ORDER_ACCOUNTING) is done as a check that the system configuration is correct for Finance ETL mappings to execute. The error message would indicate that the database link is incorrect or the view is not valid.

Action: Verify that Finance views have been created in Banner database.

FINANCIAL AID dblink test failed

Reason: A query from a single Financial Aid view in Banner database (AR_AWARD_BY_PERSON) is done as a check that the system configuration is correct for Financial Aid ETL mappings to execute. The error message would indicate that the database link is incorrect or the view is not valid.

Action: Verify that Financial Aid views have been created in Banner database.

ERROR: COMMON dblink test failed

Reason: A query from a single view in Banner database (AS_PERSON) is done as a check that the system configuration is correct for Common ETL mappings to execute. The error message would indicate that the database link is incorrect or the view is not valid.

Action: Verify that this view and other General views have been created in Banner database.

ERROR: HR dblink test failed

Reason: A query from a single Human Resources view in Banner database (AP_REVIEW) is done as a check that the system configuration is correct for Human Resources ETL mappings to execute. The error message would indicate that the database link is incorrect or the view is not valid.

Action: Verify that Human Resources views have been created in Banner database.

ERROR: STUDENT dblink test failed

Reason: A query from a single Student view in Banner database (AS_COURSE_CATALOG) is done as a check that the system configuration is correct for Student ETL mappings to execute. The error message would indicate that the database link is incorrect or the view is not valid.

Action: Verify that Student views have been created in Banner database.

Banner EDW Checks and Balances Process

Error: mapname is INVALID in the database

Recreate the mapping package in the database.

Warning: mapname parameter does not have corresponding MAPPING

If the parameter is valid, create the missing mapping package in the database. If the mapping package does not exist in the database, use the Administrative UI to remove the parameter.

View column missing in MetaData

The column name from the error message does not have meta data in Banner EDW. Choose the menu options Meta Data and Maintain Banner EDW Meta Data to add the missing information.

MetaData column missing in table

The table name from the error message has meta data in Banner EDW, but does not exist in the actual table. Verify that the table exists, and that the column is spelled accurately in the meta data. Use the Meta Data and Maintain Banner EDW Meta Data menu options.

Index indexname missing from tablename

The table name from the error message should have an index that does not exist. The name of the index is listed in the error message. Recreate the index, or remove the entry from the MGBINDX table.

Additional Index indexname found on tablename

The Banner EDW table listed in the error message had an index that is not expected. The index name is listed in the error message. If this is a valid index that your institution has added to the table, add the index to MGBINDX table.

Cleansing

Warning: Duplicate rule for Element [xxx_xxx] with Value: xx

If the data element requires a prefix element (CHART_OF_ACCOUNTS, for example) to determine its uniqueness, then verify that a parameter exists for the Internal Group in the Administrative UI. See "Set up Parameters (Banner ODS and Banner EDW)" on page 4-58.

> *Example*
>
> "INTERNAL_ACCOUNT_TYPE indicates a Value: 95 for Source: ODU - duplicate count: 2"
>
> This message indicates that based upon the values in Banner and Banner ODS, there are multiple descriptions for the code of 95. This is a data element related to the chart of accounts qualifiers which needs to be specified in the cleansing rules. See "Update a Description for Code Values" on page 2-81.

Data Element [xxx_xxx] needs a rule defined for value [xx]

There is a value in the data that does not have a cleansing value translation defined for it. You need to update the cleansing rules to reflect the identified values listed in the control reports. See "Update a Description for Code Values" on page 2-81 to update the ranges on the existing cleansing rule definitions to include that value, or set up a new value to code and a translation to handle it.

Example

"Data Element STATE_PROVINCE needs a rule defined for value…."

Materialized View Control Report Messages

Following are explanations of messages that you might see in a control report related to creating or maintaining a materialized view.

Warning: Creating materialized view: <viewName> with REFRESH WITH ROWID option due to lack of primary key constraint.

Oracle is unable to determine a primary key when joining the data used in the view. The system must use the ROWID construct to uniquely identify rows of data.

Note: Reporting View: <viewName> contains sub-query so temporary view used

Oracle doesn't support creating a materialized view from a view that contains a nested sub-query. The system is creating a temporary view on top of the reporting view. It will then materialize the temporary view

Note: Reporting View: <viewName> contains too many key columns - USING NO INDEX mode enabled.

Oracle uses the columns from the view that it determines to be the "primary key" columns when building its internal index for data access. The number of natural key columns in some reporting views exceed the Oracle limit. When this happens, the system uses the NO INDEX syntax.

Freeze Process

Process failed, no mgbfrez records for this list

The selected freeze table list does not have entries in MGBFREZ table. Click the **Freeze Data Maintenance** menu to review the tables included in the freeze list.

Multiple owners for inputted table/view to freeze. Please precede table name with owner.

The original table or view name exists in more than one schema. Verify which table the data should be selected from, and precede the table name with the owner name.

Source table not found

The original table was not found in the database.

Warning, no data found to Freeze

There are no rows in the original table, or the where condition caused no rows to be selected.

****Warning--Replace parameter is N and EVENT exists!!!- did not replace data*

Data has been previously frozen to the new table with the same event code. If the data should truly be replaced, submit the process with the **Replace parameter** checkbox checked. If the existing data should remain intact, use a different event name to freeze additional data into the new table.

Publish Meta Data (PUBLISH_META_DATA)

Configuration error: No script found for COPY_SCRIPT parameter

The location of the ftp script used to transfer the html files was not found in the MTVPARM table. Click the menu options of **Options** and **Set Up Parameters**, with the internal group = METADATA and the internal code = PUBLISH, to store the copy script.

P_MakeAllTarget - E_NoTablesFound

There were entries found in metadata tables

P_MakeAllTarget - E_NoMetafileLoc

The parameter record in MTVPARM does not exist. To create this records, click the menu options of **Options** and **Set Up Parameters** with the internal group = METADATA and the internal code2 = PUBLISH_LOCATION.

P_MakeAllTarget - E_NoUTLfileLoc

The file location supplied in the parameter is not valid. Click the menu options **Options** and **Set Up Parameters** with the internal group = METADATA and the internal code2 = PUBLISH_LOCATION to verify the correct location for the creation of the meta data files.

Reconcile (RECONCILE_JOB, RECONCILE_SINGLE_JOB)

If there are zero discrepancies, the number of rows in the source view match the number of rows extracted to Banner ODS table. Run a refresh (or load) for the mapping that has the discrepancies, then rerun the reconcile job.

mapName has ‘n’ discrepancies

There are 'n' differences between Banner and Banner ODS. (The message below provides additional details.)

Banner ODS has ‘n’ rows while the source has 'n’ rows. Key values are:

‘n’ rows while Banner has 'n’ rows. Key values are:

This indicates the key values for the rows in either Banner ODS or Banner that do not match to the other system. Use these key values to further diagnose the discrepancy.

Note

> If you run this reconcile process after the refresh process is run, records that have been updated (with changes noted in the change tables) may have caused the discrepancies - you can use the key values to confirm this. ■

Mapping processes (DELETE_mapping, UPDATE_mapping, LOAD_mapping, REFRESH_mapping)

OWB Runtime not running - waited for ‘n’ minutes...

ETL Mapping Package record not found for mapping: <map name>

Run Banner ODS Utilities - ‘Checks and Balances’ job to ensure that all parameter records exist and mapping packages are valid.

Mapping not found - Please check the mapping name and location.

Run Banner ODS Utilities - ‘Checks and Balances’ job to ensure that all parameter records exist and mapping packages are valid.

No ETL CONTROL GROUP or ETL MAP PACKAGES found for this job.

Check that records exist in MTVPARM table
`where mtvparm_internal_code_group = 'ETL MAP PACKAGE'.`

No ETL SLOT PACKAGE entry found for this table: <table name>

Check that records exist in MTVPARM table
where `mtvparm_internal_code_group = 'ETL SLOT PACKAGE'.`

Oracle Warehouse Builder Runtime Audit Browser Integration

Oracle Warehouse Builder (OWB) provides a utility called the Runtime Audit Browser (RAB) that displays status information for mappings that have been run. You can use RAB to view in depth statistics and job analysis. (For more information on setting up RAB, refer to the OWB Installation documentation).

Integration Setup

The Administrative UI can be configured to automatically link to the RAB for mappings that have been run. All you'll need to do is click a hyperlink from the control report to view RAB mapping information. A new browser window opens displaying the RAB information for that mapping. Follow the steps below to set up a parameter RAB_URL:

1. Click **Options** from the Administrative UI menu. The Options page opens.

2. Click **Set Up Parameters**. The Set Up a Parameter page opens.

3. Select Internal Group *METADATA* and Internal Code *RAB_URL* from the drop-down lists on the Set Up a Parameter page.

Note

> (If Internal Code *RAB_URL* does not appear in the drop-down list, then click **Create** to create the parameter. See "Set up Parameters (Banner ODS and Banner EDW)" on page 4-58 for instruction on how to create this optional parameter.) ■

4. Click **Search**. The Select an Existing Parameter page opens.

   The External Code on the Select an Existing Parameter page can be any value. (It is required, but ignored. You can enter a hyphen, for example.) The key is the **Description** column. It must be the URL for the RAB that you have installed and set up. It will be similar to the URL below:

   ```
   http://<machine_name>/owbb/
   RABMapExecution.uix?event=navigate&p_type=PLSQLMap&repos=RUNREP
   ```

   To access your URL, continue to the next step.

5. Open the RAB in another browser window. Copy your URL from the address bar in that window, and paste it into the **Description** column on the Select an Existing Parameter page.

Note

> The particular RAB address ("RABMapExecution.uix") and the associated parameters need to match the above address, with the exception of the "repos" parameter, which should reflect the repository owner in your system (if it isn't the default RUNREP schema/user). ■

RAB Authentication

The integration is not complete in the sense of typical web-based "single-signon." You must first sign into the RAB in that separate browser window before you can browse any of the mapping execution information. Once signed in, your RAB credentials are stored locally (in a cookie) in your browser so you can close the RAB window (after logging in).

Note

> Those cookie credentials are persistent, so future attempts to view RAB reports will succeed until you Log Out of your RAB session explicitly (via the Log Out link in the RAB window).

Subsequent links from the control report should take you directly to the mapping information for that report. Click the link after the OWB Audit Execution ID on the control report. The Runtime Repostitory page opens.

Set up E-mail Notification (Banner ODS and Banner EDW)

You can configure the Administrative UI to send an e-mail when a process (job) is completed. To do this, set up the following system parameters (MTVPARM records).

These parameters are not delivered. (You must create them. See the "Set up Parameters (Banner ODS and Banner EDW)" on page 4-58.) E-mails are only sent if all parameters (except the Administration URL) are set up. No e-mail notification takes place until you set these parameters.

Column	Description

`EMAIL_ADMIN_URL`	The complete URL to connect to the Administration system. If this parameter is defined, the URL is included in the e-mail message which makes it easier for the recipient to log into the system. The message contents are:
	Subject: <Job Name> Job Completion <with Errors> (where the Job Name is the job that ran, and “with Errors” is appended only for jobs that had errors.)
	Message: This job has completed. Check the Administrative UI for more details.
	`Job Name: <Job Name>`
	`Job User: <Admin Username of account that ran the job>`
	`Job Number: <job number>`
	`<For jobs that run mappings:>`
	`Job Execution Time:`
	`    Start of Mapping at (start time)`
	`    Process completed at (end time)`
	`<If any errors occurred during the job, they are listed next as follows:>`
	`Error Details:`
	`    Error ….`

`EMAIL_FROM_ADDRESS`	E-mail address in the **From** section of the e-mail. This is typically a server address. Required.
`EMAIL_LIST`	E-mail address to receive a job notification message for all Administration jobs that complete. Create one parameter for each recipient address. By default, you only receive e-mail notification for jobs submitted by that account. If the Administrative UI user name that ran the job matches (not case sensitive) the name in the **Description** field for this parameter. Or, you can set up an email address to receive notifications for all jobs that are run by setting the **INTERNAL_CODE_2** field to *GET_ALL_JOBS*. Required
`EMAIL_SERVER`	The machine name of your SMTP server machine. Required

Freeze Data Maintenance (Banner ODS and Banner EDW)

Freezing data enables you to take snapshots of related data at any point in time and keep a static copy of that data. You may want to run data comparison reports at the same point-in-time (example: each month, semester, or year) To do this you will need to ‘freeze’ the data at each point-in-time. As you save these data slices over time, you will create a history (freeze) of the data on which to report. You can also associate that point-in-time with an event name, for example, YearEnd, MonthEnd, or SummerSession.

Banner ODS freezes data from a single table/view or from multiple tables/views. When the freeze data has been defined, the freeze process must be scheduled to run (refer to “Freeze a Single Banner ODS Table/View” on page 4-140 and “Freeze Multiple Banner ODS Tables/Views at the Same Time” on page 4-141).

Banner EDW freezes data for all appropriate stars within a Banner EDW business concept. When the freeze data has been defined, the freeze process must be scheduled to run (refer to ‘Freeze a Banner EDW Business Concept’).

Use the Freeze Data Maintenance Option to:

- Set up Freeze Data list for Banner ODS tables/views
- Add additional Banner ODS tables to existing freeze lists
- Review events in existing Banner ODS freeze tables

Set up Banner ODS Freeze Data Lists

A Freeze List is what Banner ODS calls for one or more tables/views that have related data to be frozen at the same time. The freeze process selects data from the source table/view, creates a table with the ‘history’ name supplied, and copies (freeze) the selected source data into the history table. By default, all the columns from the source table are copied to the freeze table. Click **Select Columns** to specify if only specific columns are required for the freeze.

Example

> During a student registration cycle it may be important to capture student courses weekly. First, you would create a freeze list called STUDENT_COURSE_REGDATA. The source data would then be selected from STUDENT_COURSE.

The data from the source is stored in a freeze table which could be named STUDENT_COURSE_STATIC, for example. The new table is created the first time the freeze is run. Any successive freezes for this freeze list reuses the static table.

Note

It is recommended that your institution have a naming convention in place for freeze lists and freeze tables ■

There is an optional WHERE condition that allows you to qualify the data to be frozen from each source table. The condition is ACADEMIC_PERIOD = ‘200510’).

Note

Do *not* include the actual word WHERE in the condition; it is assumed. ■

1. Click **Options**.

2. Click **Freeze Data Maintenance**. The Set Up Freeze Data Lists page opens.

3. Click **Create** from the Set Up Freeze Data Lists page. The Create a Freeze Data Table page opens.

   The links on this page are described below:

Link	Description
View Current Lists	Opens a window of the current freeze lists.
Copy Table Name	Copies the source name to the **Freeze Table Name** field. **Note:** The freeze table name must be different than the source name. The data is frozen with this table.
Select Columns	Opens a window of existing freeze columns. Choose the column(s) to freeze or not freeze, then click the corresponding arrows to move them to the appropriate box. • Click a single arrow to move one column. • Click a multi-arrow to move all columns. • Hold down the Ctrl key while selecting to move a few columns. The number of columns selected out of the total number of columns appears on the page and in the window. For example, (178/181) indicates that 178 columns out of 181 will be frozen.

4. Enter the new list name, source name and freeze table name.

5. *(Optional)* Enter a valid PL/SQL WHERE condition. Use fields from the table or view being frozen and exclude the word “where”, which is added by the system.

Example

```
ACADEMIC_PERIOD= ‘200510’ and COURSE_LEVEL = ‘01’
```

6. Click **Save**.

Add a Table/View to a Banner ODS Freeze Data List

Maintaining freeze lists may require that additional tables be included in specific freeze lists, that a freeze list be deleted, that a freeze list be renamed or duplicated. It is also useful to review which events exist in which freeze tables.

In the example above, it is decided to capture data from STUDENT_COHORT_SLOT and so an additional table should be added to the STUDENT_COURSE_REGDATA freeze list.

- Click the freeze list called **STUDENT_COURSE_REGDATA** from the drop down list.
- Click **Add another Table**.
- Click **STUDENT_COHORT_SLOT** as the source table.
- For this example, the freeze table will be STUDENT_COHORT_SLOT_STATIC

  There is an optional WHERE condition that will allow you to qualify the data to be frozen from each source table (ACADEMIC_PERIOD = ‘200510’).   NOTE: Do *not* include the actual word WHERE in the condition. It is assumed.

1. Click **Options**.
2. Click **Freeze Data Maintenance**. The Set Up Freeze Data Lists page opens.
3. Choose the freeze list you want to modify from the drop-down list on the Set Up Freeze Data Lists page.
4. Click **Search**. The Select a Freeze Data Table page opens displaying the freeze tables associated with the displayed freeze list.
5. Click the link in the **Source Name** column for the tables/views you want to add. The Update an Existing Freeze Data Table page opens.
6. Click **Add Another Table**. The Create a Freeze Data Table page opens.

   The links on this page are described below:

Link	Description
Select Another Lists	Returns to the Select a Freeze Data Table page.

Link	Description
Add Another Table	Opens the Create a Freeze Data Table page where you can enter freeze tables to add.
Select Columns	Opens a window of existing freeze columns. Click the column(s) to freeze or not freeze, then click the corresponding arrows to move them to the appropriate box. • Click a single arrow to move one column. • Click a multi-arrow to move all columns. • Hold down the Ctrl key while selecting/ deselecting to move a multiple columns. The number of columns selected out of the total number of columns appears on the page and in the window. For example, (178/181) indicates that 178 columns out of 181 will be frozen.

7. Enter the new list name, source name and freeze table name.

8. (Optional) Enter a valid PL/SQL **Where Condition**. Use fields from the table or view being frozen and exclude the word “where”, which is added by the system.

   For example: *academic_period = ‘200510’*.

9. Click **Save**.

10. Click **Add Another Table** to add another table to your list.

Delete, Rename or Duplicate Banner ODS Freeze Data

Follow the steps below to delete, rename, or duplicate freeze data list.

1. Click **Options** from the Administrative menu.

2. Click **Freeze Data Maintenance**. The Set Up Freeze Data Lists page opens.

3. Choose the Freeze List you want to modify from the drop-down list on the Set Up Freeze Data Lists page.

4. Click **Search**. The Select a Freeze Data Table page opens. The list of tables currently included in the list displays.

   Use the links on this page to delete, rename or duplicate a freeze list. Each link is described below:

Link	Description
Delete Freeze List	Confirms that you want to delete the displayed freeze list. Tables of frozen data will not be deleted.
Rename Freeze List	Displays the Rename a Freeze List window. Enter the new freeze list name, then click **Rename**. Tables of frozen data will not be renamed.
Duplicate Freeze List	Displays the Duplicate a Freeze List window. Enter the new name, then click **Duplicate**. None of the history tables are duplicated.

5. (Optional) Check the **Show Event Names** checkbox to indicate whether to display the event within each table. An extra column of names displays.

Note

You choose how these events are handled when scheduling a job by choosing to either to insert, delete or replace the events from the **Event Handling** field on the Schedule a Process page. ■

Freeze a Single Banner ODS Table/View

You can freeze a single table using the Schedule a Process>Freeze a Single Banner ODS Table option. Follow the steps below:

1. From the Administrative menu, click **Options**.

2. Click **Schedule a Process**. The Select a Process page opens.

3. Click **Freeze A Single Banner ODS Table/View** from the Select a Process page.

4. Enter the required process parameters.

   4.1. Type the name of a table/view into the **Enter Table to Freeze** field.

   4.2. Type the new (history) table name into the **Enter Table Name to Freeze to** field. (Follow your history table naming conventions.)

5. Enter the required scheduling parameters.

   5.1. Enter a **Run Date** (format dd-mon-yyyy) and **Runtime** (format hh24:mi:ss).

   5.2. If you want to run the process on a recurring basis, enter an **Interval**. For example, to run a process every day at the same time enter *SYSDATE+1* in the **Interval** scheduling parameter.

   See "Update or Freeze Recurring Banner ODS Data" on page 4-145 for more details on setting the **Interval**.

6. Click **Save** to save the information about this freeze job. The job is entered into the job queue to run at the specified day and time.

Freeze Multiple Banner ODS Tables/Views at the Same Time

If the freeze is going to occur repeatedly, it may be useful to create a Freeze List. The Freeze List is a name/label/title for one or more tables/views with data to be frozen at the same time. See "Freeze Data Maintenance (Banner ODS and Banner EDW)" on page 4-135 for instructions on how to define a list of freeze tables.

Follow the steps below to freeze multiple tables/views:

1. From the Administrative menu, click **Options**.

2. Click **Schedule a Process**. The Select a Subprocess page opens.

3. Click **Freeze Multiple Banner ODS Tables/Views** from the Select a Process page.

   All freeze data lists defined within Freeze Data Maintenance display.

4. Click the freeze data list. The Schedule a Process page opens.

5. From the **Event Handling** drop-down list, indicate whether you want to replace, insert (add), or delete existing events from the tables in the freeze data list.

6. Choose an event to capture. The system tags the information extracted during this process with the event code you choose.

Note

You have to choose an event name when you submit the freeze job to run (refer to the "System Parameters" section). Once that freeze job is run, the data exists in the freeze tables with an 'event' name attached. There could be multiple event names in a single freeze table. ■

7. Enter the **Run Date** (format dd-mon-yyyy) and **Runtime** (format hh24:mi:ss).

If you want to run the process on a recurring basis, enter an Interval. For example, to run a process every day at the same time enter *SYSDATE+1* in the Interval scheduling parameter. See "Update or Freeze Recurring Banner ODS Data" on page 4-145for more details on setting the Interval.

8. Click **Save** to save the information about this freeze job. The job is entered into the job queue to run at the specified day and time.

Create a dynamic Freeze List parameter

You can create dynamic parameters and dropdown lists of valid values for selection parameters associated with a Freeze List. When you run a Freeze List, you can then use the parameter to reduce the amount of data that is frozen. The value selected and the associated column name are then appended to the "where" clause for each table in the freeze list.

Setting up this functionality requires the following conditions:

- Only tables or views owned by the ODSMGR schema can be used as source for a Freeze List
- A column that will be used as the "target" for a parameter value must be present in all source objects in the Freeze List

  You can create a new view to meet this condition (For example, you could create a new view joining `STUDENT` to `PERSON_DETAIL`, adding `ACADEMIC_PERIOD` to the new view (`STUDENT_PERSON`) so only the relevant "person" rows are frozen.)

Use the following steps to set up a parameter for a Freeze List.

1. Create a Freeze List.

2. Select **Options>Set Up Parameters**.

3. From the **Internal Groups** dropdown list, select PARAMETER.

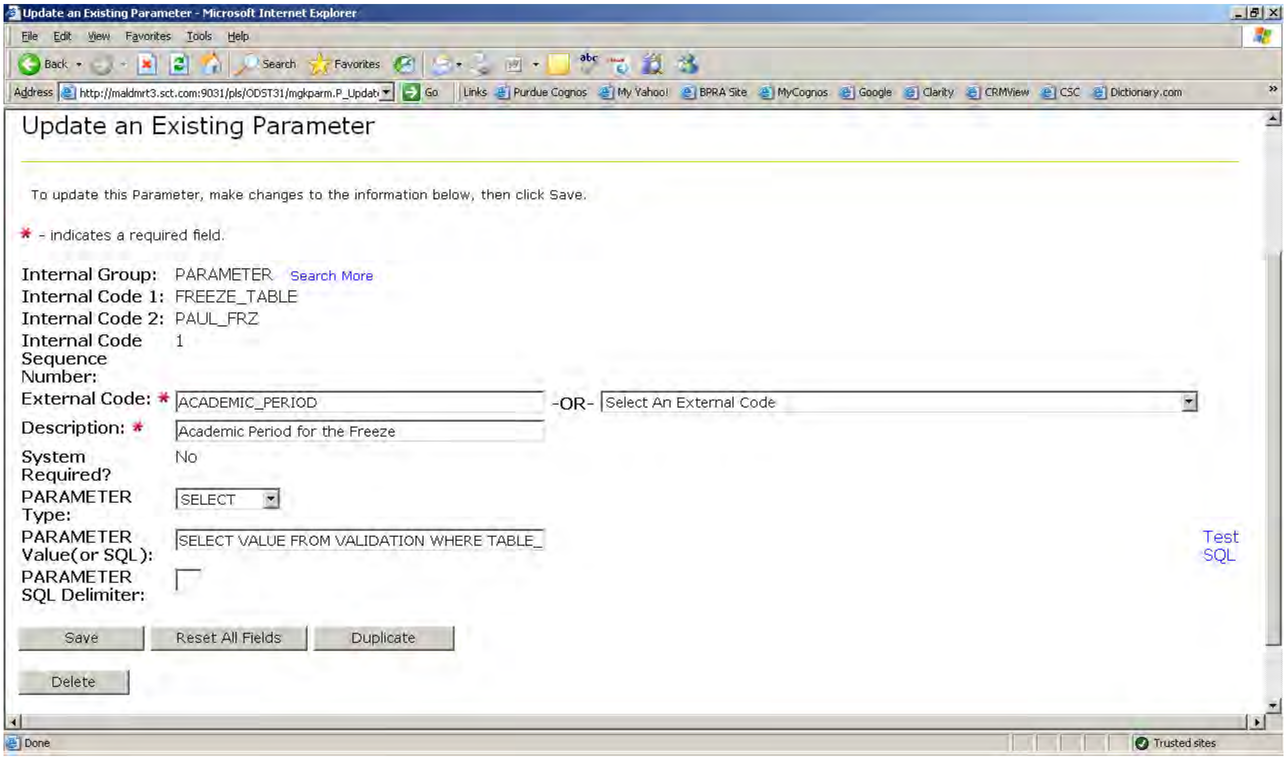
4. Click **Create** to create a new entry for the Freeze List.

5. Enter values to create a new Parameter record. Use the descriptions in the following table as a guide to define the new Parameter record.

Field	Value
Internal Group	PARAMETER
Internal Code 1	FREEZE_TABLE

Field	Value
Internal Code 2	<Freeze List Name> Enter the name of the Freeze List you created in the step 1 of this procedure.
Internal Code Sequence Number	Null or the sequence in which you want the parameter to display on the Schedule a Process page
External Code	<Source COLUMN_NAME to be tested>
Description	<Description of Process> Enter the description of the process that will display on the Schedule a Process page.
System Required	No

The following image shows the page for the “Parameter” parameter after it was created, and then updated to choose Parameter Type and input the Parameter Value.


Update an Existing Parameter - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://maldmrt3.sct.com:9031/pls/ODST31/mgkparm.P_Updat Go
Links Purdue Cognos My Yahoo! BPRA Site MyCognos Google Clarity CRMView CSC Dictionary.com
Done
Trusted sites

Update an Existing Parameter

To update this Parameter, make changes to the information below, then click Save.

* - indicates a required field.

Internal Group:	PARAMETER Search More
Internal Code 1:	FREEZE_TABLE
Internal Code 2:	PAUL_FRZ
Internal Code Sequence Number:	1
External Code: *	ACADEMIC_PERIOD -OR- Select An External Code
Description: *	Academic Period for the Freeze
System Required?	No
PARAMETER Type:	SELECT
PARAMETER Value(or SQL):	SELECT VALUE FROM VALIDATION WHERE TABLE_ Test SQL
PARAMETER SQL Delimiter:	□

Save Reset All Fields Duplicate

Delete

For SELECT parameter types the Parameter Value is the SELECT statement used to populate the dropdown list. In this example the statement was:

```
SELECT VALUE FROM VALIDATION WHERE TABLE_NAME = 'STVTERM' ORDER BY
VALUE DESC
```

You can test the correctness of the SELECT statement by clicking the **Test SQL** link in the lower right corner of the page, resulting (when the arrow is clicked) in the image to the right.

The value stored as the External Code is added to the "Where" clause for each source in the Freeze list and tested for equality to the parameter value chosen from the list as the job is submitted.

Example - Freeze List parameter

The easiest way to understand how to create a new runtime parameter is to review the existing ones. In the previous table, there are two records whose Internal Code 1 = FREEZE_TABLE and the Internal Code 2 field is blank. Each of these records defines a runtime parameter that appears on the Schedule a Process page when Freeze Multiple Banner ODS Tables/Views is selected. The Internal Code 1 field of FREEZE_TABLE on the Parameter record here matches to the External Code of FREEZE_TABLE on the INSTALLED PROCESS parameter.

If you add a runtime parameter to a freeze data list called TEST1, the following steps show the field values needed to create this new parameter.

1. Enter **Internal Group** = *PARAMETER*.

2. Enter **Internal Code 1** = *FREEZE_TABLE*. The parent process for the TEST1 freeze data list.

3. Enter **Internal Code 2** = *TEST1*. The actual name of the freeze data list to associate the parameter.

4. Enter **Internal Code Sequence Number** = *2*. The order that parameters are listed at runtime. You can add up to two parameters to a freeze data list.

5. Enter **External Code** = *ACADEMIC_PERIOD*. The actual field value that you want the user to supply at runtime.

6. Enter **Description** = *Enter Term Code*. The prompt that a user needs to supply at runtime.

7. Choose **PARAMETER Type** = *SELECT.* Identifies how the user enters the runtime parameter. The field accepts four values:
   - SELECT = User must supply a valid PL/SQL statement.
   - DATE = User must supply a valid date.
   - EDIT = User can supply a text string.
   - CHECKBOX = User must check or uncheck an option.

8. Enter **PARAMETER SQL**. This field is only required when the **PARAMETER Type** is *SELECT*. Enter a valid PL/SQL statement, which is used to populate the valid field values to display in the drop-down list of the runtime prompt.

9. Enter **PARAMETER SQL Delimiter**. This field is only required when the **PARAMETER Type** is *SELECT* and you use a delimiter in the **PARAMETER SQL** field. Specify the delimiter used in the **PARAMETER SQL** field.

Update or Freeze Recurring Banner ODS Data

You'll need to refresh the data in your Banner ODS on a regular basis to keep it synchronized with data in your administrative system. You may also want to freeze portions of Banner ODS data on a regular basis so that your users can create data comparison reports.

To automate the refresh or freeze processes, use the **Schedule a Process** option to define processes that run on a recurring basis. Specify that a job run on a recurring basis by entering a valid PL/SQL value in the **Interval** field. This field accepts a data expression value, which defines the length of time between processing runs. The key to setting the interval correctly is determining whether you need to run a job so that:

- Each execution of the job follows the previous run by a specific time interval.

  or

- The job executes on specific dates and times.

The first thing you need to do is determine when and/or how often your institution needs to update Banner ODS data.

Update Banner ODS Daily

It is recommended that Banner ODS is updated daily. Use the **Schedule a Process** option to define processes that run on a recurring basis. Specify that a job run on a recurring basis by entering a valid PL/SQL value in the **Interval** field. This field accepts a data expression value, which defines the length of time between processing runs. The key to setting the interval correctly is determining whether you need to run a job so that:

- Each execution of the job follows the previous run by a specific time interval.

  or

- The job executes on specific dates and times.

The first thing you need to do is determine when and/or how often your institution needs to update Banner ODS data.

In this case, the interval value is a date arithmetic expression like *SYSDATE+N*, where *N* represents the time interval expressed in days. So, an interval of SYSDATE+1 runs the job on a daily basis.

Job intervals set using date expressions do not guarantee that the next execution happens at a specific day or time, only that the spacing between executions is at least what was specified.

Example

> If a job is first executed at 12:00 p.m. with an interval of SYSDATE + 1, it will be scheduled to execute the next day at 12:00 p.m. However, the job is executed manually at 4:00 p.m. using DBMS_JOB.RUN, then it is rescheduled for execution at 4:00 p.m. the next day. Another example is when the database is down or the job queue is so busy that the job cannot be executed exactly at the time scheduled. In this case, the job runs as soon as it can, but the execution time then moves away from the original submission time due to the later execution.

Update Banner ODS on Specific Dates and Times

You can set the Interval to execute jobs on a specific date and time. This type of interval involves more complex interval date expressions. Specifying intervals like these can get tricky, so be sure that your date arithmetic expression is correct. The following table provides samples of both simple and more complex types of job intervals.

Note

Refer to your Oracle documentation for more information on setting job intervals. ■

Run job	Interval Value
Daily	*SYSDATE+1*
Hourly	*SYSDATE + 1/24*
Weekly (every 7 days)	*SYSDATE + 7*
Every day at 12:00 midnight	*TRUNC(SYSDATE + 1)*
Every day at 8:00 a.m.	*TRUNC(SYSDATE + 1) + 8/24*
Every Tuesday at 12:00 noon	*NEXT_DAY(TRUNC(SYSDATE), TUESDAY) + 12/24*
First day of the month at midnight	*TRUNC(LAST_DAY(SYSDATE) + 1)*

Run job	Interval Value
Last day of the quarter at 11:00 P.M.	*TRUNC(ADD_MONTHS(SYSDATE + 2/24, 3), 'Q') - 1/24*
Every Monday, Wednesday, and Friday at 9:00 a.m.	*TRUNC(LEAST(NEXT_DAY(SYSDATE, MONDAY), NEXT_DAY(SYSDATE, WEDNESDAY), NEXT_DAY(SYSDATE, FRIDAY))) + 9/24*

Freeze or Snapshot Banner EDW Business Concept

All the stars within a Banner EDW Business Concept can be frozen.

1. From the Administrative menu, click **Options**.

2. Click **Schedule a Process**.

   The Select a Process page opens.

3. Click **Schedule Banner EDW Operational Mappings**.

4. Enter the required process parameters.

   4.1. Select the time period for which you want to freeze the data (example: Academic Period).

   4.2. Select the Event that corresponds to the frozen data.

   For additional Event choices, go to Options> Set up Parameters > EVENT-EDW and set up additional Event Categories, Event Types and Event Descriptions.

   4.3. Indicate whether to replace the event, if it already exists.

5. Enter the required scheduling parameters.

   5.1. Enter a Run Date (format dd-mon-yyyy) and Runtime (format hh24:mi:ss).

   5.2. If you want to run the process on a recurring basis, enter an Interval.

   For example, to run a process every day at the same time enter *SYSDATE+1* in the **Interval** scheduling parameter.

6. Click **Save** to save the information about this freeze job.

   The job is entered into the job queue to run at the specified day and time.

Meta Data (Banner ODS and Banner EDW)

Meta data is “data about data” or information, or characteristics, about data entities such as a column name, description, format, length, origin and destination.

Meta data in Banner ODS and Banner EDW tell what data columns are in Banner ODS and Banner EDW, a definition of their business use, the type of data (number, character, date, etc.), how long they are, where they come from (in the source system) and their destination (in the target system.)

The Administrative UI meta data pages include reports that show the relationship between the data stored in Banner ODS and Banner EDW (target) and the source from which it is extracted.

Note

The meta data includes Banner ODS reporting views and source composite views, both with the original source tables and source column names. Banner ODS recreated Object:Access views are *not* delivered in the meta data. They are additional reporting views to be used for clients migrating from Datamart 1.0, or clients who used the source Object:Access views for custom reporting. Newly developed Banner ODS reporting should *not* use the Object:Access views.

Banner EDW meta data includes fact tables, dimension tables, and stars.

The following navigation links and buttons display throughout the Administrative UI meta data pages:

This Link/Button ...	Does this ...
<- ->	Moves through the subject areas in alphabetical order. Moves alphabetically through the views within a subject area. Moves through the columns within a view in column ID order.
Select	***From the Subject Area:*** Opens the Select A Subject Area window. Click the **Target** or **Source** radio group, and click **Reporting View** or **Composite View** radio group to indicate the report type with which you want to work. Choose the new subject area with which you want to work. The window closes automatically. ***From the Reporting or Composite View:*** Opens the Select A Target window. Choose the reporting or composite view with which you wish to work. The window closes automatically. ***From the EDW star Target report List*** Opens the Select a Report Type and a Star window. Choose the target or source view and select the star. The window closes automatically. ***From the Reporting or Composite View Column:*** Opens the Select a Target Column or Select a Source Column window. To have the columns listed alphabetically, click the **Sort By: Column Name** radio group. To have the columns listed in column order, click the **Sort By: Column Order** radio group. Click the column with which you wish to work. The window closes automatically.
Add Target	Adds a target view or table. The Target window opens.
Add Source	Adds a source view. The Source window opens.

This Link/Button ...	Does this ...
Add Target Column	Add a target column. The Add a New Column window opens
Add Source Column	Add a source column. The Add a New Column window opens
List Composite Views	Displays the composite views for the selected subject area.
List Banner ODS Reporting Views	Displays the reporting views for the selected subject area.
Preview	Save, then click **Preview** to review your changes. Do not click this link then click the back button. Or, click **Preview** to review the list of: • all reporting or composite views within a subject area • a list of all source, or target columns within a view • a list of fact/dimension tables within a star
Properties	Works in conjunction with the **Columns** link to toggle between the Edit Target Columns and the Edit Target Properties pages.
Columns	Works in conjunction with the **Properties** link to toggle between the Edit Target Properties and the Edit Target Columns pages.
Preferences	Opens the Institutional Preferences window.
Publish	Publishes the meta data.
CSV Export	Exports meta data into a .csv file that you can open in Microsoft Excel.
Import	Enables you to choose a view to import into the meta data. The view must exist in the ODSMGR schema. All the columns in the view are created as LOCAL meta data. Click the button to display a list of views that do not exist in the meta data for that subject area.
Show Baseline/Hide	Toggles between displaying baseline information versus local information.

This Link/Button ...	Does this ...
Save Changes	Records your changes.
Select Another Maintenance Function	Returns to the Maintain Banner ODS and Banner EDW Meta Data page.

Baseline and Local Meta Data

Baseline meta data is the meta data delivered with your solution. When you change the baseline meta data, a local copy is created and the edited version becomes your local meta data. Local meta data appears on the Administrative UI page in the color specified in your Institutional Preferences. The **Local Record** field on the Edit Target (or Source) Columns pages indicates whether the displayed meta data is the baseline or local version.

If both local and baseline meta data exist for the column meta data, only the local meta data displays and can be edited. Only local meta data can be changed or deleted.

Create Meta Data

When Banner ODS and Banner EDW are installed, the baseline meta data is installed as well. The sections “Set Up Meta Data Publish Preferences” and “Meta Data Parameter Set Up for Publishing Reports,” describe procedures that were completed during installation. They are included here for completeness, but you do not have to perform them to create meta data. The maintaining target or source meta data sections describe how to update the meta data repository with your own meta data.

Set up Institutional Meta Data Publish Preferences

The Meta Data Publish Preferences option controls which pieces of meta data can be previewed on the screen and saved (published) in a report. Meta data is considered ‘published’ after you save the selected source or target information as an HTML file using the Administrative UI. Before you publish meta data, follow the steps below to set the preferences.

1. Click **Preferences & Security** from the Administrative menu. The Preferences & Security menu opens.

2. Click **Institutional Preferences**. The Set Up Banner ODS or Banner EDW Publishing Options menu opens.

3. Click **Banner ODS or Banner EDW Publishing Preferences.** The Set Up Meta Data Publish Preference page opens.

4. In the Banner ODS or Banner EDW Meta Data Target Report Preferences area on the Set Up Meta Data Publish Preference page, check the checkbox to indicate the meta data you want to display in your meta data target or source reports. Your solution is delivered with the default check boxes selected.

5. Choose the color in which you want your report rows and local meta data information to appear.

Note

Colors appear institution-wide. They are not personal colors. ■

6. Indicate whether reports should appear in column or name order.

7. Click **Save** to keep your changes.

Meta Data Parameter Set up for Publishing Reports

The meta data reports are created as static HTML pages from the Administrative UI or from the command line. This process is called ‘publishing’. (See “Publish Meta Data from the Administrative UI” on page 4-168 for additional information on publishing meta data reports from the command line.)

There are system parameters that must be configured when Banner ODS and Banner EDW are installed. The PUBLISH_LOCATION parameter provides the directory location on the database server where the HTML pages are written when using the Publish buttons from the Administrative UI, or by running the PUBLISH.BAT script in batch mode.

There are two supported techniques for specifying the location on the file system where the HTML pages are created:

- Use an Oracle DIRECTORY object

  This is the preferred method as it does not require you to restart the database for changes to take effect, and also offers greater control over security. DIRECTORY objects are like any other objects in the Oracle database and offer the same levels of security control (grants by schema/user) while the UTL_FILE_DIR parameter setting is a global resource that does not offer tighter security control.

- Use the Oracle initialization UTL_FILE_DIR parameter

  This technique has been replaced by the DIRECTORY usage but will be supported for backward compatibility.

When using an Oracle DIRECTORY, use the following syntax to create the directory object in the IA_ADMIN schema:

```
CREATE DIRECTORY <DIRNAM> FOR '<PATH>';
```

where <DIRNAME> is a string, like *METADATA_DIR* and <PATH> is the actual path to the folder/directory location where the files are created.

The DIRECTORY object should be created and owned by the IA_ADMIN schema. The value of the PUBLISH_LOCATION parameter would then be set to the DIRECTORY name (in the above example, the value: *METADATA_DIR*.

You need to specify the initialization parameter UTL_FILE_DIR within the init.ora file for Banner ODS and Banner EDW instance. This UTL_FILE_DIR parameter must contain the name of the directory where the Admin pl/sql package (MGKPUBL) generates the meta data files on the database server.

Once this directory is known and the UTL_FILE_DIR parameter is set, then configure the PUBLISH_LOCATION parameter through the Administrative UI. (Follow the directions in the section "Configure Publishing Parameters and Create Meta Data Web Directory".)

The VIEW_URL parameter provides the Web server location where the published files are hosted. It is recommended that you use the delivered /meta data folder to store the generated reports for viewing. This is a subdirectory beneath the "document root" for the Web server instance.

- Specify the VIEW_URL parameter as a relative path to the document root.
- If the Oracle http server (Web server) is on a different computer from Banner ODS and Banner EDW database server, then newly published reports must be copied from the PUBLISH_LOCATION to the /metadata subdirectory before they can be viewed from the Operational Data Store Meta Data Reports page.

The COPY_SCRIPT parameter allows you to specify a script to accomplish the moving HTML files from the application server to the web server.

The sample script delivered (ia_admin\dbscripts\utility_scripts\copyMetaData.sh) demonstrates how to do this using FTP, but the script can be replaced with any technique (such as SFTP, copying files directly using a mapped drive, even just copying them from one directory to another if the application server and web server are on the same machine, etc.). It is recommended that you examine and customize this script as needed to comply with your institutional security requirements and policies.

Configure Publishing Parameters and Create Meta Data Web Directory

1. Login to the Administrative UI.

Example

```
http://machine.sct.com:port/pls/ods/
twbkwbis.P_GenMenu?name=bmenu.P_MainMnu
```

2. Click **Options** from the Administrative UI menu.

3. Click **Set Up Parameters**.

4. From the **Show All Internal Groups** drop-down list, select *METADATA*.

5. Click **Search**.

6. Look for the PUBLISH_LOCATION, VIEW_URL, or COPY_SCRIPT parameter in the **Internal Code 2** column.

7. Click the corresponding link in the **Description** column.

8. Each link for the selected parameter appears in the **Description** field of the Update a Parameter page.

For parameter ...	Change this Directory or Server Location ...
`PUBLISH_ LOCATION`	`UTL_FILE_DIR` location or DIRECTORY name. Select this parameter to set up the location to which the meta data is published. If the Web server is running on a Banner ODS and Banner EDW machine, set up the UTL_FILE_DIR location (for output of generated pages) to be the same as the meta data subdirectory path under the Web server document root. PUBLISH_LOCATION would be set to the same thing. *Example:* D:\ORACLE\UTL_FILE PUBLISH_LOCATION may be case sensitive. The directory name on the Banner ODS database server should be in the same case as the UTL_FILE_DIR entry. If the case does not match, you may receive the error “Unknown Status: ERR_UTL_FILE” when attempting to Publish. The description should correspond to the `UTL_FILE_DIRECTORY` setting. *Example* The `initSID.ora` file may contain this line: `utl_file_dir = D:\ORACLE\UTL_FILE` The database init parameter file (`initSID.ora`) typically resides in the `Oracle Home\database` directory (Windows) or the `Oracle Home/dbs` directory (Unix).
`VIEW_URL`	The VIEW_URL parameter is set to the meta data subdirectory of the document root. This saves you from copying files each time they are published. *Example:* \metadata If the Oracle http server (Web server) is on a different computer from the ODS database server, then newly published reports must be copied from the PUBLISH_LOCATION to the /metadata subdirectory before they can be viewed from the Banner Operational Data Store Meta Data Reports page.
`COPY_SCRIPT`	Script used to move HTML pages from the database server to the application Web server.

Tip

On the Update a Parameter page, you can only change the **External Code** and **Description** fields. But, if you click **Duplicate** you can change any of the fields. For example, to update the Internal Code you could duplicate the current one and change the Internal Code. Then, go back and delete the original parameter (to clean up). Click the back button (twice), then click **Delete**. ■

9. Click **Save** from the Update a Parameter page to save the new settings.

10. Return to the Select a Parameter page to set up a different parameter.

Edit Target Meta Data Properties

Follow the steps below to change the properties of your target meta data.

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS Meta Data or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default.

To move through the subject areas in alphabetical order, click **<-** or **->**. ■

3. Click **Select** on the View Target Report List page to choose the target subject area you want to edit. The Select A Subject Area (Banner ODS) or Select a Star (Banner EDW) window opens.

4. Click either the **Reporting View(s)** or **Composite View(s)** radio group (Banner ODS) or Target or Source (Banner EDW). Then, click the **Target(s)** radio group.

5. Click the subject area you want to view. The window closes automatically.

Note

Check the **Show locally modified targets only** checkbox to display local target views only. ■

6. Click the reporting or composite view whose properties you want to change. The View Target Report page opens.

7. Click **Properties** located to the right of the reporting/composite view name. The Edit Target Properties page opens.

8. Make your changes.

9. Click **Save Changes** at the bottom of the page to keep your new information. The page refreshes automatically.

After the page refreshes, the **Local Record** field changes from *No* to *Yes* to indicate that this is now local meta data. The field names also display in the color that was set up in your Institutional Preferences page to indicate local meta data.

The **Show Baseline** and **Delete Local** links appear to the right of the **Local Record** field after you save.

Add Target Views and Target Columns

Follow the steps below to add target reporting or composite views and target columns to a subject area.

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS Meta Data or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default.

To move through the subject areas in alphabetical order, click **<-** or **->**. ■

3. Click **Select** on the View Target Report List page to choose the target subject area you want to edit. The Select A Subject Area (Banner ODS) or Select a Star (Banner EDW) window opens.

4. Click either the **Reporting View(s**) or **Composite View(s)** radio group (Banner ODS) or Target or Source (Banner EDW). Then, click the **Target(s)** radio group.

Note

Check the **Show locally modified targets only** checkbox to display local target views only. ■

5. Click **Add Target**. The Add Target window opens.

6. Enter the new target name.

7. Click **Add Target** to save the new view. The View Target Report page opens displaying the new target reporting or composite name.

8. Click **Add Target Column** to add columns to the view. The Add a New Column window opens.

9. Enter the new information, then click **Add Colum**n to save. The View Target Report page refreshes and displays the new target column information.

Edit Target Views and Target Columns

Follow the steps below to change the information for target and reporting or composite views.

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS Meta Data or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default.

To move through the subject areas in alphabetical order, click **<-** or **->**. ■

3. Click **Select** on the View Target Report List page to choose the target subject area you want to edit. The Select A Subject Area (Banner ODS) or Select a Star (Banner EDW) window opens.

4. Click either the **Reporting View(s**) or **Composite View(s)** radio group (Banner ODS) or Target or Source (Banner EDW). Then, click the **Target(s)** radio group.

5. Click the subject area you want to view. The window closes automatically.

Note

Check the **Show locally modified targets only** checkbox to display local target views only. ■

6. Click the reporting or composite view you want to change. The View Target Report page opens.

7. Click the Target Column you want to change. Banner EDW users can select the dimension or fact table you want to change.

   The Edit Target Columns page opens.

8. Enter your changes. Click **Save Changes** to keep your changes.

Synchronize Meta Data Comments with Reporting Views

Use this option to generate multiple comments on a reporting view. The meta data business definitions for the reporting view and the meta data business definitions for each of the columns is copied from the meta data into the database **Comments** field. Any existing comment will be overwritten.

This process (for a single or for multiple business definitions) can also be scheduled from Banner ODS Utilities menu. (See “Schedule a Process (Banner ODS and Banner EDW)” on page 4-65 for instructions on how to schedule a process.)

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS Meta Data.** The View Target Report List page opens.

3. Click **Select** to choose the subject area, or click <- or -> to move through the subject areas in alphabetical order.

Note

If you click **Select,** which opens the Select a Subject Area window, keep the default Reporting View and Target radio groups. The window closes automatically after you select a subject area. ■

4. Click the reporting view to which you want to add comments.

5. Click the **Sync Comments** link.

The business definitions are copied to the database comments.

Delete Local Target Properties

Follow the steps below to delete local target properties:

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS Meta Data or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default.

To move through the subject areas in alphabetical order, click **<-** or **->**. ■

3. Click **Select** on the View Target Report List page to choose the target subject area you want to edit. The Select A Subject Area (Banner ODS) or Select a Star (Banner EDW) window opens.

4. Click either the **Reporting View(s**) or **Composite View(s)** radio group (Banner ODS) or Target or Source (Banner EDW). Then, click the **Target(s)** radio group.

5. Click the subject area you want to view. The window closes automatically.

Note

Check the **Show locally modified targets only** checkbox to display local target views only. ■

6. Click the reporting or composite view whose target properties you want to delete. The View Target Report page opens.

7. Click **Delete**. A message window appears.

8. Click **OK** to delete the target, or **Cancel** to keep the target. If you delete the target, the View Target Report List page opens. If you keep the target, you remain on the View Target Report page.

Delete Local Target Columns

Follow the steps below to delete local target columns.

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS Meta Data or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default.

To move through the subject areas in alphabetical order, click **<-** or **->**. ■

3. Click **Select** on the View Target Report List page to choose the target subject area you want to edit. The Select A Subject Area (Banner ODS) or Select a Star (Banner EDW) window opens.

4. Click either the **Reporting View(s**) or **Composite View(s)** radio group (Banner ODS) or Target or Source (Banner EDW). Then, click the **Target(s)** radio group.

5. Click the subject area you want to view. The window closes automatically.

Note

Check the **Show locally modified targets only** checkbox to display local target views only. ■

6. Click the reporting or composite view whose columns you want to delete. The View Target Report page opens.

7. Click the target column you want to delete. The Edit Target Column page opens.

8. Click **Delete Local**. A message window appears.

9. Click **OK** to delete the target, or **Cancel** to keep the target. If you delete the target, you return to the View Target Report page. If you keep the target, you remain on the Edit Definitions page.

Edit Source Meta Data Properties

Follow the steps below to change the properties of your source meta data.

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS Meta Data or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default.

To move through the subject areas in alphabetical order, click **<-** or **->**. ■

3. Click **Select** on the View Target Report List page to choose the target subject area you want to edit. The Select A Subject Area (Banner ODS) or Select a Star (Banner EDW) window opens.

4. Click either the **Reporting View(s**) or **Composite View(s)** radio group (Banner ODS) or Target or Source (Banner EDW). Then, click the **Target(s)** radio group.

5. Click the subject area you want to view. The window closes automatically.

Note

Check the **Show locally modified sources only** checkbox to display local target views only. ■

6. Click the Source Name whose properties you want to change. The View Source Report page opens.

7. Click **Properties** located to the right of the source name. The Edit Source Properties page opens.

8. Make your changes.

9. Click **Save Changes** at the bottom of the page to keep your new information. The page refreshes automatically.

   After the page refreshes, the **Local Record** field changes from *NO* to *YES* to indicate that this is now local meta data. The field names also display in the color that was set up in your Institutional Preferences page to indicate local meta data.

The **Show Baseline** and **Delete Local** links appear to the right of the **Local Record** field after you save.

Add Source Names and Source Columns

Follow the steps below to add source names and source columns to a subject area for reporting and composite views

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS Meta Data or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default.

To move through the subject areas in alphabetical order, click **<-** or **->**. ■

3. Click **Select** on the View Target Report List page to choose the target subject area you want to edit. The Select A Subject Area (Banner ODS) or Select a Star (Banner EDW) window opens.

4. Click either the **Reporting View(s)** or **Composite View(s)** radio group (Banner ODS) or Target or Source (Banner EDW). Then, click the **Target(s)** radio group.

5. Click the subject area you want to view. The window closes automatically.

Note

Check the **Show locally modified targets only** checkbox to display local target views only. ■

6. Click **Add Source**. The Add Source window opens.

7. Enter the new source name. Click **Add Source** to save the new name. The View Source Report page opens displaying the new source name.

8. Click **Add Source Column** to add columns to the source. The Add a New Column window opens.

9. Enter the new column information, then click **Add Colum**n to save. The Edit Source Columns page opens and displays the new source column information.

Edit Source Names and Source Columns

Follow the steps below to change the properties of your source meta data for reporting and composite views.

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS Meta Data or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default.

To move through the subject areas in alphabetical order, click **<-** or **->**. ■

3. Click **Select** on the View Target Report List page to choose the target subject area you want to edit. The Select A Subject Area (Banner ODS) or Select a Star (Banner EDW) window opens.

4. Click either the **Reporting View(s**) or **Composite View(s)** radio group (Banner ODS) or Target or Source (Banner EDW). Then, click the **Target(s)** radio group.

5. Click the subject area you want to view. The window closes automatically.

Note

Check the **Show locally modified Sources only** checkbox to display local source views only. ■

6. Click the source name you want to change. The View Source Report page opens.

7. Choose the source column you want to change. The Edit Source Columns page opens.

8. Enter your changes. Click **Save Changes** to keep your changes.

Delete Local Source Properties

Follow the steps below to delete local source properties:

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS Meta Data or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default. To move through the subject areas in alphabetical order, click **<-** or **->**. ■

3. Click **Select** on the View Target Report List page to choose the target subject area you want to edit. The Select A Subject Area (Banner ODS) or Select a Star (Banner EDW) window opens.

4. Click either the **Reporting View(s**) or **Composite View(s)** radio group (Banner ODS) or Target or Source (Banner EDW). Then, click the **Target(s)** radio group.

5. Click the subject area you want to view. The window closes automatically.

Note

Check the **Show locally modified sources only** checkbox to display local target views only. ■

6. Click the source name whose source properties you want to delete. The View Source Report page opens.

7. Click the **Delete**. A message window appears.

8. Click **OK** to delete the source, or **Cancel** to keep the source. If you delete the source, the View Source Report List page opens. If you keep the source, you remain on the View Source Report page.

Delete Local Source Columns

Follow the steps below to change the properties of your source meta data.

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS Meta Data or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default. To move through the subject areas in alphabetical order, click **<-** or **->**. ■

3. Click **Select** on the View Target Report List page to choose the target subject area you want to edit. The Select A Subject Area (Banner ODS) or Select a Star (Banner EDW) window opens.

4. Click either the **Reporting View(s**) or **Composite View(s)** radio group (Banner ODS) or Target or Source (Banner EDW). Then, click the **Target(s)** radio group.

5. Click the subject area you want to view. The window closes automatically.

Note

Check the **Show locally modified sources only** checkbox to display local target views only. ■

6. Click the source name whose columns you want to delete. The View Source Report page opens.

7. Click the source column you want to delete. The Edit Source Column page opens.

8. Click the **Delete Local**. A message window appears.

9. Click **OK** to delete the source, or **Cancel** to keep the source. If you delete the source, you return to the Edit Source Column page. If you keep the source, you remain on the Edit Source Column page.

Add and Delete Source to Target Meta Data Local Mappings

Meta data contains information about which source column in the source system contained the information that is in the target column. You can create your own local source to target meta data mappings.

Follow the steps below to add or delete local source to target mappings to the meta data:

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS Meta Data or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default. To move through the subject areas in alphabetical order, click **<-** or **->**. ■

3. Click **Select** on the View Target Report List page to choose the target subject area you want to edit. The Select A Subject Area (Banner ODS) or Select a Star (Banner EDW) window opens.

4. Click either the **Reporting View(s**) or **Composite View(s)** radio group (Banner ODS) or Target or Source (Banner EDW). Then, click the **Target(s)** radio group.

5. Click the subject area you want to view. The window closes automatically.

Note

Check the **Show locally modified targets only** checkbox to display local target views only. ■

6. Click the reporting view to map. The View Target Report page opens.

7. Click the target column you want to map. The Edit Target Columns page opens.

8. *To Add:*
   Click **Add Local Mapping** at the bottom of the web page. The Add a Source Mapping window opens. (Continue to the next step below.)

   To Delete:
   Click the **Delete Local Mapping** link at the bottom of the web page.

   Click **OK** to delete the local mapping.

9. Enter the source subject area, table and column (required fields). Or, search for them using the corresponding links. Choose the table or column from the drop-down list associated with that link.

10. Click **Add Mapping** to save the newly mapped meta data.

Import Target and Source Meta Data

The Import option enables you to import an entire view into the meta data. The view must exist in the ODSMGR schema. All the columns in the view are created as local meta data.

Follow the steps below to change the properties of your source meta data.

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS Meta Data or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default. To move through the subject areas in alphabetical order, click **<-** or **->**. ■

3. Click **Select** on the View Target Report List page to choose the target subject area you want to edit. The Select A Subject Area (Banner ODS) or Select a Star (Banner EDW) window opens.

4. Click either the **Reporting View(s**) or **Composite View(s)** radio group (Banner ODS) or Target or Source (Banner EDW). Then, click the **Target(s)** radio group.

5. Click the subject area you want to view. The window closes automatically.

Note

Check the **Show locally modified targets (or sources) only** checkbox to display local target (or source) views only. ■

6. Click **Import** located at the top right side of the web page. The Select a View window opens.

7. Click one or more views to import.

To choose more than one view, click the first view, the hold down the Ctrl key while selecting the remaining views.

8. Click **Import**.

CSV Export

The Export option enables you to export target and source meta data into a .csv file that you can open in Microsoft Excel, or similar application.

Follow the steps below to change the properties of your source meta data.

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS Meta Data or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default. To move through the subject areas in alphabetical order, click **<-** or **->**.

3. Click **Select** on the View Target Report List page to choose the target subject area you want to edit. The Select A Subject Area (Banner ODS) or Select a Star (Banner EDW) window opens.

4. Click either the **Reporting View(s**) or **Composite View(s)** radio group (Banner ODS) or Target or Source (Banner EDW). Then, click the **Target(s)** radio group.

5. Choose the subject area you want to view. The window closes automatically.

Note

Check the **Show locally modified targets only** checkbox to display local target views only.

6. To export all reporting or composite views in a subject area, click **CSV Export** located at the top right side of the View Target (or Source) Report List page.

7. A window opens either to warn you that the operation will take a long time, or to indicate whether you want to save or open the file. Click **Cancel** to stop.

Publish Meta Data from the Administrative UI

Meta data is considered ‘published’ after the selected source or target information is saved as an HTML file and a meta data report is created. And, it can be published for some or all sources and targets. Meta data enables users to easily view the relationships between Banner ODS and Banner EDW columns and their sources. Meta data can be published from the Administrative UI, or from the command line outside the Administrative UI. Once a meta data report is published, it can be stored on a server that is accessible to reporting users.

Note

If the Web server is not on the Banner ODS and Banner EDW machine, the files need to be copied to the Web server after publishing.

Publish Meta Data for an Entire Subject Area

Follow the steps below to publish meta data for an entire subject area (Student, Finance, etc.).

1. Click **Meta Data** from the Administrative UI menu.
2. Click **Maintain Banner ODS or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default. To move through the subject areas in alphabetical order, click **<-** or **->**.

Note

Check the **Show locally modified targets only** checkbox to display local target views only.

3. Click **Select** on the View Target Report List page. The Select A Subject Area window opens.
4. Click the **Target(s)** radio group.
5. Click the subject you want to view. The window closes automatically.
6. Click **Publish** located at the top right side of the web page.
7. Click **Ok** to confirm that you want to publish all reports for the subject area.

Publish Meta Data for One Source or Target

Follow the steps below to preview and publish the meta data for one source or target.

1. Click **Meta Data** from the Administrative UI menu.

2. Click **Maintain Banner ODS or Banner EDW Meta Data.** The View Target Report List page opens.

Note

All subject areas on the View Target Report List page display in alphabetical order by default. To move through the subject areas in alphabetical order, click **<-** or **->**.

Note

Check the **Show locally modified targets only** checkbox to display local target views only.

3. Click **Select** on the View Target Report List page. The Select A Subject Area window opens.

4. Click the **Target(s)** radio group.

5. Click the subject you want to view. The window closes automatically.

6. Click the reporting view whose meta data you want to preview or publish.

7. Click Preview to open the View Target Report List page, and preview the report. The meta data is not permanently published until you complete the following step.

8. Click **Publish** at the top of the web page. An HTML file is published (saved as a report). The file is saved to the location specified by the parameters with an internal group *METADATA* and internal_code_2= *PUBLISH_LOCATION*.

Publish Meta Data Reports

Meta data can be published using three methods.

- Publish all meta data by scheduling a process. See “Publish Meta Data by Scheduling a Process” on page 4-170
- Publish for an entire subject area. See “Publish Meta Data for an Entire Subject Area” on page 4-168

- Publish for one source or target. See “Publish Meta Data for One Source or Target” on page 4-169

Note

Baseline meta data reports are provided when your solution is installed. Therefore, you should *not* need to perform the publishing step initially. ■

Publish Meta Data by Scheduling a Process

You can schedule meta data to publish at a predetermined day and time. Follow the steps in the “Schedule a Process (Banner ODS and Banner EDW)” on page 4-65 section. You should click the **Publish Meta Data** process.

Publish Meta Data from the Command Line

You can publish all meta data reports using the MGKPUBL.P_MakeAllReports procedure. A sample script, PUBLISH.SQL, is provided in the dbscripts/utility_scripts for publish.sql. To generate all the meta data reports, use the following command:

```
SQLPLUS IA_ADMIN/<password> @PUBLISH.SQL
```

The following `PUBLISH.BAT` script (in the `web_files/metadata` directory) can be customized to perform the entire process (generating the files, and then using FTP to put them on a remote server):

```
if "%1" == "move" goto movem

echo Publishing...

echo SET SERVEROUTPUT ON SIZE 500000 > doit.sql

echo EXEC MGKPUBL.P_MakeAllReports >> doit.sql

echo QUIT >> doit.sql

type doit.sql

sqlplus ia_admin/<password>@<Oracle database> @doit.sql

:movem

echo Moving...

if exist *.html del *.html

ftp -n -s:getfiles.dat <ODS machine>

ftp -n -s:putfiles.dat <web server machine>
```

View Published Meta Data

Meta data is considered ‘published’ after the selected source or target information is saved as an HTML file and, as a result, a meta data report is created. There are two kinds of reports for reporting view and composite view meta data. They are target reports and source reports.

Target Reports:
Show the relationship between the columns in Banner ODS or Banner EDW reporting views (or composite views) and the columns to which they are mapped in the source system tables.

Source Reports:
Show the relationship between columns in the source system tables and the columns to which they are mapped in Banner ODS or Banner EDW reporting view (or composite view).

Reporting View Meta Data

Use the following steps to view a published reporting view meta data report.

1. Click **Meta Data** from the Administrative menu.
2. Click Banner Operational Data Store or Banner Enterprise Data Warehouse. Banner Operational Data Store or Banner Enterprise Data Warehouse Reporting View Meta Data Reports page opens.
3. Choose a subject area from Banner Operational Data Store or Banner Enterprise Data Warehouse Reporting View Meta Data Reports page. The Reporting View Meta Data Reports page opens.
4. Choose a reporting view. The selected report displays.

Note

Sometimes the number of targets in the source report can exceed a 30,000 character limit. If this happens the output for the source is cut off, and a message *“(More Targets…)”* displays. ■

Composite View Meta Data

Banner ODS composite view meta data is also available as published meta data. Use the following steps to view published composite view meta data reports.

1. Click **Meta Data** from the Administrative menu.
2. Click **Banner Operational Data Store**. The Reporting View Meta Data Reports page opens.

3. Click **Banner ODS Composite View Meta Data Reports** located in the top right-hand corner of Banner Operational Data Store Reporting View Meta Data Reports page. The Banner Operational Data Store Composite View Meta Data Reports page opens.

4. Choose the subject area. The Composite View Meta Data Reports page opens listing the view name and description.

5. To view the column details associated with the selected composite view, choose one of the composite views. A report opens listing the Local Target, Target Column, Business Definition, Database Data Type, Source Name and Source Column.

Metamodel

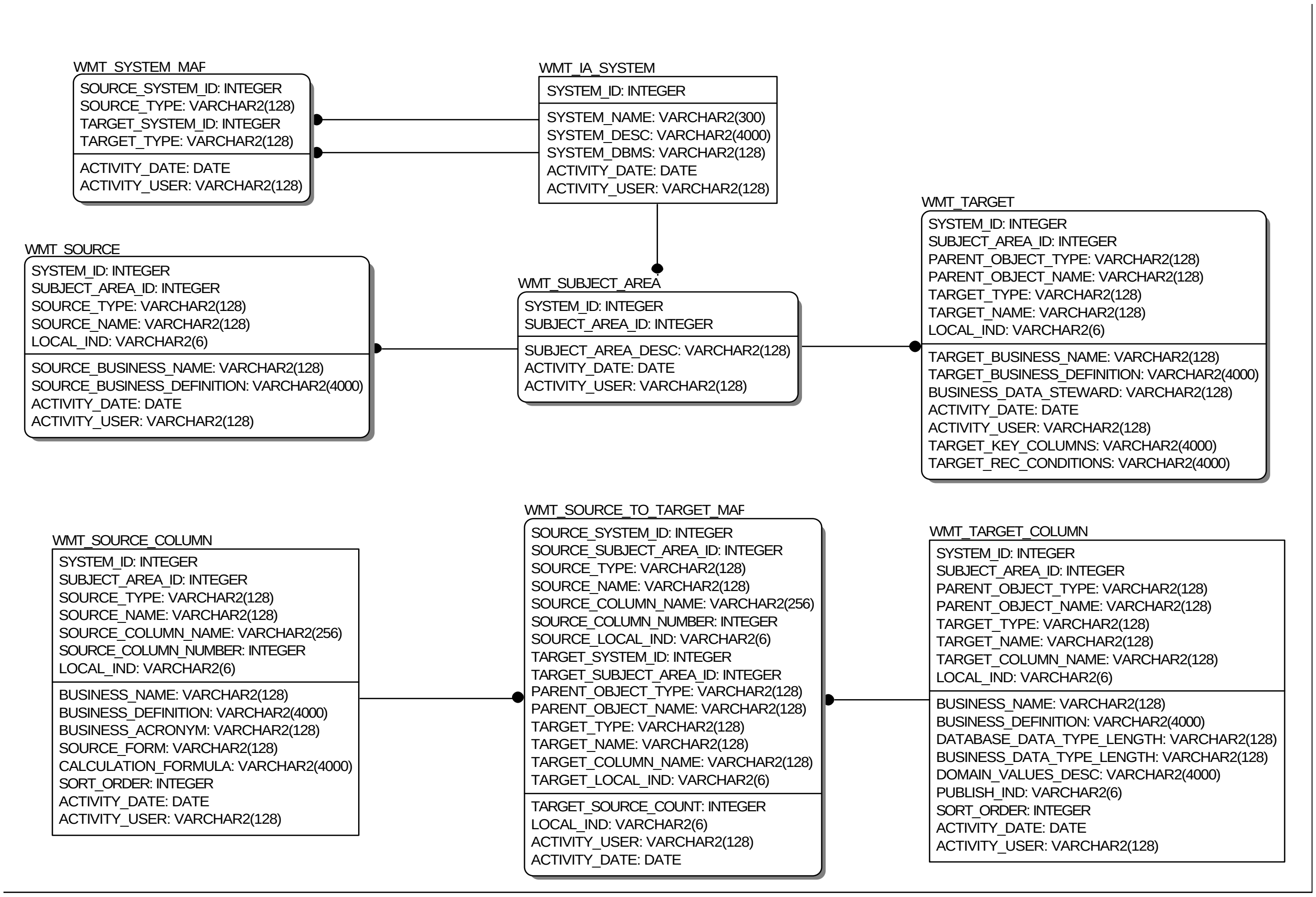
The delivered metamodel is the physical relational data model that stores the meta data. This should not be confused with the meta data repository, which refers to the physical database tables that contain the meta data.

Meta data tables are stored in a repository that is owned by the user - IA_ADMIN. Each table in the meta data repository begins with a “WMT_” prefix to identify it as a Banner ODS/Banner EDW “Warehouse Meta Data Table.” In addition, there is a public synonym for each table that simply removes the “WMT_” prefix.

The meta data tables and views that make up the metamodel illustrate the different pieces of meta data available, and how they relate to each object type. The object types are the reporting views and the source tables.

Meta Data Table Name	Synonym
WMT_IA_SYSTEM	IA_SYSTEM
WMT_SOURCE	SOURCE
WMT_SOURCE_COLUMN	SOURCE_COLUMN
WMT_SOURCE_TO_TARGET_MAP	SOURCE_TO_TARGET_MAP
WMT_SUBJECT_AREA	SUBJECT_AREA
WMT_SYSTEM_MAP	SYSTEM_MAP
WMT_TARGET	TARGET
WMT_TARGET_COLUMN	TARGET_COLUMN

A diagram of the metamodel follows:


WMT_SYSTEM_MAP
SOURCE_SYSTEM_ID: INTEGER
SOURCE_TYPE: VARCHAR2(128)
TARGET_SYSTEM_ID: INTEGER
TARGET_TYPE: VARCHAR2(128)
ACTIVITY_DATE: DATE
ACTIVITY_USER: VARCHAR2(128)
WMT_IA_SYSTEM
SYSTEM_ID: INTEGER
SYSTEM_NAME: VARCHAR2(300)
SYSTEM_DESC: VARCHAR2(4000)
SYSTEM_DBMS: VARCHAR2(128)
ACTIVITY_DATE: DATE
ACTIVITY_USER: VARCHAR2(128)
WMT_TARGET
SYSTEM_ID: INTEGER
SUBJECT_AREA_ID: INTEGER
PARENT_OBJECT_TYPE: VARCHAR2(128)
PARENT_OBJECT_NAME: VARCHAR2(128)
TARGET_TYPE: VARCHAR2(128)
TARGET_NAME: VARCHAR2(128)
LOCAL_IND: VARCHAR2(6)
TARGET_BUSINESS_NAME: VARCHAR2(128)
TARGET_BUSINESS_DEFINITION: VARCHAR2(4000)
BUSINESS_DATA_STEWARD: VARCHAR2(128)
ACTIVITY_DATE: DATE
ACTIVITY_USER: VARCHAR2(128)
TARGET_KEY_COLUMNS: VARCHAR2(4000)
TARGET_REC_CONDITIONS: VARCHAR2(4000)
WMT_SOURCE
SYSTEM_ID: INTEGER
SUBJECT_AREA_ID: INTEGER
SOURCE_TYPE: VARCHAR2(128)
SOURCE_NAME: VARCHAR2(128)
LOCAL_IND: VARCHAR2(6)
SOURCE_BUSINESS_NAME: VARCHAR2(128)
SOURCE_BUSINESS_DEFINITION: VARCHAR2(4000)
ACTIVITY_DATE: DATE
ACTIVITY_USER: VARCHAR2(128)
WMT_SUBJECT_AREA
SYSTEM_ID: INTEGER
SUBJECT_AREA_ID: INTEGER
SUBJECT_AREA_DESC: VARCHAR2(128)
ACTIVITY_DATE: DATE
ACTIVITY_USER: VARCHAR2(128)
WMT_SOURCE_TO_TARGET_MAP
SOURCE_SYSTEM_ID: INTEGER
SOURCE_SUBJECT_AREA_ID: INTEGER
SOURCE_TYPE: VARCHAR2(128)
SOURCE_NAME: VARCHAR2(128)
SOURCE_COLUMN_NAME: VARCHAR2(256)
SOURCE_COLUMN_NUMBER: INTEGER
SOURCE_LOCAL_IND: VARCHAR2(6)
TARGET_SYSTEM_ID: INTEGER
TARGET_SUBJECT_AREA_ID: INTEGER
PARENT_OBJECT_TYPE: VARCHAR2(128)
PARENT_OBJECT_NAME: VARCHAR2(128)
TARGET_TYPE: VARCHAR2(128)
TARGET_NAME: VARCHAR2(128)
TARGET_COLUMN_NAME: VARCHAR2(128)
TARGET_LOCAL_IND: VARCHAR2(6)
TARGET_SOURCE_COUNT: INTEGER
LOCAL_IND: VARCHAR2(6)
ACTIVITY_USER: VARCHAR2(128)
ACTIVITY_DATE: DATE
WMT_SOURCE_COLUMN
SYSTEM_ID: INTEGER
SUBJECT_AREA_ID: INTEGER
SOURCE_TYPE: VARCHAR2(128)
SOURCE_NAME: VARCHAR2(128)
SOURCE_COLUMN_NAME: VARCHAR2(256)
SOURCE_COLUMN_NUMBER: INTEGER
LOCAL_IND: VARCHAR2(6)
BUSINESS_NAME: VARCHAR2(128)
BUSINESS_DEFINITION: VARCHAR2(4000)
BUSINESS_ACRONYM: VARCHAR2(128)
SOURCE_FORM: VARCHAR2(128)
CALCULATION_FORMULA: VARCHAR2(4000)
SORT_ORDER: INTEGER
ACTIVITY_DATE: DATE
ACTIVITY_USER: VARCHAR2(128)
WMT_TARGET_COLUMN
SYSTEM_ID: INTEGER
SUBJECT_AREA_ID: INTEGER
PARENT_OBJECT_TYPE: VARCHAR2(128)
PARENT_OBJECT_NAME: VARCHAR2(128)
TARGET_TYPE: VARCHAR2(128)
TARGET_NAME: VARCHAR2(128)
TARGET_COLUMN_NAME: VARCHAR2(128)
LOCAL_IND: VARCHAR2(6)
BUSINESS_NAME: VARCHAR2(128)
BUSINESS_DEFINITION: VARCHAR2(4000)
DATABASE_DATA_TYPE_LENGTH: VARCHAR2(128)
BUSINESS_DATA_TYPE_LENGTH: VARCHAR2(128)
DOMAIN_VALUES_DESC: VARCHAR2(4000)
PUBLISH_IND: VARCHAR2(6)
SORT_ORDER: INTEGER
ACTIVITY_DATE: DATE
ACTIVITY_USER: VARCHAR2(128)

These meta data tables that store information about the meta data are further described in the “Banner ODS Meta Data Object Types” section.

Banner ODS Meta Data Object Types

Information exists in the meta data layer for the following types of objects:

Object	Description
Target View	Banner ODS reporting views that join related information from Banner ODS tables. Use these views to build reports. *Example*: CONSTITUENT reporting view is the Advancement constituent data.
Source Table	Database tables in your source system used as the source for the data in Banner ODS. *Example* APBCONS is the Constituent Base Table.
Source Function	Functions that use data from the source system's source tables to create new data to be stored in Banner ODS and Banner EDW.

Banner EDW Meta Data Object Types

Meta data information exists for the following types of objects:

Object	Description
Target Table	Banner EDW fact and dimension tables that represent the data from Banner ODS reformatted into star schema.
Source View	Reporting views in Banner ODS

Source Meta Data Tables

The following meta data tables store information about the source of Banner ODS and Banner EDW data. In Banner ODS and Banner EDW, this is meta data about the source systems.

Source Table (WMT_SOURCE)

Columns	Descriptions
System ID	Unique ID for the source system.
Subject Area ID	Unique ID for the subject area.

Columns	Descriptions
Source Type	Identifies whether the source is a table, view, or function. Sample source types are TABLE, REPORTING VIEW and FUNCTION.
Source Name	Source table, view, or function name.
Source Business Name	Source table, view, or function descriptive name.
Source Business Definition	Table or view business purpose.
Local Ind	Indicates whether the row is a local or baseline version.
Activity Date	Date the meta data was changed.
Activity User	User who changed the meta data.

Source Column Table (WMT_SOURCE_COLUMN)

Columns	Descriptions
System ID	Unique ID for the source system.
Subject Area ID	Unique ID for the subject area.
Source Type	Identifies whether the source is a table, view or function. Sample source types are TABLE, REPORTING VIEW, and FUNCTION.
Source Name	Source table, view or function name.
Source Column Name	Source table/view column name. If the source name is FUNCTION, the function name is entered. If the source name is CONSTANT, the value of the constant is entered. If the source name is CALCULATION, the calculation is entered.
Source Column Number	Distinguishes between source columns that have the same names.
Local Ind	Indicates whether the row is a local or baseline version.
Business Name	Descriptive name for the column in the source.
Business Definition	Source column defined in business terms.
Business Acronym	Acronym for the source column, if it has one.

Columns	Descriptions
Source Form	Source system form name from which the data was captured.
Calculation Formula	Any calculations that are applied to create the data in the target column.
Sort Order	Column order in the table or view. It is determined by numbering the columns in alphabetical order.
Activity Date	Date the meta data was changed.
Activity User	User who changed the meta data.

Target Meta Data Tables

The following meta data tables store information about the target of Banner ODS and Banner EDW data, Banner ODS reporting views or Banner EDW stars, fact, and fact or dimension tables.

Target Table (WMT_TARGET)

Columns	Descriptions
System ID	Unique ID for Banner ODS and Banner EDW.
Subject Area ID	Unique ID for the subject area.
Parent Object Type	This column is used in Banner EDW only. In the case of Banner EDW, the parent object type is STAR. Not used in Banner ODS.
Parent Object Name	This column is used in Banner EDW only. In Banner EDW, this identifies the star to which the target belongs.
Target Type	Stores whether this is a Banner ODS and Banner EDW table or view. Currently, reporting and composite view information is available. Sample values for Banner ODS are REPORTING VIEW and COMPOSITE VIEW. Sample values for the EDW are DIMENSION TABLE, FACT TABLE, and STAR.
Target Name	Table or view name.

Columns	Descriptions
Target Business Name	Target descriptive name.
Target Business Definition	Target business purpose.
Business Data Steward	Person or department responsible for the data in the target.
Local Ind	Indicates whether the row is a local or baseline versions.
Activity Date	Date the meta data was changed.
Activity User	User who changed the meta data.
Target Key Column	Describes how the data is to be returned when extracted, with any information and/or comments specific to this particular set of data.
Target Rec Conditions	Columns used in report filters and queries that return the best performance for the specified reporting view. These conditions are not mandatory, but recommended for performance. You may retrieve data from the reporting views using different criteria.

Target Column Table (WMT_TARGET_COLUMN)

Columns	Descriptions
System ID	Unique ID for Banner ODS and Banner EDW.
Subject Area ID	Unique ID for the subject area.
Parent Object Type	This column is used in Banner EDW only. In Banner EDW, the parent object type is STAR.
Parent Object Name	This column is used in Banner EDW only. In Banner EDW, this identifies the star to which the target belongs.
Target Type	Stores whether this is a Banner ODS and Banner EDW table or view. Currently, reporting view information is available. Sample values for Banner ODS are REPORTING VIEW and COMPOSITE VIEW. Sample values for Banner EDW are DIMENSION TABLE, FACT TABLE and STAR.
Target Name	Table or view name.

Columns	Descriptions
Target Column Name	Target column name.
Local Ind	Indicates whether the row is a local or baseline version.
Business Name	Descriptive name for the column in the target.
Business Definition	Defines the target column in business terms. This is the comment on column in the relational database data dictionary in your target system.
Database Data Type Length	Comes from the relational database data dictionary in Banner ODS and Banner EDW. This is stored in the meta data tables, not just the relational database data dictionary, so that it is easily available in one place with the rest of the meta data.
Business Data Type Length	Used when writing reports for formatting purposes. The business data type may be character, integer, float, etc. It also contains the length of the data. ***Example*** The relational database data type and length for the internal ID may be varchar(63), but the business data type and length is 8-digits. Even though the database allows for a width up to 63 characters, the column will never be more than eight.
Domain Values Desc	Description of the valid values that a column can contain. It could be a list of codes and code descriptions.
Publish Ind	Indicates whether to publish the column information to meta data reports so users can use the meta data for reporting purposes. It may not be published because the column contains sensitive information. The column may also contain technical information like a key that would not be used in a report.
Sort Order	Physical order of the columns in the table or view from the relational database data dictionary.
Activity Date	Date the meta data was changed.
Activity User	User who changed the meta data.

Source and Target Meta Data Tables

The following meta data tables store information about the source and target of the data. This includes meta data about the source systems and Banner ODS and Banner EDW.

System Table (WMT_IA_SYSTEM)

Columns	Descriptions
System Id	Unique ID for a system.
System Name	Administrative source or Banner ODS/Banner EDW solution system name.
System Desc	Administrative source or Banner ODS/Banner EDW solution system description.
System DBMS	Database management system software, Oracle for example, used to implement the source or target system.
Activity Date	Date the meta data was changed.
Activity User	User who changed the meta data.

Subject Area Table (WMT_SUBJECT_AREA)

Columns	Descriptions
System ID	Unique ID for the system.
Subject Area ID	Unique ID for the subject area.
Subject Area Desc	Advancement, Student or Human Resources, for example.

System Map Table (WMT_SYSTEM_MAP)

Columns	Descriptions
Source System ID	Source system unique ID.
Source Type	Identifies whether the source is a table, view, or function. Sample source types are TABLE, REPORTING VIEW, and FUNCTION.
Target System Id	Banner ODS and Banner EDW unique ID.

Columns	Descriptions
Target Type	Stores whether this is a Banner ODS and Banner EDW table or view. Currently, reporting and composite view information is available. Sample values for Banner ODS are REPORTING VIEW and COMPOSITE VIEW. Sample values for Banner EDW are DIMENSION TABLE, FACT TABLE, and STAR.
Activity Date	Date the meta data was changed.
Activity User	User who changed the meta data.

Source to Target Map Table (WMT_SOURCE_TO_TARGET_MAP)

Columns	Descriptions
Source System ID	Source system unique ID.
Source Subject Area ID	Subject area unique ID.
Source Type	Identifies whether the source is a table, view or function. Sample source types are TABLE, REPORTING VIEW and FUNCTION.
Source Name	Source table, view or PL/SQL function name.
Source Column Name	Source column name from the source table or view, if the source is a table or view. If the source name is FUNCTION, the function name is entered. If the source name is CONSTANT, the value of the constant is entered. If the source name is CALCULATION, the calculation is entered.
Source Column Number	Distinguishes between source columns that have the same names.
Source Local Ind	Indicates whether the row is a local or baseline version.
Target System Id	Banner ODS and Banner EDW unique ID.
Target Subject Area Id	Subject area unique ID.
Parent Object Type	This column is used in Banner EDW only. In Banner EDW, the parent object type is STAR.

Columns	Descriptions
Parent Object Name	This column is used in Banner EDW only. In Banner EDW, this identifies the star to which the target belongs.
Target Type	Stores whether this is a Banner ODS and Banner EDW table or view. Currently, reporting and composite view information is available. Sample values for Banner ODS are REPORTING VIEW and COMPOSITE VIEW. Sample values for Banner EDW are DIMENSION TABLE, FACT TABLE, and STAR
Target Name	Table or view name.
Target Column Name	Column name in the target reporting view.
Target Local Ind	Indicates whether the row is a local or baseline version.
Target Source Count	Count indicates how many sources there are for a target.
Local Ind	Indicates whether the row is a local or baseline version.
Activity User	User who changed the meta data.
Activity Date	Date the meta data was changed.

Reporting Meta Data Views

The following views exist in the meta data repository, and are owned by the user IA_ADMIN.

Views	Descriptions
WMV Source	Lists all information associated with sources and source columns.
WMV Source To Target Map	Lists all information associated with sources, targets, and source and target columns.
WMV Target	Lists all information associated with targets and target columns.

Each view joins a specific combination of the data stored within the meta data tables. You can use these views to query and report the meta data information. They provide easier access to the meta data in the same way that Banner ODS reporting views provide access to the data in Banner ODS and Banner EDW tables.

Source Meta Data View (WMV_SOURCE)

Views	Descriptions
System Name	Administrative source or Banner ODS/Banner EDW solution system name.
System Desc	Administrative source or Banner ODS/Banner EDW solution system description.
Subject Area Desc	Advancement, Student or Human Resources, for example.
Source Type	Identifies whether the source is a table, view or function. Sample source types are TABLE, REPORTING VIEW and FUNCTION.
Source Name	Source table, view or PL/SQL function name.
Source Business Name	Source descriptive name.
Source Business Definition	Source business purpose description.
Source Column Name	Source column name from the source, if the source is a table or view. Function name if the source is a function.
Business Definition	Target column description in business terms.
Calculation Formula	Any calculations that are applied to create the data in the target column.
Sort Order	Column order in the table or view. It is determined by numbering the columns in alphabetical order.
Business Name	Column name in the source.
Business Acronym	Source column acronym, if it has one.
Source Form	Source system form name from which the data was captured.
Local Ind	Indicates whether the row is a local or baseline version.

Source to Target Map Meta Data View (WMV_SOURCE_TO_TARGET_MAP)

Views	Descriptions
Target System Name	Solution system name.
Target System Desc	Solution system description.
Target Subject Area Desc	Advancement, Student or Human Resources, for example.
Parent Object Type	This column is used in Banner EDW only. In the case of Banner EDW, the parent object type is STAR.
Parent Object Name	This column is used in Banner EDW only. In the case of Banner EDW this identifies the star to which the target belongs.
Target Type	Stores whether this is a Banner ODS and Banner EDW table or view. Currently Reporting Views information is available. A sample value for Banner ODS is REPORTING VIEW. Sample values for Banner EDW are DIMENSION TABLE, FACT TABLE and STAR.
Target Name	Table or view name.
Target Business Name	Target descriptive name.
Target Business Definition	Target business purpose.
Business Data Steward	Person or department responsible for the data in the target.
Target Column Name	Target column name.
Target Column Business Name	Target column descriptive name.
Target Column Business Def	Target column description in business terms. This is the comment on column in the relational database data dictionary in your target system.

Views	Descriptions
Database Data Type Length	Comes from the relational database data dictionary in Banner ODS and Banner EDW. This is stored in the meta data tables, not just the relational database data dictionary, so that it is easily available, in one place with the rest of the meta data, for meta data users.
Business Data Type Length	Used when writing reports for formatting purposes. The business data type may be character, integer, float, etc. It also contains the length of the data. ***Example:*** The relational database data type and length for an internal ID may be varchar(63), but the business data type and length is eight digits. Even though the database allows for a width up to 63-characters, the column will never be more than eight.
Domain Values Desc	Description of the valid values that a column can contain. It could be a list of codes and code descriptions.
Publish Ind	A flag that indicates whether to publish the column information to meta data reports so users can use the meta data for reporting purposes. It may not be published because the column contains sensitive information. The column may also contain technical information like a key that would not be used in a report.
Target Sort Order	Columns physical order in the table or view from the relational database data dictionary.
Target Local Ind	Indicates whether the row is a local or baseline version.
Source System Name	Solution system name.
Source System Desc	Solution system description.
Source Subject Area Desc	Advancement, Student or Human Resources, for example.
Source Type	Identifies whether the source is a table, view or function. Sample source types are TABLE, REPORTING VIEW and FUNCTION.
Source Name	Source table, view or function name.

Views	Descriptions
Source Business Name	Source descriptive name.
Source Business Definition	Business purpose of the source.
Source Column Name	Source column name from the source, if the source is a table or view. Function name if the source is a function.
Source Column Business Name	Column in the source descriptive name.
Source Column Business Def	Source column described in business terms.
Business Acronym	Source column acronym, if it has one.
Calculation Formula	Any calculations that are applied to create the data in the target column.
Source Sort Order	Column order in the table or view. It is determined by numbering the columns in alphabetic order.
Source Form	Source system form name from which the data was captured.
Source Local Ind	Indicates whether the row is a local or baseline version.

Target Meta Data View (WMV_ TARGET)

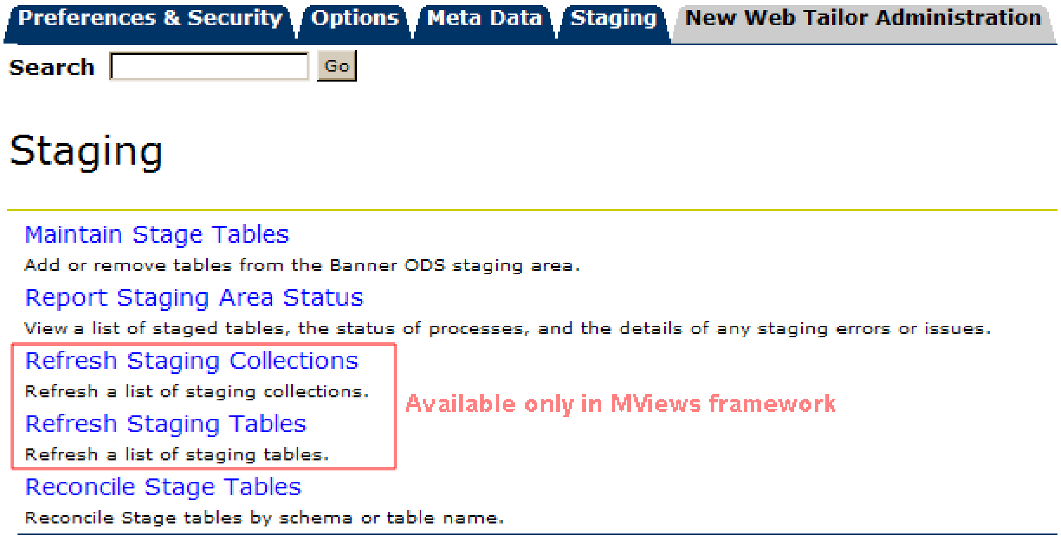
Views	Descriptions
System Name	Administrative source or Banner ODS/Banner EDW solution system name.
System Desc	Administrative source or Banner ODS/Banner EDW solution system description.
Subject Area Desc	Advancement, Student or Human Resources, for example.
Parent Object Type	This column is used in Banner EDW only. In the case of Banner EDW, the parent object type is STAR.

Views	Descriptions
Parent Object Name	This column is used in Banner EDW only. In the case of Banner EDW this identifies the star to which the target belongs.
Target Type	Stores whether this is a Banner ODS and Banner EDW table or view. Currently, Reporting and Composite View information is available. Sample values for Banner ODS are REPORTING VIEW and COMPOSITE VIEW. Sample values for Banner EDW are DIMENSION TABLE, FACT TABLE, and STAR.
Target Name	Table or view name.
Target Business Name	Target descriptive name.
Target Business Definition	Target business purpose.
Business Data Steward	Person or department responsible for the data in the target.
Target Column Name	Column name in the target.
Business Name	Descriptive name for the column in the target.
Business Definition	Target column in business terms. This is the comment on column in the relational database data dictionary in your target system.
Database Data Type Length	Comes from the relational database data dictionary in Banner ODS and Banner EDW. This is stored in the meta data tables, not just the relational database data dictionary, so that it is easily available in one place with the rest of the meta data, for meta data users.

Views	Descriptions
Business Data Type Length	Used when writing reports for formatting purposes. The business data type may be character, integer, float, etc. It also contains the length of the data. ***Example:*** The relational database data type and length for an internal ID may be varchar(63), but the business data type and length is 8-digits. Even though the database allows for a width up to 63 characters, the column can never be more than 8.
Domain Values Desc	Description of the valid values that a column can contain. It could be a list of codes and code descriptions.
Publish Ind	Indicates whether to publish the column information to meta data reports so users can use the meta data for reporting purposes. It may not be published because the column contains sensitive information. The column may also contain technical information like a key that would not be used in a report.
Sort Order	Columns physical order in the table or view from the relational database data dictionary.
Local Ind	Indicates whether the row is a local or baseline version.

Staging and data replication

You have the option to use either the Oracle Streams or Materialized Views architecture framework for staging data in Banner ODS. You chose which option to implement when you installed or upgraded Banner ODS. Regardless of which framework you use, you can maintain the staging environment using the options on the Staging menu of the Administrative UI.



Staging

Maintain Stage Tables
Add or remove tables from the Banner ODS staging area.

Report Staging Area Status
View a list of staged tables, the status of processes, and the details of any staging errors or issues.

Refresh Staging Collections
Refresh a list of staging collections.

Refresh Staging Tables
Refresh a list of staging tables.

Available only in MViews framework

Reconcile Stage Tables
Reconcile Stage tables by schema or table name.

You must assign an Administrative UI user the “BPRA Staging” role in Web Tailor to allow that user access to the Staging tab. Refer to the section “Update User Roles” for instructions on how to do this.

Note

Refer to “Extract, Transform, and Load process (ETL)” section of the “Architecture” chapter for more information about the Materialized Views architecture and how it is used by the target database.

Staging options

The following options on the Staging menu let you manage the staging environment. These options are available no matter which framework (Streams or MViews) you implement. The jobs allow you to do the following tasks.

- Maintain Stage Tables - add or delete non-baseline staging tables and schemas to or from the Banner ODS staging area. (Refer to the “Maintain stage tables” section for more information.)
- Report Staging Area Status - view a list of staged tables and perform checks on the status of various staging area items that may require user action, for example, it can

list unknown mviews or missing baseline staged mviews. (Refer to the "Staging Area Status" section for more information.)

- Reconcile Stage Tables - compare target database tables (materialized views in the MViews framework) to source tables and restage staging tables in the target database. (Refer to the "Reconcile Stage tables" section for more information.)

Materialized Views staging options

The following options on the Staging menu let you manage how and when to refresh and reconcile data in the target database *when you implement the Materialized Views framework as your approach to replicate data in the Banner ODS. These options only display when you implement the MViews framework.* The jobs allow you to do the following:

- Refresh Staging Collections - refresh a collection of materialized views
- Refresh Staging Tables - refresh selected materialized views

The actual jobs associated with these menu options are defined in the SUBPROCESS parameter and named as follows:

- RECONCILE_STAGE_SCHEMA
- RECONCILE_STAGE_TABLE

Maintain stage tables

If you want to include additional data from the source database that isn't included in your baseline target database, you need to create stage tables for the new data in the target database. Depending on your data, you may also need to create new schemas associated with the new tables.

You have the ability to add or remove non-baseline stage tables and add a schema using the **Maintain Stage Tables** page available from the **Staging** menu in the Administrative UI.

Add a non-baseline staging table to the Banner ODS

You may want to replicate data from source Banner tables that are not part of the baseline Banner ODS. Perform the following steps to add non-baseline stage tables to the Banner ODS.

1. Click **Staging** from the Administrative UI menu.
2. Click **Maintain Stage Tables**.

3. Choose a **Source Database**. You will only need to select a source database if your institution uses multiple source databases.

4. Click the table owner for the area of tables that you want to add.

5. Select tables from the **Tables to Add** list. Use Shift-click to select a contiguous range of tables or Ctrl-click to select noncontiguous tables.

6. Enter a **Run Date** (format dd-mon-yyyy) and **Runtime** (format hh24:mi:ss) to schedule when to run the job that will add the stage tables to the Banner ODS. Enter *NOW* in each field to run the job immediately.

7. Click **Submit** to schedule the job to run.

The selected tables are added to the Banner ODS stage environment. A local record for each table is also created in the MGBSTGE table if a record doesn't already exist in the table.

Remove a non-baseline staging table from the Banner ODS

Perform the following steps to remove local stage tables from the Banner ODS. You can only remove stage tables that are not part of baseline Banner ODS. These are the stage tables that your institution added locally.

1. Click **Staging** from the Administrative UI menu.

2. Click **Maintain Stage Tables**.

3. Choose a **Source Database**. You will only need to select a source database if your institution uses multiple source databases.

4. Click **Remove Stage Tables**.

5. Select tables from the **Tables to remove** list. Use Shift-click to select a contiguous range of tables or Ctrl-click to select noncontiguous tables.

6. Enter a **Run Date** (format dd-mon-yyyy) and **Runtime** (format hh24:mi:ss) to schedule when to run the job that will remove the stage tables from the Banner ODS. Enter *NOW* in each field to run the job immediately.

7. Click **Submit** to schedule the job to run.

The selected tables are removed from the Banner ODS stage environment. The local record for each table is also removed from the MGBSTGE table.

Add a schema

Depending on which data you want to add to the Banner ODS stage tables, you may need to add a user schema within the Administrative UI to obtain access to the additional table data. Use the following steps to add a schema to the Maintain Stage Tables Administrative UI page.

The following considerations apply when adding a schema.

- A schema must exist in both the Banner and Banner ODS databases before you can add it to this menu.
- The Default Tablespace name for the schema in Banner ODS must match the Default Tablespace name for the schema in Banner.
- The Default Tablespace name cannot be either SYSTEM or SYSAUX.

1. Click **Staging** from the Administrative UI menu.
2. Click **Maintain Stage Tables**.
3. Choose a **Source Database**. You will only need to select a source database if your institution uses multiple source databases.
4. Click **Add Another Schema to this List**.
5. Select a schema from the **Schema to Add** list.
6. Click **Submit**.

The selected schema is added to the list of tables on the Maintain Stage Tables page.

Remove a schema

Run the following command as IA_ADMIN on the Banner ODS to remove a schema from the list of available staging schemas.

```
SET SERVEROUTPUT ON
EXEC mgksstg.P_DelOwnerRecs(source alias, schema to remove);
```

Staging Area Status

You can get information about the state of the Banner ODS staging tables by running the Staging Area Status (STAGE_AREA_STATUS) process from the Administrative UI. When you run this job, the information included in the report differs depending on which staging framework your Banner ODS uses. Refer to the following section for the framework you implement.

Streams framework

When you run the Staging Area Status job against the Oracle Streams framework, you can get the following information.

- The source archive log files.
- The status of various Oracle Streams processes.
- Errors encountered by the apply process while making changes to the stage area.
- List of source tables currently in the staging area.
- The status of various staging area items.

Materialized Views framework

When you run the Staging Area Status job against the Materialized Views framework, you can get the following information.

- List all the staged tables in Banner ODS (including any non-baseline tables added to the staging area)

  or

- List staged tables that have at least a specified number of changes in them

Run Staging Area Status report

Perform the following steps from the Banner ODS or Banner EDW Administrative UI to run the Staging Area Status process.

1. Click **Staging** from the Administrative UI menu.
2. Click **Report Staging Area Status**.
3. Select the **Source Database** to identify which source database to run the report against. If your institution includes information from multiple sources in the Banner ODS, there will be one entry for each database in the Source Database drop-down list.
4. Check the items that you want to include in the report.
   - Check **Display Process Status** to include in the report status the relevant information about various Oracle Streams components. (Available only in Oracle Streams framework.)
   - Check **Display Apply Errors** to include in the report any errors the apply process encountered while making changes to the staging area. (Available only in Oracle Streams framework.)

- Check **Display Staged Tables** to include in the report a list of all source tables that are currently in the stage area.

  If you select the value *Yes, with Change Counts* for the **Display Staged Tables** field, you need to enter a **Change Count Limit** value as well. This value defines the minimum number of changes required on a stage table for it to get listed in the Staged Tables list.

- Check **Perform Staging Checks** to include in the report the status of various staging area items that may require user action.

Note

You don't need to perform the staging checks every time you run the Staging Area Status job. However, you will want to periodically run the report with the option to perform staging checks turned on so that you can ensure that the staging environment isn't encountering any of the issues flagged by the checks. ■

- Check the **Check Staging Triggers** option to include in the report any baseline table triggers that are missing from the staged tables replicated from the source database.

5. Enter a **Run Date** and **Run Time** to schedule when to run the job. Enter NOW in each field to run the job immediately.

6. Click **Submit**.

7. Click **View Control Reports**.

8. Select the **STAGE_AREA_STATUS** process associated with your User ID to view the status report.

Report Status Information	Report Includes
Required Source Archived Logs (Oracle Streams framework only)	Any source archived log files that are required by the Oracle Streams capture processes. The report includes the directory path where archived log files are saved. Refer to the "Required Source Archived Logs" later in this chapter for more information.
Process Status (Oracle Streams framework only)	Status and relevant information about the following Oracle Streams components: • Capture process - Status, State, First SCN, Last captured SCN, and Last applied SCN • Capture queue - Enqueue, Dequeue, Number of messages, and Spill messages • Propagation schedule - Status • Apply queue - Enqueue, Dequeue, Number of messages, and Spill messages • Apply process name - Status, Reader state, Coordinator state, Server state, and Apply tag
Apply Errors (Oracle Streams framework only)	Any errors the apply process encountered while making changes to the stage area are entered into the error queue.
Staged Tables List	All source tables that are currently in the stage area.

Report Status Information	Report Includes
Staging Checks (Oracle Streams framework)	When the **Perform Staging Checks** option is selected in the Oracle Streams framework, the report includes information on the following items. • Source table columns that are missing from the stage tables. • Banner ODS stage table triggers that are set to fire once. • Captured source tables that are not instantiated in the destination. • Instantiated destination tables that are not captured at the source.
Staging Checks (Materialized Views framework)	When the **Perform Staging Checks** option is selected in the Materialized Views framework, the report includes information on the following items. • Unknown (Non-Baseline) staging materialized views - lists any materialized views in one of the staging schemas that have not been created using the baseline process and are not recorded in the MGBSTGE table. • Missing baseline staging materialized views - lists any missing materialized views that should be in the baseline target database. • Staging materialized views that are not in a Refresh Group or Staging Collection - lists any Staging materialized views that are not in a Refresh Group or Refresh Staging Collection and are not getting refreshed by one of the Banner ODS processes. (This will only be an issue if you made a mistake while adding a new Refresh Group or changing a delivered Refresh Group.) • Refresh Groups not associated with an ODS Refresh process. (This will only be an issue if you reorganized Refresh Groups and didn't create the necessary Staging Collection record ETL PACKAGE parameter.) • Missing baseline warehouse staging indexes - lists any missing indexes that are specific to the baseline target database and are used to improve performance of the ETL mappings.

Maintain Oracle Streams framework

There are a number of tasks you may want to perform to maintain the Oracle Streams framework after it is implemented at your institution. For example, you can remove the Streams framework, remove baseline stage tables not used at your institution, or stop and start various Streams components. You perform these tasks outside the Administrative User Interface

Refer to the following topics for more information about maintaining the Streams architecture.

- “Create Streams Framework”
- “Remove Streams Framework”
- “Configure Streams Replication for Baseline Tables”
- “Remove a Baseline Staging Table from the Banner ODS”
- “Start or Stop the Streams Capture Process”
- “Start or Stop the Streams Propagation Schedule”
- “Start or Stop the Streams Apply Process”

Note

> The “source alias” specified in the following sections is the parameter that identifies the source database if you load data from multiple sources into the Banner ODS. Refer to the *Source Alias* section of the *Architecture* chapter for more information. ■

Create Streams Framework

The Streams framework includes the queues, propagation schedule, and capture and apply processes. The Streams framework is created during the install or upgrade process. If you need to recreate the Streams framework, perform the following steps:

1. Log in to the Banner ODS.

2. Issue the following commands:

```
    SET SERVEROUTPUT ON
    EXEC MGKSTRC.P_CREATE_LOCAL_ENV (database link, source
alias);
```

   where you enter your institution’s values for the parameters in parentheses.

Remove Streams Framework

The Streams framework includes the queues, propagation schedule, and capture and apply processes. Perform the following steps to remove the Streams framework.

Warning

Before you remove the Streams framework, you need to remove all staging tables from the Streams replication environment.

1. Log in to the Banner ODS.

2. Issue the following commands:

```
SET SERVEROUTPUT ON
EXEC MGKSTRC.P_DROP_LOCAL_ENV (source alias);
```

where you enter your institution’s values for the parameter in parentheses.

Configure Streams Replication for Baseline Tables

The MGBSTGE table stores the schemas and tables from Banner that will be included in the baseline Banner ODS Streams staging environment. Replication of these tables is performed during the install or upgrade process. If you need to configure the replication of Streams baseline Banner ODS tables at some point after installation, perform the following steps:

1. Log in to the Banner ODS.

2. Issue the following commands:

```
   SET SERVEROUTPUT ON
   EXEC MGKSTRM.P_ADD_TO_LOCAL_ENV(source alias, schema, table
name);
```

where you enter your institution’s values for the parameters in parentheses.

Note

You can add all of the tables in the MGBSTGE table at once by using the ‘%’ value for the ‘table name’ parameter.

Remove a Baseline Staging Table from the Banner ODS

The MGBSTGE table stores the schemas and tables from Banner that will be included in the baseline Banner ODS Streams environment. If you need to remove a baseline Banner ODS staging table from the Streams replication process, perform the following steps:

1. Log in to the Banner ODS.
2. Issue the following commands:

```
SET SERVEROUTPUT ON
EXEC MGKSTRM.P_REMOVE_FROM_LOCAL_ENV(source alias, schema,
table name);
```

   where you enter your institution’s values for the parameters in parentheses.

Note

You can remove all of the tables in the MGBSTGE table at once by using the ‘%’ value for the ‘schema’ and 'table name' parameters. ■

Start or Stop the Streams Capture Process

The Streams capture process identifies relevant changes in the Banner database redo log, converts them into logical change records, and puts them in a queue to be applied in the Banner ODS.

Perform the following steps to ***start*** *the Streams capture process.*

1. Log in to the Banner ODS.
2. Issue the following commands:

```
SET SERVEROUTPUT ON
EXEC MGKSTRM.P_START_CAPTURE(source alias);
```

   where you enter your institution’s values for the parameter in parentheses.

Perform the following steps to ***stop*** *the Streams capture process.*

1. Log in to the Banner ODS.
2. Issue the following commands:

```
SET SERVEROUTPUT ON
EXEC MGKSTRM.P_STOP_CAPTURE(source alias);
```

   where you enter your institution’s values for the parameter in parentheses.

Start or Stop the Streams Propagation Schedule

The Streams propagation schedule moves the change messages identified by the capture process from the source Banner database queue to a queue on the Banner ODS.

*Perform the following steps to **start** the Streams propagation schedule.*

1. Log in to the Banner ODS.

2. Issue the following commands:

```
SET SERVEROUTPUT ON
EXEC MGKSTRM.P_START_PROPAGATION(source alias);
```

   where you enter your institution’s values for the parameter in parentheses.

*Perform the following steps to **stop** the Streams propagation schedule.*

1. Log in to the Banner ODS.

2. Issue the following commands:

```
SET SERVEROUTPUT ON
EXEC MGKSTRM.P_STOP_PROPAGATION(source alias);
```

   where you enter your institution’s values for the parameter in parentheses.

Start or Stop the Streams Apply Process

The Streams apply process in the Banner ODS removes change messages from the queue and applies them directly to the destination tables in the Banner ODS.

*Perform the following steps to **start** the Streams apply process.*

1. Log in to the Banner ODS.

2. Issue the following commands:

```
SET SERVEROUTPUT ON
EXEC MGKSTRM.P_START_APPLY(source alias);
```

   where you enter your institution’s values for the parameter in parentheses.

*Perform the following steps to **stop** the Streams apply process.*

1. Log in to the Banner ODS.

2. Issue the following commands:

```
SET SERVEROUTPUT ON
EXEC MGKSTRM.P_STOP_APPLY(source alias);
```

where you enter your institution’s values for the parameter in parentheses.

Required Source Archived Logs

Archived log files on the source Banner system may be required by the Oracle Streams capture processes.These log files track any changes to the source Banner system. The capture process uses information in the archived log files to synchronize the Banner ODS staging tables with the source Banner tables.

Be sure to keep the archived log files in the archive directory on the source Banner database server. If the capture process is unable to locate a required archived log file, then the capture will abort. When this happens, restore the archived log file to its expected location so the capture process can continue replicating changes to the Banner ODS staging area.

The Staging Area Status process control report includes the directory path where archived log files are saved. The following query displays the archive redo log files required by the Streams capture process:

```
SELECT R.CONSUMER_NAME "Capture Process Name",
          R.SOURCE_DATABASE "Source Database",
          R.SEQUENCE# "Sequence Number",
          R.NAME "Required Archived Log Name"
     FROM DBA_REGISTERED_ARCHIVED_LOG R, DBA_CAPTURE C
    WHERE R.CONSUMER_NAME = C.CAPTURE_NAME AND
          R.NEXT_SCN >= C.REQUIRED_CHECKPOINT_SCN;
```

Monitor Streams for Apply Errors

The Streams environment includes an error queue. Any errors encountered when applying logical change record (LCR) messages to the Banner ODS database are placed in this queue. Use the following steps to monitor the errors in the queue.

1. Query the DBA_APPLY_ERROR data dictionary view to see any errors in the error error queue.

   Alternately, you can run the Staging Area Status process from the Banner ODS or Banner EDW Administrative UI, select to display apply errors, and then view the process control report.

   1.1. Click **Staging** from the Administrative UI menu.

   1.2. Click **Report Staging Area Status**.

1.3. Check the **Display Apply Errors?** process parameter and any other parameters that you want to display in the status report.

1.4. Enter *NOW* in the a **Run Date** and **Runtime** fields.

1.5. Click **Submit**.

1.6. Click **View Control Reports**.

2. Issue the following commands to see detailed information about error messages in the destination:

```
SQL> SET SERVEROUTPUT ON
   SQL> EXEC MGKSTRE.P_PRINT_ERRORS;
```

3. Fix the errors.

4. Reapply or delete the messages in the error queue.

- Issue the following command to reapply all error messages for the “apply1” process in the error queue:

```
BEGIN
   DBMS_APPLY_ADM.EXECUTE_ALL_ERRORS(
      apply_name      => ‘apply1’,
      execute_as_user => false);
END;
/
```

- Issue the following command to delete all error messages for the “apply1” process in the error queue:

```
BEGIN
   DBMS_APPLY_ADM.DELETE_ALL_ERRORS(
      apply_name  => 'apply1');
END;
/
```

Monitor Source Capture Queue for Growth

If captured messages don’t get propagated to the destination database, the number of messages will grow in the capture queue. You need to periodically monitor the capture queue to determine how full it is.

Use the following query to display the number of messages in memory and spilled to disk for each queue.

```
SELECT QUEUE_SCHEMA "Schema",
      QUEUE_NAME "Queue Name",
      (NUM_MSGS - SPILL_MSGS) "Messages in Memory",
      SPILL_MSGS "Messages Spilled",
      NUM_MSGS "Total Messages"
   FROM V$BUFFERED_QUEUES;
```

Monitor the Status of Source Propagation Jobs

If the Streams propagation schedule becomes disabled, the capture queue on the source Banner database may fill up. Perform the following steps to monitor the propagation jobs.

1. Issue the following query to display the status of each propagation job.

```
SELECT DISTINCT P.PROPAGATION_NAME "Propagation",
      P.SOURCE_QUEUE_NAME "Source Queue",
      P.DESTINATION_QUEUE_NAME "Destination Queue",
      S.LATENCY "Latency",
      DECODE(S.SCHEDULE_DISABLED,
               'Y', 'DISABLED',
               'N', 'ENABLED') "Propagation Status",
      S.PROCESS_NAME "Process Name",
      S.FAILURES "Failures"
   FROM DBA_QUEUE_SCHEDULES S, DBA_PROPAGATION P
   WHERE P.DESTINATION_DBLINK = S.DESTINATION
   AND S.SCHEMA = P.SOURCE_QUEUE_OWNER
   AND S.QNAME = P.SOURCE_QUEUE_NAME;
```

2. Determine whether any propagation jobs are disabled.

3. Stop and restart any disabled propagation jobs using the steps provided in the “Start or Stop the Streams Propagation Schedule” section.

Set the Checkpoint Frequency and Retention Time (optional)

The Oracle Streams capture process takes a checkpoint periodically and stores it in the Banner SYSAUX tablespace. Checkpoints record information on scanned system change numbers (SCN) and allow for a quicker restart of the capture process. By default, a checkpoint is taken every 10MB of scanned redo.

The required checkpoint SCN is associated with the lowest SCN at which the capture process requires redo data. The checkpoint retention time defines how long to keep checkpoints prior to the required checkpoint SCN. The default retention time is 60 days.

The first SCN for a capture process defines the lowest SCN at which the capture process can begin capturing changes. Once a checkpoint is purged, the first SCN is updated to match the next stored checkpoint. You may need to modify these values based on the amount of redo generated and size of the SYSAUX tablespace in your Banner database.

Run the following in the Banner database to modify the checkpoint frequency and retention time:

```
BEGIN
   DBMS_CAPTURE_ADMIN.SET_PARAMETER(
      capture_name => capture process name,
      parameter    => '_checkpoint_frequency',
      value        => size of scanned redo in MB);
END;
/

BEGIN
   DBMS_CAPTURE_ADMIN.ALTER_CAPTURE(
      capture_name              => capture process name,
      checkpoint_retention_time => number of days);
END;
/
```

Refer to Oracle Streams Concepts and Administration Guide for more information about capture process checkpoints.

Avoid NOLOGGING and UNRECOVERABLE Keywords in Source

Avoid using the keywords `NOLOGGING` and `UNRECOVERABLE` in the Banner source code. If you use these keywords in the source code, related source changes are not written to the redo logs. Because the Streams capture process mines the redo logs for change information, any changes using these keywords will be lost. Streams will not be able to replicate these changes in the Banner ODS stage tables. As a result, the source Banner and destination Banner ODS tables will not be synchronized leading to future Streams errors.

Use Streams Tags with Batch Processes

All changes made to the Banner database are written to the redo log in the Banner system. By default, the Banner ODS Streams environment is set up to capture only changes in the redo log that do not include a tag. You can set a Streams tag on specified log entries, which basically includes an additional column with each entry in the redo log. A Streams capture process can then examine the extra column for each tagged entry in the redo log.

You can set a Streams tag with a non-NULL value before running a large batch file in a Banner session. Any inserts, updates, or deletes made during that tagged session will not be caught by the capture process and the changes will not be replicated to the Banner ODS.

Be aware that after the batch process has completed some destination tables in the Banner ODS may no longer be synchronized with the source Banner tables. When this happens, the affected destination tables need to be dropped from Streams and re-added after running the batch process.

Issue the following command to set the Streams tag to a value of “00” for a specific Banner session:

```
BEGIN
   DBMS_STREAMS.SET_TAG(TAG => HEXTORAW('00'));
END;
/
```

Use Data Dictionary Views to Display Streams Information

Refer to the “Monitoring a Streams Environment” chapter of Oracle's *Streams Concepts and Administration Guide* for a complete list of the static data dictionary views and dynamic performance views related to Streams.

Maintain Materialized Views framework

Keeping the target database materialized views synchronized with the source database tables is the key to maintaining the Materialized Views framework. You will need to refresh data in the materialized views on a regular basis (timing to be determined by your institution) so that, in turn, the staging tables are up to date with data before running the ETL processes to load or reload the warehouse. In addition, upgrades to source products may modify table definitions in the source database. When this happens, it will be necessary to restage the affected materialized views in the target database so they match the source tables.

To ensure that the materialized views are refreshed before data is loaded into the warehouse, materialized views refresh processing has been added to run at the beginning of the “Refresh” ETL jobs that refresh the warehouse data. For example, the REFRESH_STUDENT job in the Banner ODS, which refreshes all of the MST composite tables, now includes a job at the beginning to refresh the associated SATURN (student-related) materialized views prior to refreshing the composite tables. This addition to the Refresh jobs ensures that all materialized views related to the data being updated in the warehouse will get refreshed before the ETL loads new data into the warehouse.

There are staging refresh jobs that refresh only the stage tables and materialized views. You can use these jobs to synchronize data on a more frequent basis. This will reduce the amount of work that the integrated materialized views refresh has to do as part of the nightly Banner ODS refresh process.

Refresh groups

To help you manage the process of refreshing materialized views, Oracle includes a component called a Refresh Group, which lets you group together materialized views. Once materialized views are grouped in a Refresh Group, you gain the ability to refresh all of the materialized views within a Refresh Group at the same point in time.

Banner ODS uses Refresh Groups to make it easier to manage related groups of materialized views. When delivered, baseline Banner ODS includes defined Refresh Groups that group together sets of materialized views by schema and table type, for example, grouping validation versus non-validation tables. For each schema staged in Banner ODS, two refresh groups are associated with that schema:

```
ODS_REFGROUP_<schema>ODS_REFGROUP_<schema>_VAL
```

For example, there is a Student Refresh Group that includes all Student related materialized views, and a Student Validation Refresh Group that includes just the Validation tables (STV*) from the Student product

By default, any table staged using the APIs, mgkmview.P_stage_mview(), will add that materialized view to one of the two refresh groups based on the Banner table naming standard. If a table has a "V" character in the 3rd position of the table name, that table is included in the validation refresh group.

This mechanism is used to group all source tables together by Subject Area when delivered. You can reorganize these groupings using command-line API calls as delivered in baseline scripts. Also, the Oracle DBMS_REFRESH package provides APIs to create or delete Refresh Groups, and to associate materialized views with them that can be used to reorganize refresh processing.

Staging Refresh Collections

A materialized view can only exist in a single Refresh Group at a time. However, you can include a materialized view in more than one functional area at a time. The STAGING REFRESH COLLECTION parameter stored in the MTVPARM table lets you create a “group of refresh groups”, which is called a “collection” in BPRA. You define a collection by creating multiple MTVPARM records that link multiple Refresh Groups to one ODS Refresh Job.

STAGING REFRESH COLLECTION entries are delivered for all baseline schemas. The delivered records associate all the Refresh Groups with both the subject area specific Refresh jobs (for example, STUDENT, GENERAL, ALUMNI) as well as the REFRESH_ALL job and the REFRESH_VALIDATION jobs, so that by default the materialized views will get refreshed as part of the nightly Banner ODS Refresh jobs.

Associate Staging Collections with Refresh Jobs

You associate a staging collection with the actual Banner ODS Refresh jobs using the ETL PACKAGE parameter as a “PRE” step, which means it runs before any mappings or slotted packages. The ETL PACKAGE parameter records link a collection of Refresh Groups to the Banner ODS Refresh job.

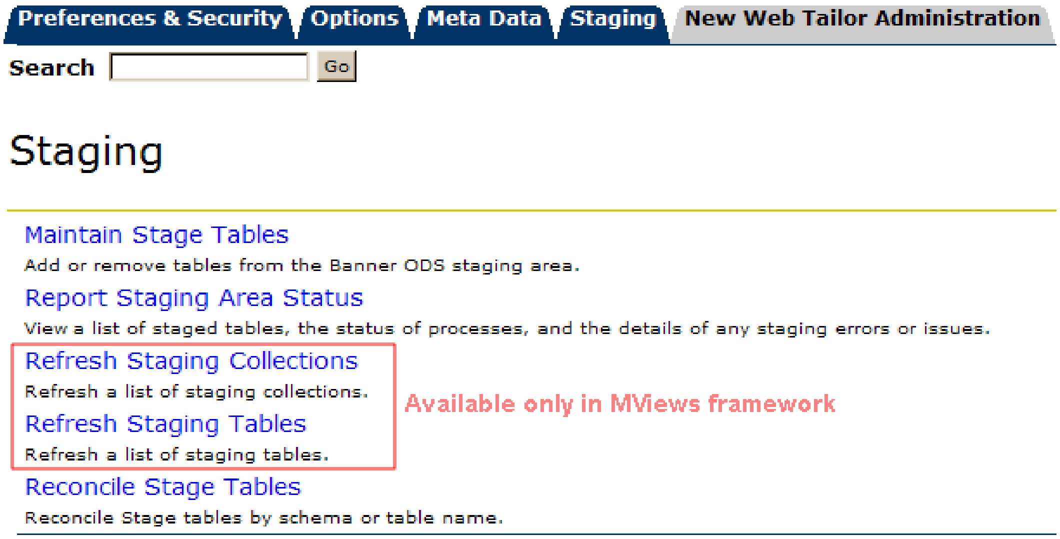
Refresh materialized views

When delivered, the system is set up to refresh the materialized views whenever you run any of the ETL Refresh jobs. It is recommended that you run the ETL Refresh jobs at least once a day typically during the nightly build. You will probably want to refresh the materialized views more often than once a day.

You have two options for refreshing the materialized views outside of the ETL Refresh jobs:

- Refresh a collection of materialized views (staging collection)
- Refresh selected materialized views (staging tables)

These two refresh options are jobs that you run from the Staging menu in the Administrative UI. They are the option that are only present on the Staging menu if you implement the Materialized Views framework.



Staging

Maintain Stage Tables
Add or remove tables from the Banner ODS staging area.

Report Staging Area Status
View a list of staged tables, the status of processes, and the details of any staging errors or issues.

Refresh Staging Collections
Refresh a list of staging collections.

Refresh Staging Tables
Refresh a list of staging tables.

Available only in MViews framework

Reconcile Stage Tables
Reconcile Stage tables by schema or table name.

Refresh a Staging Collection

A “collection” of materialized views is a group of Refresh Groups that have been associated together in the STAGING REFRESH COLLECTION parameter.

The “Refresh Staging Collections” job allows you to select one or more Collections to refresh. Use the following steps to refresh a collection of tables.

1. Click **Staging** from the Administrative UI menu.

2. Click **Refresh Staging Collections**.

3. Choose the **Staging Groups to Refresh**. Use Shift-click to select a contiguous range of tables or Ctrl-click to select noncontiguous tables.

4. Choose the **Logging Mode** which defines the level of detail to include in the control report.

   - Display summary stats by Refresh Group - displays summary of information for each Refresh Group selected for staging (default selection)

   - Display detail about each MView Refresh - displays information about each materialized view in all Refresh groups selected for staging

5. Enter a **Run Date** and **Run Time** to schedule when to run the job that will remove the stage tables from the Banner ODS. Enter NOW in each field to run the job immediately.

6. Click **Submit** to schedule the job to run.

Refresh selected materialized views

The “Refresh Staging Tables” job allows you to select one or more materialized views to refresh. Use the following steps to refresh a collection of tables.

1. Click **Staging** from the Administrative UI menu.

2. Click **Refresh Staging Tables**.

3. Choose the **Staging Tables to Refresh**. Use Shift-click to select a contiguous range of tables or Ctrl-click to select noncontiguous tables.

   The list of materialized views includes the Refresh Group name in parentheses to help organize the listing and allow easier selection of a group of materialized views related by Refresh Group.

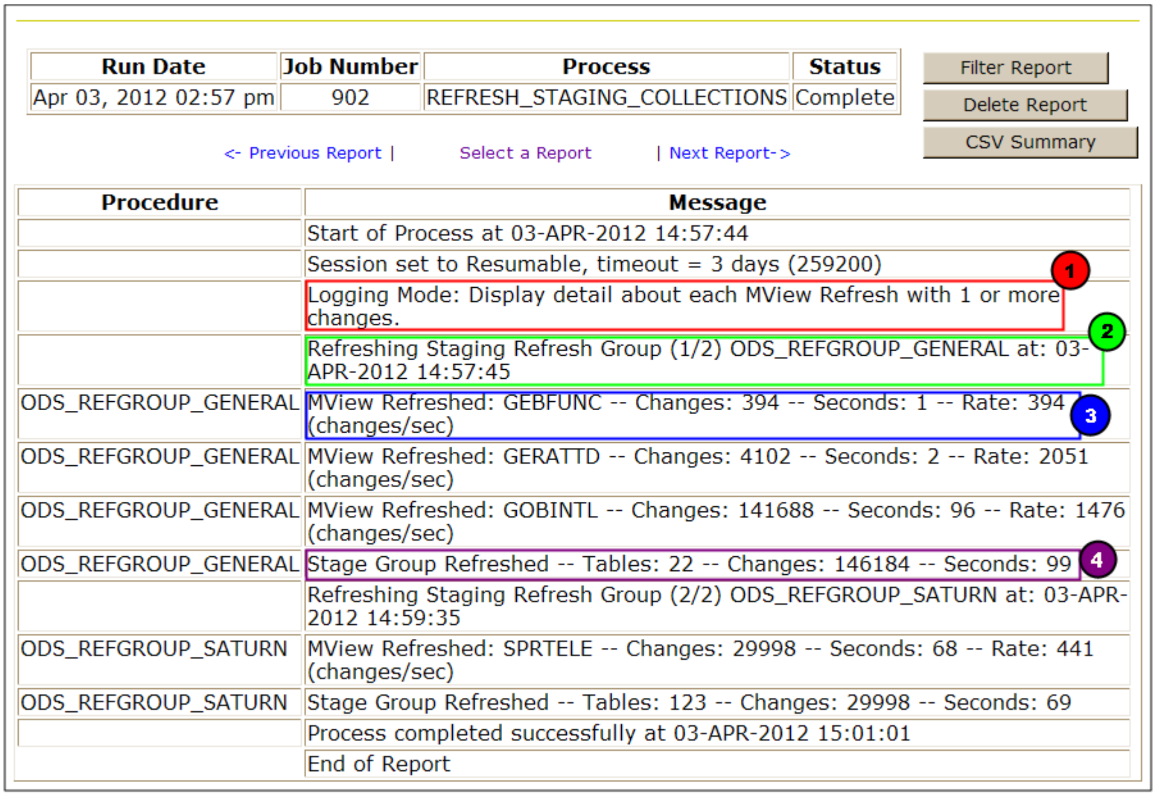
4. Enter a **Run Date** and **Run Time** to schedule when to run the job that will remove the stage tables from the Banner ODS. Enter NOW in each field to run the job immediately.

5. Click **Submit** to schedule the job to run.

Refresh Staging Collection job control reports

When you submit a job to Refresh a Staging Collection or Refresh Staging Tables, each job generates a standard control report listing details about the Refresh Groups or materialized views processed by the job. In addition, because the standard ETL Refresh jobs also refresh Staging Collections at the beginning of each job, similar output is displayed in the ETL Refresh job control report.

The following figure illustrates the control report for the Refresh Staging Collections job.



Run Date	Job Number	Process	Status
Apr 03, 2012 02:57 pm	902	REFRESH_STAGING_COLLECTIONS	Complete

Procedure	Message
	Start of Process at 03-APR-2012 14:57:44
	Session set to Resumable, timeout = 3 days (259200)
	Logging Mode: Display detail about each MView Refresh with 1 or more changes.
	Refreshing Staging Refresh Group (1/2) ODS_REFGROUP_GENERAL at: 03-APR-2012 14:57:45
ODS_REFGROUP_GENERAL	MView Refreshed: GEBFUNC -- Changes: 394 -- Seconds: 1 -- Rate: 394 (changes/sec)
ODS_REFGROUP_GENERAL	MView Refreshed: GERATTD -- Changes: 4102 -- Seconds: 2 -- Rate: 2051 (changes/sec)
ODS_REFGROUP_GENERAL	MView Refreshed: GOBINTL -- Changes: 141688 -- Seconds: 96 -- Rate: 1476 (changes/sec)
ODS_REFGROUP_GENERAL	Stage Group Refreshed -- Tables: 22 -- Changes: 146184 -- Seconds: 99
	Refreshing Staging Refresh Group (2/2) ODS_REFGROUP_SATURN at: 03-APR-2012 14:59:35
ODS_REFGROUP_SATURN	MView Refreshed: SPRTELE -- Changes: 29998 -- Seconds: 68 -- Rate: 441 (changes/sec)
ODS_REFGROUP_SATURN	Stage Group Refreshed -- Tables: 123 -- Changes: 29998 -- Seconds: 69
	Process completed successfully at 03-APR-2012 15:01:01
	End of Report

The control report includes the following information identified by the callout numbers in the sample control report above.

1 The report identifies the Logging Mode that was defined when the report ran.

2 Each Refresh Group associated with a selected Refresh Collection is refreshed with the start time noted.

3 Each MView Refreshed in a Refresh Group is identified including the number of Changes, how many Seconds it took to refresh that Materialized view, and the Rate of the refresh for that.

4 The report identifies the overall status of the Stage Group Refreshed including the number of Tables, Changes, and Seconds to complete. For example, in the sample report above the General Refresh Group (ODS_REFGROUP_GENERAL) refreshed 22 tables with 141688 changes in 99 seconds.

If an exception occurs when refreshing a single materialized view within a refresh group, the entire refresh for that group will not be processed. This is illustrated in the following sample error message for the Financial Aid group ODS_REFGROUP_FAISMGR, which could display in a control report.

```
ORA-12048: error encountered while refreshing materialized view
“FAISMGR”.”RPATRM” ORA-04098: trigger
‘FAISMGR.RT_RPRATRM_INSERT_ODS_CHANGE’ is invalid and failed
re-validation
```

When this kind of error occurs, you can do one of the following to address it:

- Fix the underlying problem (in the above case, an invalid trigger on one of the materialized views)
- Use the Refresh Staging Tables option to manually select and refresh all of the materialized views except the one that caused the error. When you use the Refresh Staging Tables job, the materialized views are all listed, grouped by Refresh Group, so that you can easily identify which materialized views belong in which Refresh Group.

Web Tailor Administration

The Enterprise Administrative application uses the Web Tailor application to build its look and feel. Web Tailor delivers customizable global Web rule definitions and procedures, customizable menus, menu items, graphics and text definitions.

From the Administrative Tool, use the Web Tailor Administration menu item to access the Web Tailor options. The tasks under this menu item allow you to customize various aspects of the Administrative Tool. Other sections include references to the various Web Tailor options that you may want to customize. To learn more about Web Tailor, refer to the *Web Tailor User Guide*.

Functions

Web Tailor lets you build the look, feel, and unique personality of all your institution’s web applications, so you can personalize your institution’s interface to the world. Web Tailor delivers customizable global web rule definitions and procedures, customizable menus, menu items, graphics and text definitions.

The following Web Tailor functions are available from the Web Tailor Administration Menu.

- “Customize a Web Menu or Procedure”
- “Customize a Graphic Element”
- “Customize a Set of Information Text”
- “Customize a Set of Menu Items”
- “Update User Roles”
- “Customize a Web Module”
- “Customize Web Rules”
- “Customize Web Tailor Parameters”
- “Customize a Login Return Location”
- “Customize Web Tailor Overrides”
- “Customize Global User Interface Settings”

Customize a Web Menu or Procedure

The **Customize a Web Menu or Procedure** option allows you to define the menus that will appear on your institution’s web pages for the different self-service applications, and to specify the procedures behind them.

Refer to the Web Tailor Online Help for more information about creating or customizing a web menu or procedure.

Customize a Graphic Element

The **Customize a Graphic Element** option allows you to specify the images that will be available for use on your web pages. For each image, you can specify its name, the directory where it is located, its height, its width, and various other aspects.

Refer to the Web Tailor Online Help for more information about creating or customizing a graphic element.

Customize a Set of Information Text

The **Customize a Set of Information Text** option allows you to add or customize Information Text (Info Text). Info Text can be:

- Instructions on how to use a page
- Help for the page

- Error messages

Refer to the Web Tailor Online Help for more information about customizing Information Text.

Customize a Set of Menu Items

The **Customize a Set of Menu Items** option allows you to define the items that will appear on the menus on your institution's web pages.

Refer to the Web Tailor Online Help for more information about customizing a set of menu items.

Update User Roles

The **Update User Roles** option allows you to change the roles to which a person has been assigned. The User Roles define a high level of security and allow you to give users access to selected components of the Administrative User Interface. A users assigned roles determine which areas of the Administrative User Interface the user can access and make changes within the Banner ODS or the Banner EDW.

Refer to the "Update User Roles" section for more information about updating user roles.

Customize a Web Module

This function allows you to modify a specific application or module that uses Web Tailor, such as Accounts Receivable, Student Self-Service, or Banner Performance Reporting and Analytics.

Refer to the Web Tailor Online Help for more information about customizing a web module.

Customize Web Rules

This function allows you to define certain rules for your institution's web pages. For example, you can identify the number of minutes a person can be inactive before they are timed out, or specify the format for the date and time information that appears on your pages.

Refer to the Web Tailor Online Help for more information about customizing a web rules.

Customize Web Tailor Parameters

This function allows you to customize parameters used in Web Tailor processing, such as the maximum length of PINs. You must use great care when modifying these parameters.

Refer to the Web Tailor Online Help for more information about customizing a Web Tailor parameter.

Customize a Login Return Location

Use this function to specify the page you would like to be displayed when a user is timed out, then logs back in.

Refer to the Web Tailor Online Help for more information about customizing a login return location.

Customize Web Tailor Overrides

This page allows you to replace certain procedures and functions with your own under certain circumstances. This is necessary because you may have a stand-alone product you would like to use with the self-service products, and you need to use some of the procedures and functions in the other system. If an override is defined, that code will be run instead of the Web Tailor code.

Refer to the Web Tailor Online Help for more information about customizing Web Tailor overrides.

Customize Global User Interface Settings

This function allows you to set up rules that will apply to your institution's web pages as a whole. You can specify:

- Header information
- The location URL of CSS that control the pages' look-and-feel
- The location URL of CSS that control the look-and-feel of your Help text
- The location URL of where your Help text files are stored

Note

It is recommended that you use Info Text as your Help text. ■

- Images that represent errors and warnings
- An image that indicates that a field is required

Refer to the Web Tailor Online Help for more information about customizing global user interface settings.

5 Cognos and BPRA Integration

A critical factor in determining the success of a reporting solution is the existence of a well defined and useful meta data layer. The meta data layer enables you to define relationships between objects in the database. It also enables additional filtering or formatting that can be useful to you when creating reports.

The Cognos Business Intelligence meta data layer is delivered as part of the warehouse. Relationships between the reporting views in the warehouse are included in these meta data layers for the supported reporting tools. Relationships between the dimension and fact tables in the Banner Enterprise Data Warehouse (EDW) are included in these meta data layers.

The meta data layer provides the joins used by the database to connect the views or database tables so that you do not need to define those relationships when creating queries or reports using the reporting tools. You can use any reporting tool with the Banner Performance Reporting & Analytics products; however, you gain added value from using the Cognos meta data layers created and delivered with the products.

The warehouse reporting meta data defines the database column definitions and how to aggregate measures. It also provides a presentation view of the data to facilitate reporting.

In Cognos Business Intelligence, the reporting meta data is defined using the Cognos Business Intelligence Framework Manager (FM) to create FM models.

IBM Cognos Business Intelligence (IBM Cognos BI) software

Ellucian offers IBM Cognos Business Intelligence (IBM Cognos BI) software as part of our solution to address the needs of your institution. IBM Cognos BI solutions let your institution improve and direct performance by helping to enable the key steps in the management cycle – from query and reporting, to analysis, to measuring and monitoring performance with dashboarding and scorecarding.

The Ellucian partnership with IBM Cognos provides you with a superior business intelligence solution, primarily by leveraging the Banner Operational Data Store (Banner ODS) and Banner Enterprise Data Warehouse (Banner EDW) solutions in conjunction with the Cognos Business Intelligence suite. As part of the Banner ODS/Banner EDW solutions we also provided the end-user reporting layer for IBM Cognos. Additionally, for the Banner EDW solution, Ellucian has built the IBM Cognos cubes required to leverage the IBM Cognos OLAP reporting tool.

IBM Cognos provides a unified decision workspace that lets you view, assemble and personalize data quickly, according to your needs and without help from IT. With IBM Cognos, you can:

- Combine data from any source and explore it from any perspective for a complete understanding of outcomes, opportunities, threats and trends.
- Expand your expertise beyond traditional business intelligence capabilities without needing to switch tools or learn new applications.
- Analyze facts and anticipate outcomes simply by shifting perspectives and time horizons.
- Enable business users to increase their personal productivity and independence from IT.

IBM Cognos Business Intelligence Query and Reporting

From professional report authors responsible for designing one-to-many reports for the enterprise, to business users who need the ability to create their own ad hoc queries or customize existing reports, IBM Cognos Business Intelligence Query and Reporting fits the needs of users across the organization. With IBM Cognos Business Intelligence Query and Reporting, users are equipped with the information they need to make fact-based decisions in a system that's simpler, faster and easier to manage.

This includes Report Studio which is a 100% Web-based query and report authoring tools that is adaptive and flexible in supporting a variety of simple and complex, compound report layouts, including multi-page, multi-query, and multi-source reports. Deployment options for content include On-demand as well as bursting, scheduling and event based deployment.

For end-users who want to build single query reports and charts with an easy-to-use tool, the Query Studio tool is provided. Users can construct reports in a WYSIWIG manner, dragging and dropping data elements into lists, cross-tab, or chart style reports. This tool also provides easy to implement, flexible formatting including templates, conditional formatting, different calculation options for groups and footer summaries. Compound filters can be implemented via menu options to enable ad-hoc reports to be easily adapted to new business questions.

Cognos BI query and reporting capabilities include professional report authoring capabilities that are easy to use and minimize the effort for report authors.

- Design and build – Create report templates to include standard report objects, queries, and layouts. WYSIWYG capability enables flexible and intuitive report assembly and core BI capabilities.

- Analyze and share – View, interact with and analyze the result set, and share the results to follow a train of thought and generate a unique perspective around information.
- Assemble and format – Drag-and-drop trusted content (BI, Metric Studio, and PowerPlay), filters and other content (RSS, HTML, text and images).
- Modify and arrange – Add colors and text, add comments and personalize widgets.
- Interact and analyze – Change display, add calculations, filter prompt, drill up/down, sort data and seamlessly move to Business Insight Advanced for additional analysis.
- Share and collaborate – Share and distribute objects and reports to be consumed by others

Collaborative reporting

Cognos BI query and reporting capabilities offer these collaborative reporting features to help business users communicate with others in order to drive decisions and gain additional insight.:

- Comments and annotations – Create and save notes on reports that other users can then reference, making it much easier for users to share information and capture valuable insights.
- Enhanced event management – Assign owners and approval processes for specific actions with the BI workflow.

Author once, consume anywhere

Cognos BI query and reporting capabilities enable you to create a single report that users can access on multiple devices, in multiple formats, and in other applications and processes.

- Import and refresh BI content in MS Office using Cognos for Microsoft Office.
- Embed BI content in third-party applications and portals using Cognos Mashup Services.
- Take reports with you on your iPhone, Blackberry and iPad with Cognos Mobile.
- Consume interactive report output offline or in a disconnected manner with Cognos Active Report.

Single metadata layer for all reporting

Regardless of the variety of reporting your organization performs, all reporting data passes through a single metadata layer. The metadata layer manages translation from the data you need to the information you see. With Cognos BI you can report, analyze, scorecard and monitor your performance with a single view of all data sources.

Dashboarding

IBM Cognos provides dashboard capabilities that allow any user to access, interact and personalize content in a way that supports the way they make decisions through Cognos Connection and Business Insight Dashboards.

Through Cognos Connection, dashboards composed of portlets can be configured by the user to display the charts of their choice, including charts that they create themselves. The user can create multiple dashboard tabs to easily access different dashboard views. Each report has multiple filters that can be configured by the user for their particular focus by selecting the prompts button for the chart. Users can also navigate to report content directly from the portal dashboard to run/schedule particular reports through the report navigation portlet or by searching report content using the search portlet. Ellucian solutions deliver several preconfigured portlet dashboards.

IBM Cognos Business Insight

IBM Cognos Business Insight provides an integrated business intelligence experience for business users. This web-based tool allows you to use IBM Cognos content to build sophisticated interactive dashboards that facilitate collaborative decision-making. These dashboards can include traditional business intelligence assets as well as external data.

IBM Cognos Business Insight enables users to assemble and explore all types of data in any time horizon through a dynamic, highly personalized interface.

IBM Cognos Business Insight helps you see more and do more with your information. It enables users to assemble and explore data in depth and in detail, then collaborate with others easily—all in a highly personalized business intelligence workspace. IBM Cognos Business Insight helps organizations drive alignment and build consensus for more agile and responsive decision-making—delivering smarter decisions for better business results.

Fast, easy access to vital information

- Gain an “at a glance” view of the business in a pre-assembled workspace.
- Shift seamlessly to business authoring or in-depth analysis on the fly, through a graduated user interface.
- Search documents and databases by keyword to find the information you need.
- Integrate external data from spreadsheets, third-party or departmental applications for ad hoc reporting and analysis
- Change chart types, add pre-defined or custom calculations, sort and reorder data and add filters to highlight key information.
- Merge external data with dimensional and relational data sources using simple drag-and-drop tools.

- Use intuitive, menu-driven tools to manipulate data easily, without assistance from IT.

The Business Insight Dashboard tool provides users with a web-based user interface that allows you to open or edit a dashboard or to create a dashboard. This dynamic dashboard interface provides an intuitive and interactive to allow business users to personalize content to fit their needs. Business users can use a free-form layout to add dashboard elements such as reports and charts, images, RSS feeds, textual objects, or slider filters. In addition, they can interact with reports to Change display, add calculations, filter/prompt, drill up/down, and sort data and seamlessly move to Business Insight Advanced (Called Cognos Workspace Advanced) for additional analysis.

Scorecarding

IBM Cognos provides scorecard capabilities that allow you to see how your institution is performing based on defined key performance indicators (KPIs).

At-a-glance information to successfully link strategy to operations

IBM® Cognos® Business Intelligence Scorecarding enables organizations to capture corporate strategy and communicate that strategy at the operational level. It helps executives and managers provide quantifiable goals and targets, and allows them to track performance across business units, operating subsidiaries and geographic regions to quickly identify areas that need attention.

IBM Cognos Business Intelligence Scorecarding automates the strategy management process, enabling organizations to monitor, measure and manage their business metrics against strategic and operational objectives throughout the organization.

Align strategy with operations

- Track performance against key performance indicators (KPIs) to link corporate strategy to operational tactics.
- Set quantifiable goals for any time horizon and use scorecards to monitor progress on specific projects and activities.
- Create strategy maps, impact diagrams, and other elements of the Balanced Scorecard.
- Maintain metrics in a centralized data store to ensure consistent definitions.

Communicate strategy and track your progress

- Visually capture organizational strategy to help departments and employees set priorities.

- Use status indicators and plan-versus-actual data to reveal progress against targets.
- Access business intelligence reports and other content from within the scorecarding environment to see details and context.
- Use menus to define threshold ranges, benchmarks, data source definitions, contact names and URLs.

Ensure accountability for performance

- Assign a primary owner for every metric to ensure greater accountability.
- Organize and view scorecards by status to quickly identify problem areas and focus on high-priority objectives.
- Set alerts to stay aware of changes in the status of metrics.
- Assign actions against metrics to apply corrective actions to poorly performing business areas.

Share with more user communities

- Populate metrics in any language, then display in the user’s preferred language.
- Monitor performance of business units, operating subsidiaries and geographic regions to quickly identify areas that need attention.
- Gain quick access through a single sign-on dialog box.
- Leverage scorecard information in dashboards and portals to access and communicate scorecard content and encourage adoption throughout the organization.

Enjoy simple deployment and administration

- Create metrics, process diagrams and scorecards once and then share them across the organization.
- Integrate cross-functional data from virtually any source.
- Use intuitive wizards to guide you through the design process and reduce the time spent in scorecard maintenance and updates.
- Benefit from a zero-footprint, Web-based deployment model, which reduces the administrative burden while encouraging user adoption.

IBM Cognos Mobile

IBM Cognos Mobile provides timely and convenient access to IBM Cognos Business Intelligence information from a mobile device. This provides decision makers with access to business-critical information wherever they are and whenever they need it.

Framework Manager models

Databases are typically designed to store data captured through business processes. The stored data is not easily accessible for reporting and analysis to make enterprise decisions in business terms. Because of this, data requires metadata, the ‘data about data’, so that it can be effectively retrieved for analysis and reporting. The Cognos Framework Manager (FM) tool allows you to redefine the data in the database to answer business questions.

Cognos is designed to deliver centralized metadata via the FM model. The model provides a common definition of data in business terms that add value across the organization. The database is redefined so that you can publish metadata in a package and make it available through the Cognos Connection to the Cognos BI reporting tools Report Studio, Query Studio, and Analysis Studio to answer business questions.

The Framework Manager model presents the data using business terms and definitions. This enables you to use, build, and modify your own reports and enables consistent understanding and use of data and metrics across your institution. The logical relationships between data are defined within the model to enable complete data integration so that you spend less time gathering and organizing data.

For more information about data modeling, see the “Framework Manager User Guide” or “Metric Designer User Guide”. You can find them by searching the IBM Cognos Business Intelligence information center.

Framework Durable models

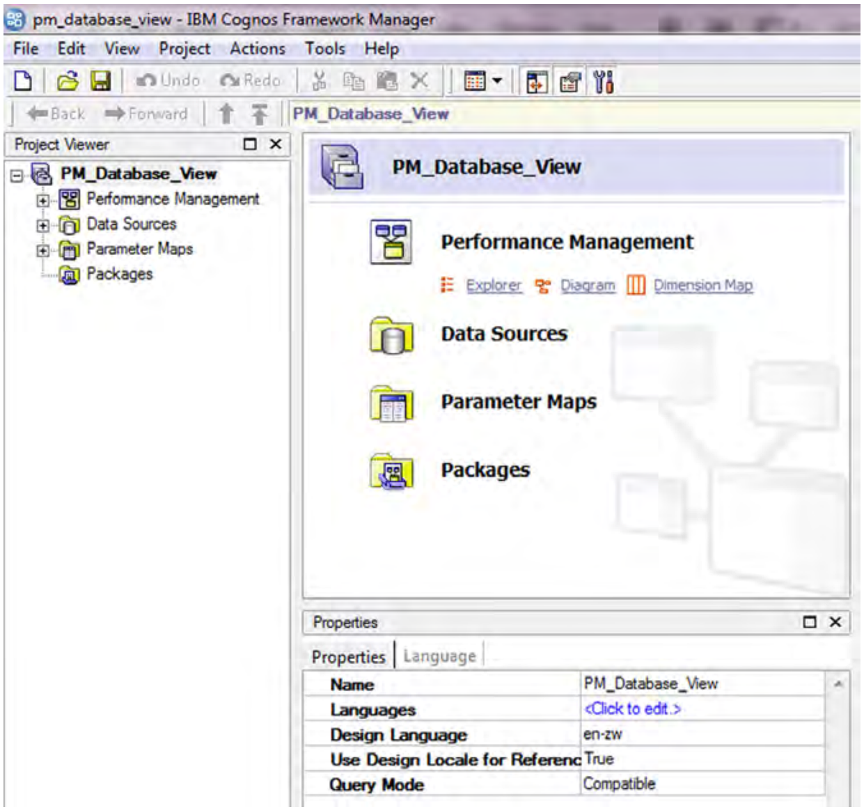
The models delivered with your Performance product are Durable Models. A durable model enables you to make necessary changes to the published packages and corresponding content in the Content Store of the Framework Manager model without impacting existing reports, report authors, and end users. This section describes how to make your customized FM models Durable.

Note

Make these changes to pm_database_view.cpf first, as this is the segmented model for all PM specific models.

1. Add the design language if the model does not already have it.

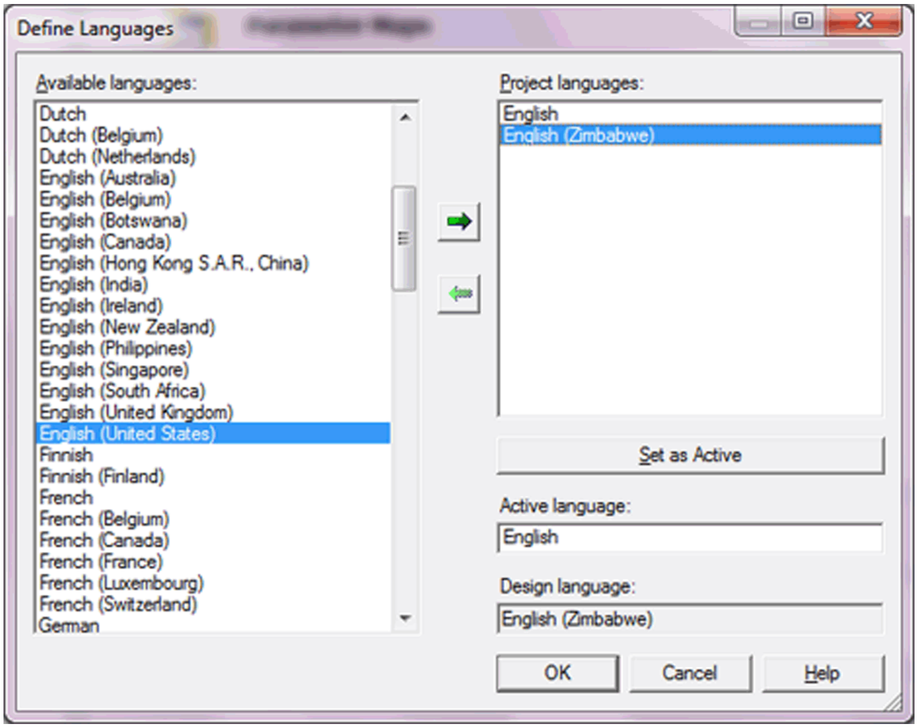
   1.1. Select <Click to edit> in the 'Languages' column under the 'Properties' tab to open the 'Define Languages' popup.



Name	PM_Database_View
Languages	<Click to edit.>
Design Language	en-zw
Use Design Locale for Referenc	True
Query Mode	Compatible

   1.2. In the 'Define Languages' popup, select 'English (Zimbabwe)' as the design language, and click on the green right arrow to add it to 'Project languages'.

   1.3. Click **OK**.



Available languages:

Dutch
Dutch (Belgium)
Dutch (Netherlands)
English (Australia)
English (Belgium)
English (Botswana)
English (Canada)
English (Hong Kong S.A.R., China)
English (India)
English (Ireland)
English (New Zealand)
English (Philippines)
English (Singapore)
English (South Africa)
English (United Kingdom)
English (United States)
Finnish
Finnish (Finland)
French
French (Belgium)
French (Canada)
French (France)
French (Luxembourg)
French (Switzerland)
German

Project languages:

English
English (Zimbabwe)

Active language: English

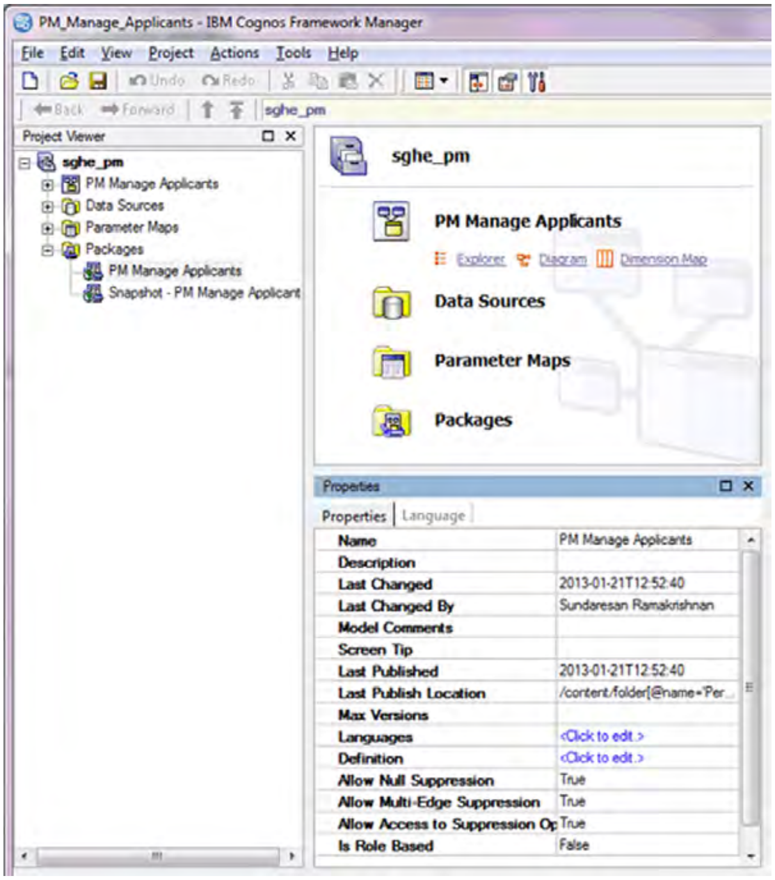
Design language: English (Zimbabwe)

1.4. Click **OK** in the resulting ‘pm_database_view – IBM Cognos Framework Manager’ popup window.

2. Make these changes to the languages of all the packages based on the model as shown in the following screen shots. Save and close the model.

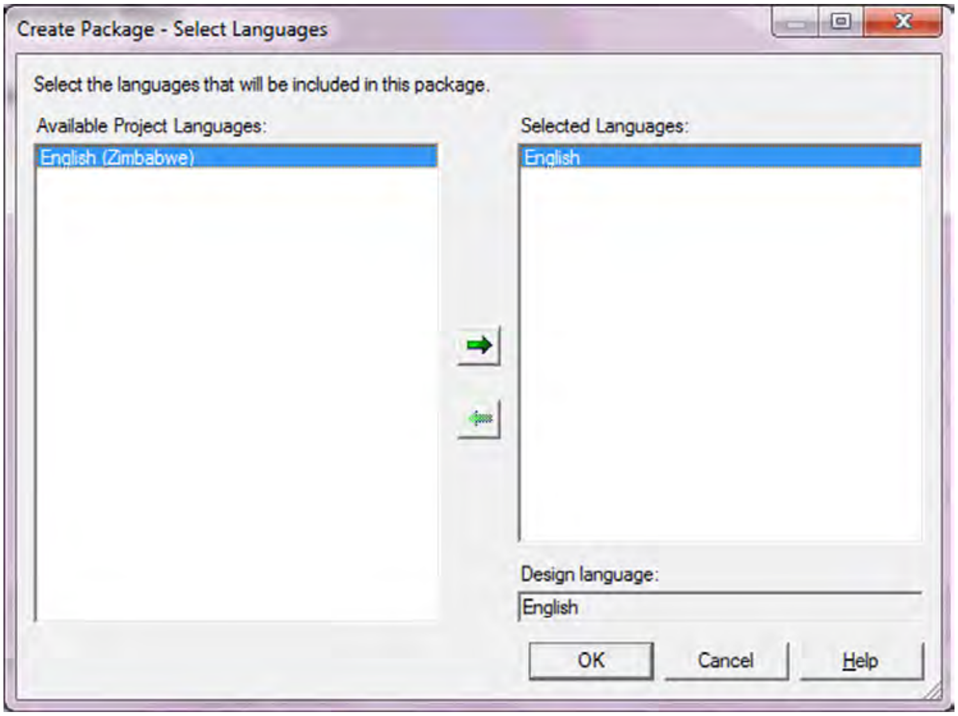
Note

This is applicable only for PM specific models and not pm_database_view.cpf.



Name	PM Manage Applicants
Description	
Last Changed	2013-01-21T12:52:40
Last Changed By	Sundaresan Ramakrishnan
Model Comments	
Screen Tip	
Last Published	2013-01-21T12:52:40
Last Publish Location	/content/folder[@name='Per...
Max Versions	
Languages	<Click to edit.>
Definition	<Click to edit.>
Allow Null Suppression	True
Allow Multi-Edge Suppression	True
Allow Access to Suppression Op	True
Is Role Based	False

3. Select English (Zimbabwe) from the left pane of the ‘Create Package - Select Languages’.


Create Package - Select Languages

Select the languages that will be included in this package.

Available Project Languages:

English (Zimbabwe)

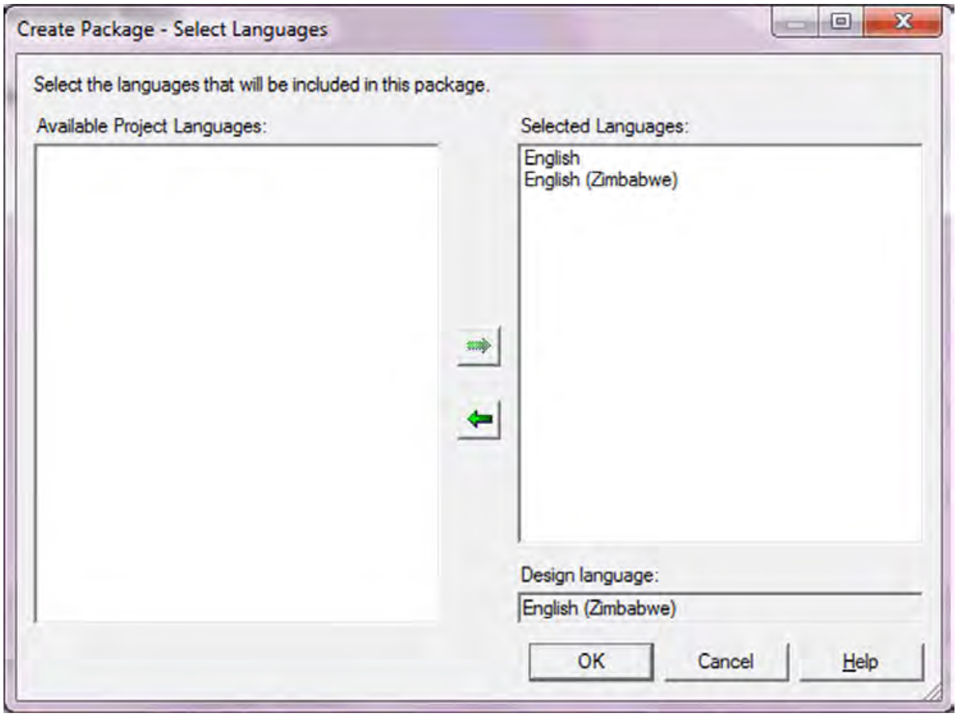
Selected Languages:

English

Design language:

English

4. Click the green right-facing arrow to add ‘English (Zimbabwe) to the list of Selected Languages.


Create Package - Select Languages

Select the languages that will be included in this package.

Available Project Languages:

Selected Languages:

English
English (Zimbabwe)

Design language:

English (Zimbabwe)

5. Using an XML or text editor, open the model.xml file and near the top of the file, change the language value from ‘en’ to ‘en-zw’ as shown here:

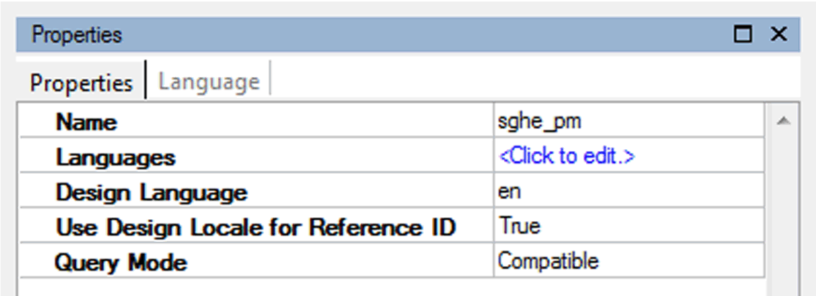
Original code:

```
<defaultLocale>en</defaultLocale>
```

Change language as shown:

```
<defaultLocale>en-zw</defaultLocale>
```

6. Under project level ‘Properties’, set the value of ‘Use Design Locale for Reference ID’ to **True**.



Name	sghe_pm
Languages	<Click to edit.>
Design Language	en
Use Design Locale for Reference ID	True
Query Mode	Compatible

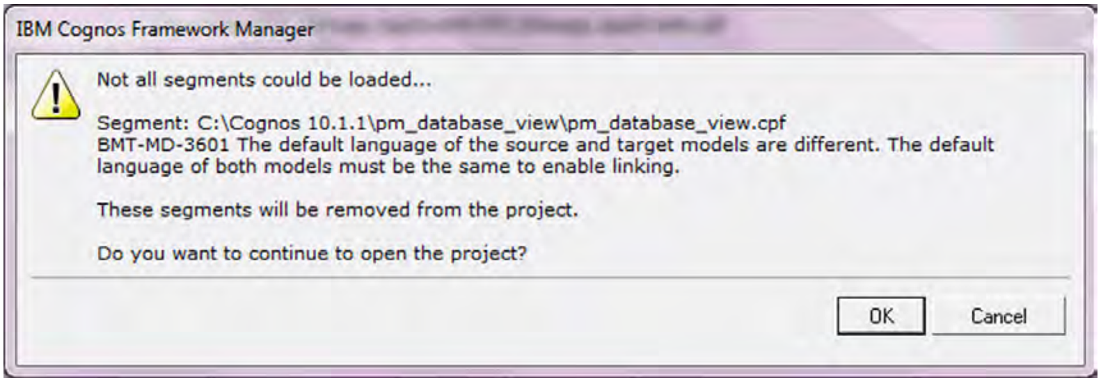
Once a model is made durable, packages based on the model will need to be re-published and existing content checked for durability. The models are segmented and have links from business view(s) to database view. After you make the pm_database_view.cpf as a durable model, follow the steps given below to convert the packages to the Durable model:

1. Open PM specific model.

Note

The images shown in these steps are of the PM_MANAGE_APPLICANTS.cpf file

2. Select **OK** to any warnings and repeat steps 1 to 4 from above for all the models.



Not all segments could be loaded...

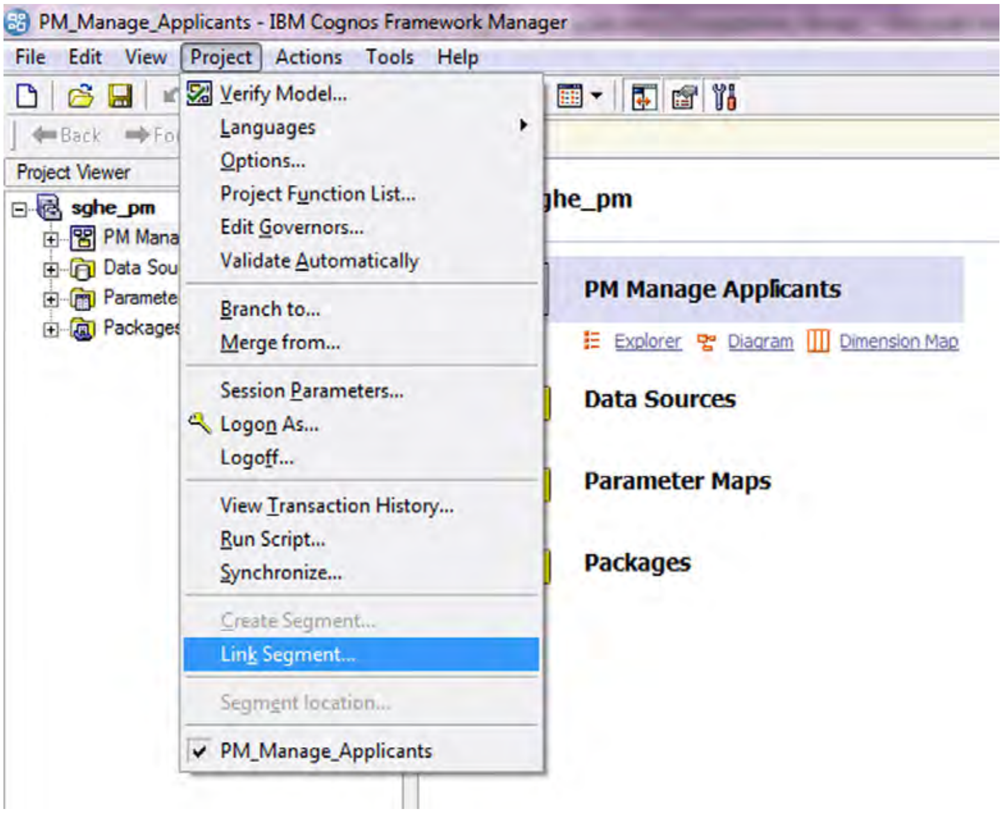
Segment: C:\Cognos 10.1.1\pm_database_view\pm_database_view.cpf
BMT-MD-3601 The default language of the source and target models are different. The default language of both models must be the same to enable linking.

These segments will be removed from the project.

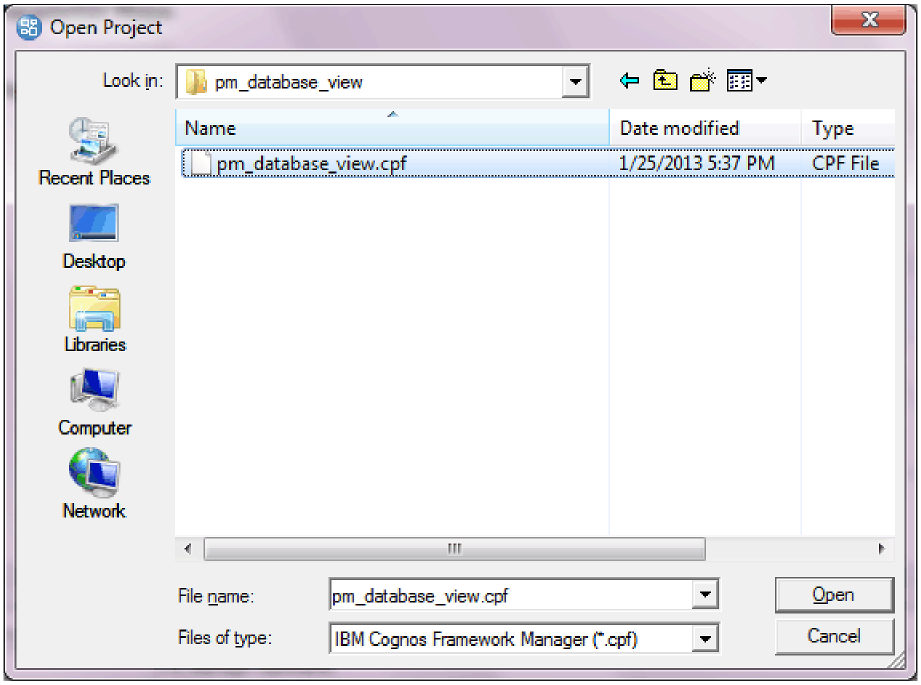
Do you want to continue to open the project?

3. Add the segmented model (link pm_database_view.cpf model) to all the PM specific models as shown below:

   3.1. Select **Project** > **Link Segment** from the main menu options.



3.2. Browse to the location of `pm_database_view.cpf` file. Select and open the file.



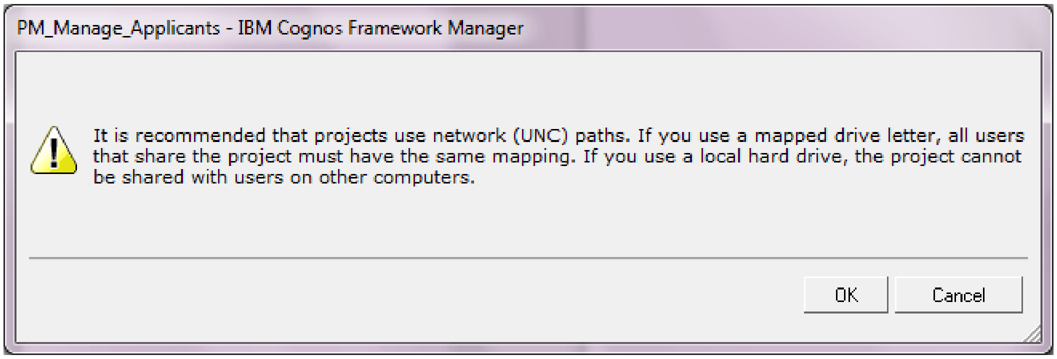
Look in: pm_database_view

Name	Date modified	Type
pm_database_view.cpf	1/25/2013 5:37 PM	CPF File

File name: pm_database_view.cpf

Files of type: IBM Cognos Framework Manager (*.cpf)

3.3. Click **OK** when prompted to use network (UNC) paths.

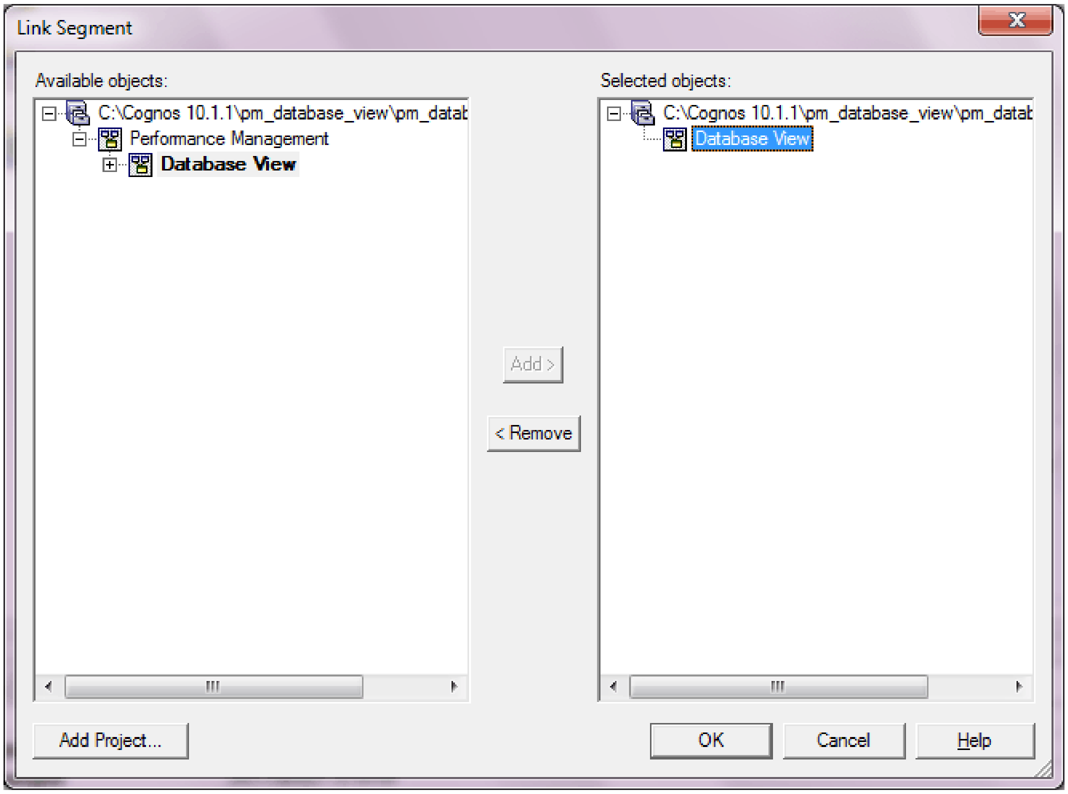


PM_Manage_Applicants - IBM Cognos Framework Manager

It is recommended that projects use network (UNC) paths. If you use a mapped drive letter, all users that share the project must have the same mapping. If you use a local hard drive, the project cannot be shared with users on other computers.

3.4. In the ‘Link Segment’ window, select **Database View**.

3.5. Click the **Add** > button to include Database View in the ‘Selected objects’ list.



3.6. Click **OK**.

Now your Framework Manager model is a durable model.

Metadata layers

Cognos Framework Manager provides the ability to layer metadata as a means to insulate end users from changes made to the underlying data sources and the defined data relationships within the database. When changes to an existing model are required, Framework Manager can identify the impact to existing reports. This enables your institution to manage model changes without having to rewrite reports.

The delivered FM models use two layers to manage the metadata content: the database view and the business view. A third layer, the presentation layer, is used to publish the data in logical groupings.

Database view metadata layer

The database view metadata layer is the layer into which Framework Manager imports all database objects.

There is very little difference between the database view and the database itself. The only differences are the following:

- Object names for columns that are eventually published from subsequent layers include a business name using mixed-case nomenclature and no underscores.
- Some calculated columns make commonly used functions more readily available. They generate a unique key for specific fact tables, and provide the flexibility to configure institution-specific descriptions for certain concepts using parameter maps.

Business view metadata layer

The business view metadata layer organizes content around a specific business process or processes. The business view layer references objects from the database view and relationships among them are defined to support the associated business process.

The content defined within the business view is as follows:

- The warehouse business concepts define the relationships between the fact and dimension tables. These joins define the SQL generated behind the scenes by the various Cognos BI Reporting Tools.
- Determinants are defined for the various dimensions to ensure that the proper cardinality is preserved with multi-fact queries that have a conformed, or shared, dimension.
- Role-based, or ‘alias’, query subjects are defined for those objects that serve multiple business purposes. An example of one such object would be ‘Application Date’ which is a copy of the ‘Calendar Date’ query subject. These role-based query subjects allow an object to be utilized multiple times within the same query for different purposes.

Presentation view metadata layer

The presentation view metadata layer is the layer in which information is reorganized into useful logical groups of data that you can use together for reporting. The query subjects in the presentation layer include data elements and folders of data elements that present the data in an intuitive fashion so it is easy for you to locate desired data for any report.

The following standards were applied when creating the presentation layer:

- Related data or query items are grouped in the same query subject.
- Subsets of data that are typically used together are organized into folders.

- Commonly used filters are defined to enhance functionality. Examples of delivered filters include Student Level Undergraduate, Student Level Graduate, and Student Level Professional.
- Commonly used calculations have been added to make reporting easier.
- Additional range and aging concepts have been added that work in conjunction with parameter maps. Each has an accompanying ‘order’ concept to ensure they appear in proper order when you use them.

From the presentation layer you can publish a complete package of all the data in that presentation view and or a number of smaller packages of information that target specific types of analysis and users. These packages allow you to create and use dashboards, run reports, build ad hoc reports, and analyze trends without the need to sift through large amounts of unneeded information.

Cognos and BPRA Meta Data Integration

Banner ODS includes Cognos Framework Manager (FM) models and packages for the ODS Reporting Views organized in groupings called business concepts. EDW includes Cognos Framework Manager models and packages, and Transformer cubes.

The Performance products - Banner Recruiting and Admissions Performance, Banner Student Retention Performance, and Advancement Analytics for Cognos - include Cognos Framework Manager models and packages and Transformer cubes. The products also include a variety of reports, performance charts, dashboards, and scorecard samples and the supporting Metric Studio/Metric Manager content using the Cognos BI tools.

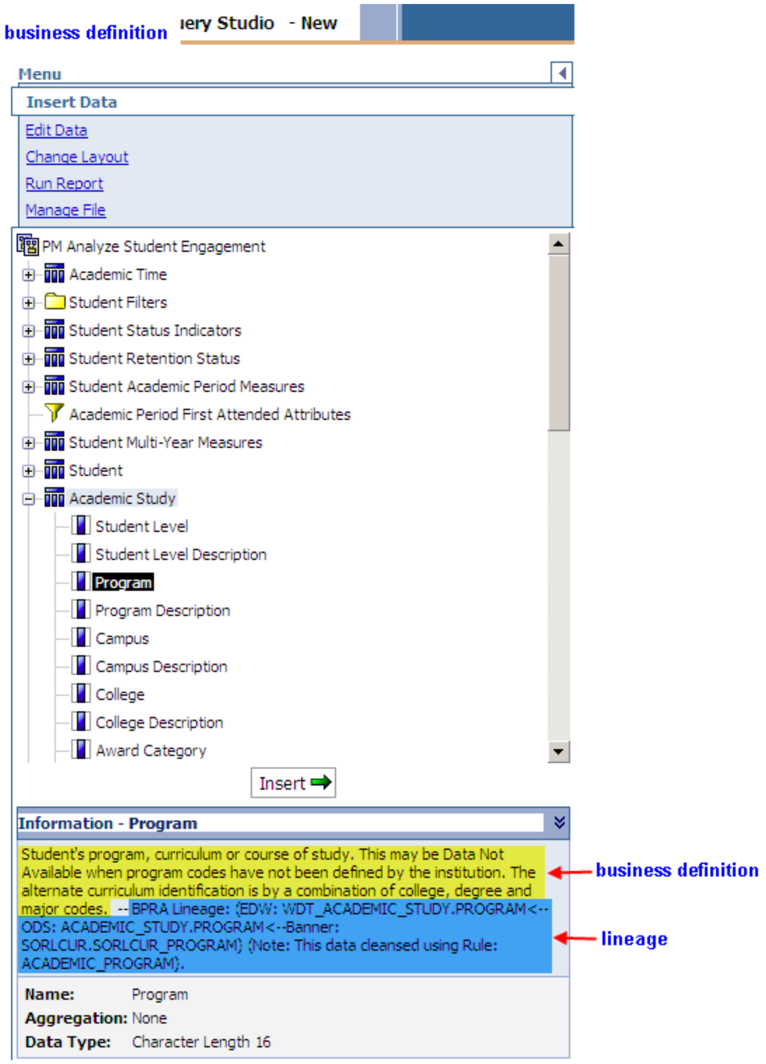
The layers of Cognos content relate to the underlying warehouse data structures that include the reporting views and the EDW fact and dimension tables. The bulk of the data dictionary that describes these data structures is defined as BPRA Meta Data that is stored in the “IA-Admin Meta Data”, (IAMD) and is delivered with each BPRA product.

The BPRA Meta Data includes column level “business definitions” that describe the data and is stored for each target column along with its source system, table, and column. Another key part of the BPRA Meta Data is the mapping from one layer to the next. For a given EDW column, which Banner ODS column it comes from, and in turn, which Banner (source) column that ODS column is sourced from. (This mapping information is hereafter referred to as “lineage”.)

To provide a meaningful relationship between the BPRA Meta Data information and the Cognos reporting tools, the BPRA Meta Data is integrated with the Cognos tools. The Meta Data business definition and lineage information are delivered in the FM models and packages and displayed in the Cognos reporting tools (Query Studio, Report Studio, and Analysis Studio). Within each query item in the FM models, the Description field includes the business definition and lineage, while the ScreenTip field includes the EDW source column name.

View BPRA Meta Data in Cognos

To view the business definition and lineage for a query item, use the arrows at the bottom of the navigation window to open the information for the selected query item. See examples of BPRA Meta Data displayed in Query Studio and Report Studio as illustrated in the following figures.

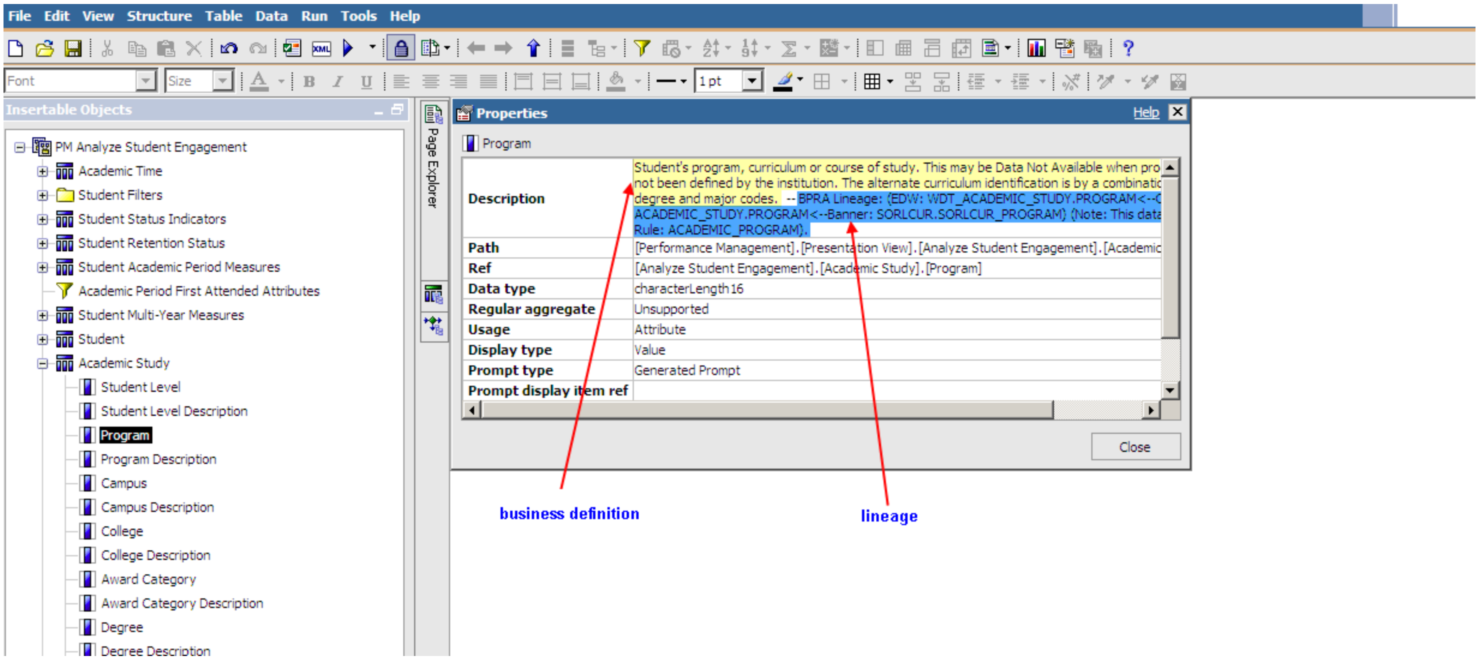


Information - Program

Student's program, curriculum or course of study. This may be Data Not Available when program codes have not been defined by the institution. The alternate curriculum identification is by a combination of college, degree and major codes. --BPRA Lineage: (EDW: WDT_ACADEMIC_STUDY.PROGRAM<--ODS: ACADEMIC_STUDY.PROGRAM<--Banner: SORLCUR.SORLCUR_PROGRAM) (Note: This data cleansed using Rule: ACADEMIC_PROGRAM).

Name: Program
Aggregation: None
Data Type: Character Length 16

Figure 23: View BPRA Meta Data in Query Studio



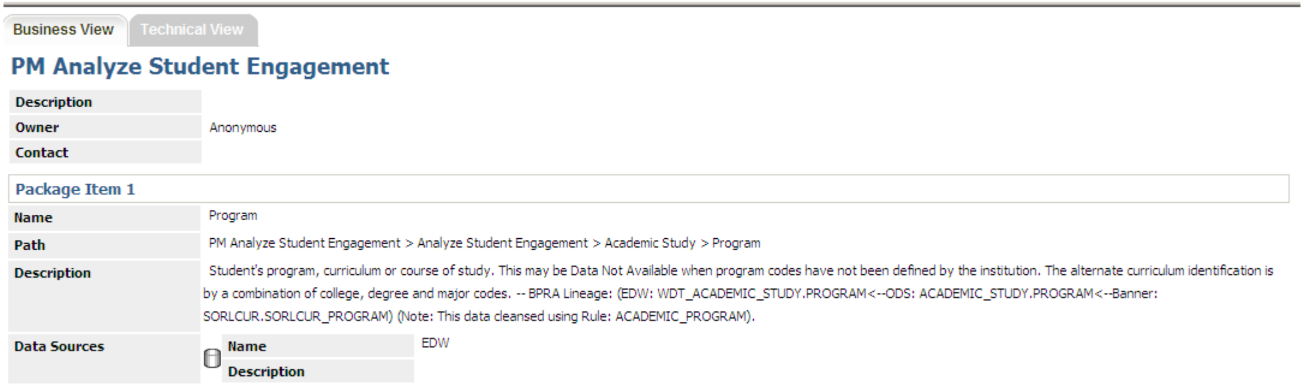
Program

Description	Student's program, curriculum or course of study. This may be Data Not Available when pro not been defined by the institution. The alternate curriculum identification is by a combinatic degree and major codes. -- BPRA Lineage: (EDW: WDT_ACADEMIC_STUDY.PROGRAM<--C ACADEMIC_STUDY.PROGRAM<--Banner: SORLCUR.SORLCUR_PROGRAM) (Note: This data Rule: ACADEMIC_PROGRAM).
Path	[Performance Management].[Presentation View].[Analyze Student Engagement].[Academic
Ref	[Analyze Student Engagement].[Academic Study].[Program]
Data type	characterLength16
Regular aggregate	Unsupported
Usage	Attribute
Display type	Value
Prompt type	Generated Prompt
Prompt display item ref	

Figure 24: View BPRA Meta Data in Report Studio

View BPRA Meta Data details

If you are using Cognos 8.4 or higher, you can also view more detailed information in both Query Studio and Report Studio. Right-click a query item and select the Lineage option to view the Database and Technical information displayed in Query Studio and Report Studio as illustrated in the following figures.



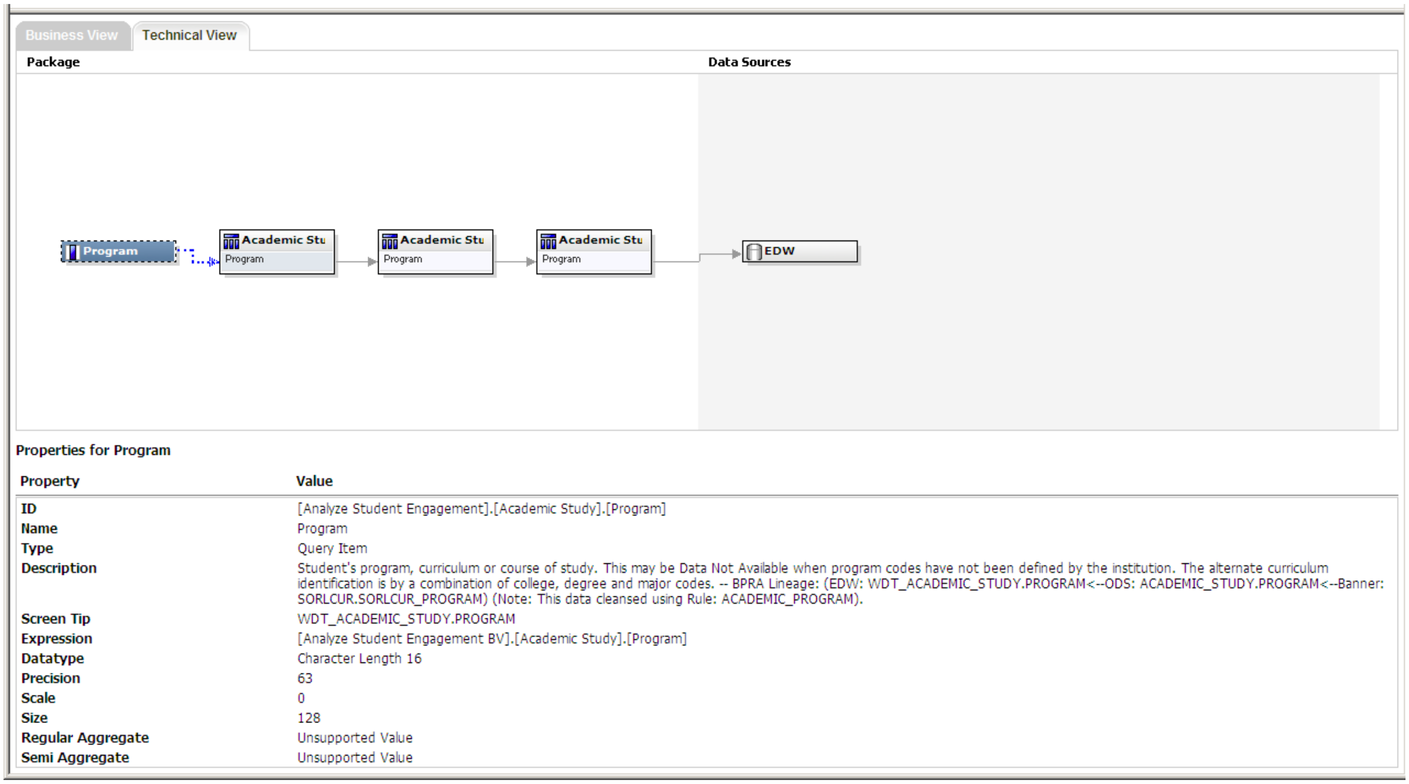
PM Analyze Student Engagement

Description	
Owner	Anonymous
Contact	

Package Item 1

Name	Program
Path	PM Analyze Student Engagement > Analyze Student Engagement > Academic Study > Program
Description	Student's program, curriculum or course of study. This may be Data Not Available when program codes have not been defined by the institution. The alternate curriculum identification is by a combination of college, degree and major codes. -- BPRA Lineage: (EDW: WDT_ACADEMIC_STUDY.PROGRAM<--ODS: ACADEMIC_STUDY.PROGRAM<--Banner: SORLCUR.SORLCUR_PROGRAM) (Note: This data cleansed using Rule: ACADEMIC_PROGRAM).
Data Sources	**Name** EDW **Description**

Figure 25: View Database and Technical BPRA Meta Data in Query Studio

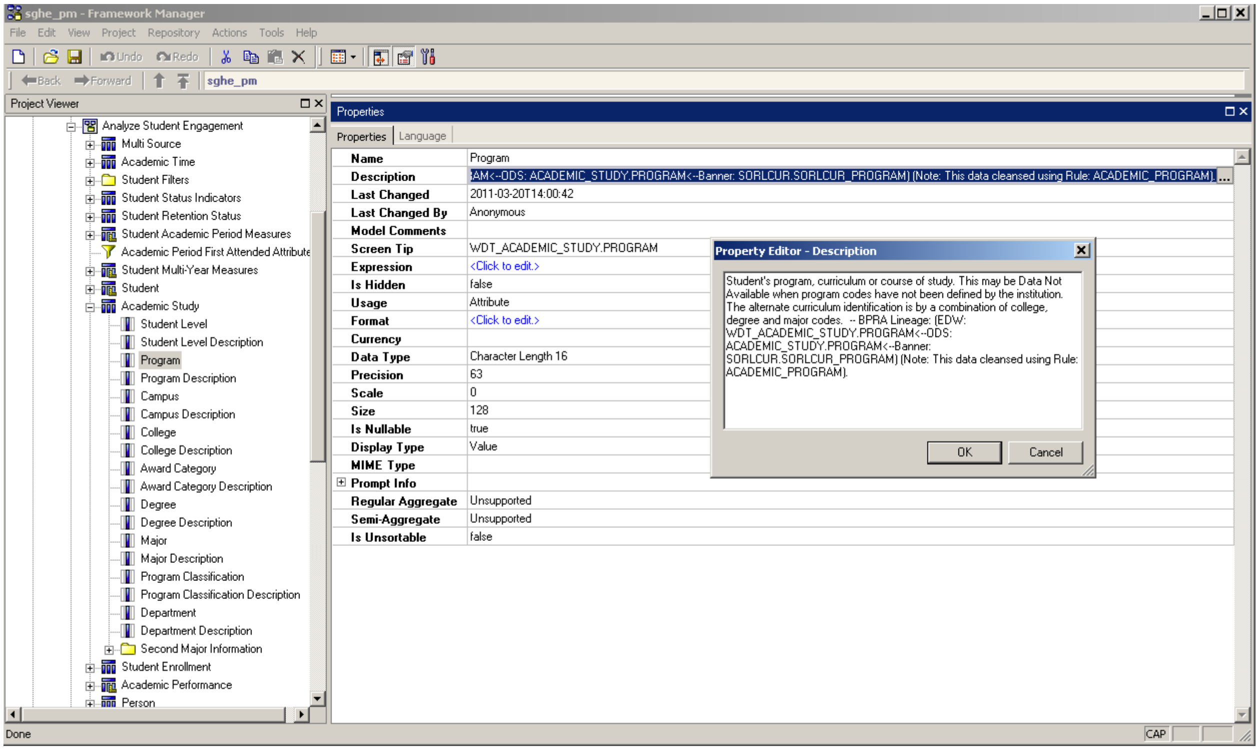


Properties for Program

Property	Value
ID	[Analyze Student Engagement].[Academic Study].[Program]
Name	Program
Type	Query Item
Description	Student's program, curriculum or course of study. This may be Data Not Available when program codes have not been defined by the institution. The alternate curriculum identification is by a combination of college, degree and major codes. -- BPRA Lineage: (EDW: WDT_ACADEMIC_STUDY.PROGRAM<--ODS: ACADEMIC_STUDY.PROGRAM<--Banner: SORLCUR.SORLCUR_PROGRAM) (Note: This data cleansed using Rule: ACADEMIC_PROGRAM).
Screen Tip	WDT_ACADEMIC_STUDY.PROGRAM
Expression	[Analyze Student Engagement BV].[Academic Study].[Program]
Datatype	Character Length 16
Precision	63
Scale	0
Size	128
Regular Aggregate	Unsupported Value
Semi Aggregate	Unsupported Value

Figure 26: View Database and Technical BPRA Meta Data in Report Studio

The BPRA Meta Data business definition and lineage are stored in the FM model query item Description as illustrated in the following picture.



Name	Program
Description	AM<--ODS: ACADEMIC_STUDY.PROGRAM<--Banner: SORLCUR.SORLCUR_PROGRAM) (Note: This data cleansed using Rule: ACADEMIC_PROGRAM).
Last Changed	2011-03-20T14:00:42
Last Changed By	Anonymous
Model Comments	
Screen Tip	WDT_ACADEMIC_STUDY.PROGRAM
Expression	<Click to edit.>
Is Hidden	false
Usage	Attribute
Format	<Click to edit.>
Currency	
Data Type	Character Length 16
Precision	63
Scale	0
Size	128
Is Nullable	true
Display Type	Value
MIME Type	
⊞ Prompt Info	
Regular Aggregate	Unsupported
Semi-Aggregate	Unsupported
Is Unsortable	false

Property Editor - Description

Student's program, curriculum or course of study. This may be Data Not Available when program codes have not been defined by the institution. The alternate curriculum identification is by a combination of college, degree and major codes. -- BPRA Lineage: (EDW: WDT_ACADEMIC_STUDY.PROGRAM<--ODS: ACADEMIC_STUDY.PROGRAM<--Banner: SORLCUR.SORLCUR_PROGRAM) (Note: This data cleansed using Rule: ACADEMIC_PROGRAM).

Figure 27: BPRA Meta Data Information in Framework Manager

The EDW source table and column are also stored in the FM model query item Screen Tip, which will be displayed when you move the cursor over a query item in either the Cognos Query Studio or Report Studio reporting tool.

Packages

A package is a subset of data designed to support a specific set of reporting needs. Packages may contain content designed within Framework Manager or cubes generated using the Cognos Transformer tool. They are the means by which Query Studio, Report Studio, and Analysis Studio are able to access data using the Cognos BI reporting tools. They are essentially the data sources used for reporting and analysis.

Within the various Cognos studios you can report against only one package at a time. It is important to use the correct package for the intended business purpose. When creating a new report, you are prompted to select which package to use.

Filters

Filtering capabilities simplify and enhance reporting. When using the reporting tool metadata to write a report, you can apply a filter on any columns of the report so that specific report will retrieve a subset of the data in the database.

There are multiple ways to add filters to the metadata layer. One way is to add a query item to the metadata that will filter a subset of data that is used on a regular basis. This type of filter is referred to as a stand-alone or pre-defined filter. A stand-alone filter can be included in multiple data model packages. For example, the time filter “Student Level Undergraduate” is included in several packages. The filter definition is the same across all packages that include it. When you place a stand-alone filter on a report, the report will select only the data defined with that filter.

Another way to define a filter is to apply it to an entire set of data, like a query subject in the Cognos FM Model. When there is a need to define a a subset of data by one of the attributes, a role based or alias query subject is defined. This type of filter would have the specific restriction embedded in the filter query item.

The Banner ODS and the data warehouse includes both types of filters. Your institution can define additional filters of either type within the Framework Manager tool to meet your specific reporting requirements.

Banner EDW Filters

Filter Type	Example
Embedded	Secondary School Post Secondary School
Stand-alone	Student Undergraduate Level Student Level Graduate Student Level Professional Highest Test Score Latest Test Score

Functionality

Filter	Data Elements
ACT Composite Test (filter)	ACT Composite Test
SAT Combined Test (filter)	SAT Combined Test
Student Level Undergraduate (filter)	Student Level Undergraduate
Student Level Graduate (filter)	Student Level Graduate
Student Level Professional (filter)	Student Level Professional

Preselected Records of Interest

For some business areas, it is useful to represent certain records of interest along with the entirety of records. Concepts such as ‘First Contact’, ‘Latest Contact’, and ‘Highest Test Score’ are typically of interest. It, therefore, is desirable to make such items easily available within the presentation view. To support this functionality, such concepts have been included in various query subjects where deemed to be useful.

An example of this can be found within the Contact query subject. Information relative to all contacts is included as well as content associated with the first contact and the latest contact.

Indicators

Some query subjects within the presentation layer include indicator fields. Dependent upon the database source for a given data element, the indicator is translated to a meaningful ‘Yes’ or ‘No’ representation. For those data elements drawn from dimension tables within the warehouse, indicator fields are translated via cleansing during ETL processing.

For those data elements drawn from fact tables, however, the indicator descriptions remain as either a 1 or 0. Two Parameter Map parameters, defined within the Administrative User Interface, define the ‘Yes’ and ‘No’ description values for indicators based on fact table data. Refer to the sections *Negative Indicators (0) parameter map* and *Positive Indicators (1) parameter map* for more information about them.

Derived Concepts

In some business cases it is important to associate certain values together for reporting purposes that may not otherwise have an association in the database. To address this need derived concepts have been created that utilize parameter maps to define how values translate to the new data element.

For example, the Traditional Student Ind is based on the prospective student's Admit Age as of the start date for the Academic Period in context, to determine if the prospective student should be classified as a traditional or non-traditional student at the time of their admission defined within a parameter map.

Predefined Value Concepts

There are specific data elements that are commonly used but may have different codes from client to client. For these situations, you should have a predefined element that could be driven off a parameter map to provide a standardized structure within the presentation layer. Elements such as this have been added to the presentation layer.

An example of this is the Test query subject. Analysis of undergraduate applicants typically centers on test scores such as the ACT Composite and the SAT Combined. The codes within the database, however, may vary from institution to institution. A parameter has been defined to allow these codes to be translated so that they are standardized and meaningful.

Distinct Counts

One measure often used for analysis within higher education is unduplicated headcount. This concept, as well as other unduplicated counts, can be a bit complex to create in a report because one needs to have a true understanding of what the uniqueness of a record truly is. Various counts have been added throughout the presentation layer to eliminate any such confusion and to ensure a “single version of the truth” for these measures.

Headcount is a primary example of this concept which is included in the presentation layer. An additional example would be counts based upon the set of financial aid steps a person may have completed within the Financial Aid Status query subject. A distinct count is calculated for each indicator based upon whether they have a “Yes” value.

Special Calculations

Special calculations are frequently required based upon various measures already represented in the presentation layer. Some of these useful calculations have been included in their own folders seen at the bottom of available components in the presentation layer. Examples of such calculations would be the various calculations provided with the Analyze Student Progress business concept which provides Course rates found in a folder within the Student Course query subject.

Internal Keys

Unique identifiers for people or records may be required in certain situations within Cognos when performing more complex analysis. For this reason, an additional query subject has been included within each business concept to house these unique identifiers. These values are useful when joining queries within Report Studio and when identifying distinct headcounts or applications counts within a cube model.

Cognos Security Integration

Cognos Authorization and fine-grained access

In a security context, authorization refers to permissions or defining “who can see what.” Cognos provides a complete infrastructure to define rules regarding “object” permissions (the ability to see folders or reports) as well as “data” permissions (which rows or columns of data individual users or groups are permitted to see). Cognos picks up its list of users and groups from the authentication providers defined at a given site, and maintains its own list of data permissions internally.

Data permissions can also be defined within the Banner Performance Reporting and Analytics (BPRA) database using the Fine-Grained Access (FGA) facility which allows for centralized maintenance of those rules for any non-Cognos based access as well. A typical Cognos configuration uses a single database connection (using a single Oracle username and password) for the BPRA database which does not allow for use of the BPRA FGA feature. However, it is also possible to configure Cognos to use multiple database connections, which then use the BPRA Fine-Grained Access rules.

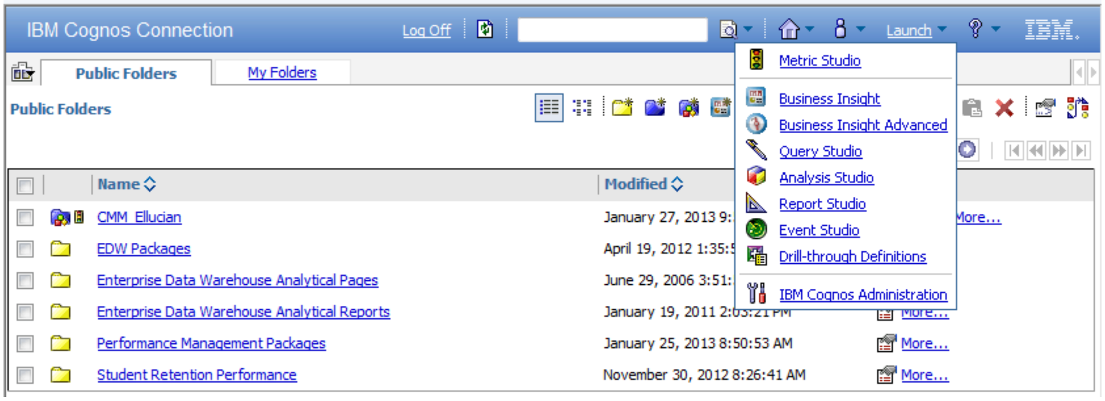
Cognos and BPRA authorization

Authorization enables you to create logins so that each user can access the same data source while still allowing them to use the fine-grained access rules already defined for them in the Administrative User Interface. Authorization could be used to set up more general Oracle users whose associated fine-grained access rules might apply to a type of report writer instead of a single person. Multiple Cognos users or roles could then be set up to secure the actual Cognos content (reports, dashboards, etc.), and then matched with data source signons which would provide the means to secure the actual data contained in the database.

For existing users, you would remove or disable the extra users so that as each user performs a query, their fine-grained access rules would be used. This should be done because their signon would be using their actual Oracle username to access the database.

1. Open Cognos Connection.

2. Click **Launch**.



Public Folders

☐		Name	Modified	
☐		CMM_Ellucian	January 27, 2013 9:	More...
☐		EDW Packages	April 19, 2012 1:35:5	
☐		Enterprise Data Warehouse Analytical Pages	June 29, 2006 3:51:	
☐		Enterprise Data Warehouse Analytical Reports	January 19, 2011 2:03:21 PM	More...
☐		Performance Management Packages	January 25, 2013 8:50:53 AM	More...
☐		Student Retention Performance	November 30, 2012 8:26:41 AM	More...

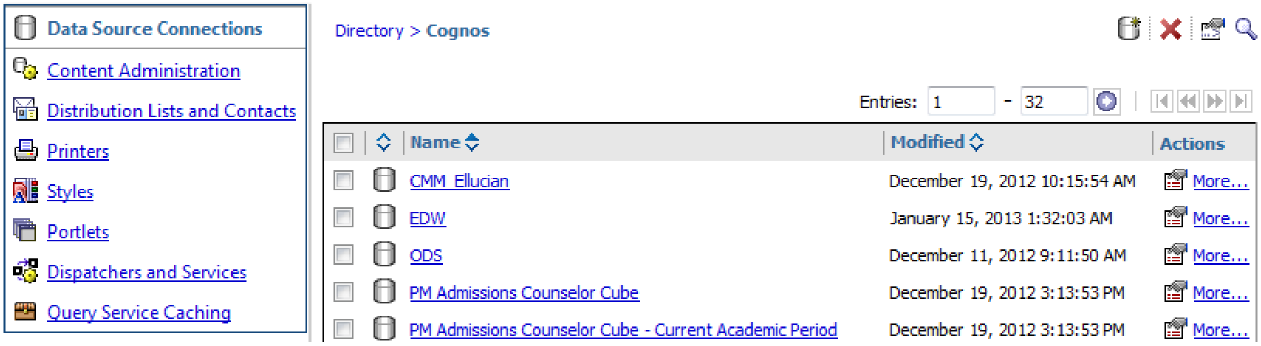
3. Click **Cognos Administration**.

4. Click the **Configuration** tab.

   The named data source connections display. The connections provide detailed connectivity information as to where to retrieve data.

5. Click one of the data sources to view the possible servers on which source data may reside.

   In the screen samples, we have chosen the warehouse data source. By default, the defined server connection has the same name as the data source connection. (See the navigation bread crumbs at the top of the screen.)



- Data Source Connections
- Content Administration
- Distribution Lists and Contacts
- Printers
- Styles
- Portlets
- Dispatchers and Services
- Query Service Caching

Directory > Cognos

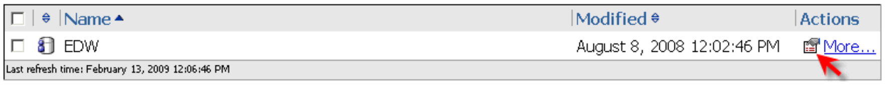
Entries: 1 - 32

☐		Name	Modified	Actions
☐		CMM_Ellucian	December 19, 2012 10:15:54 AM	More...
☐		EDW	January 15, 2013 1:32:03 AM	More...
☐		ODS	December 11, 2012 9:11:50 AM	More...
☐		PM Admissions Counselor Cube	December 19, 2012 3:13:53 PM	More...
☐		PM Admissions Counselor Cube - Current Academic Period	December 19, 2012 3:13:53 PM	More...

6. Navigate to the next layer of detail to define what users connect to this data source.

   Again, as with the server connection name, the user connection name is inherited by the data source connection unless otherwise specified.

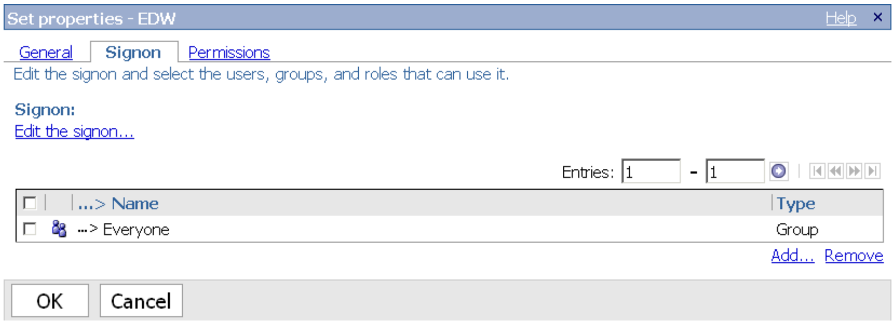
7. Click the **Set Properties** icon in the Actions column.



□	Name ▲	Modified	Actions
□	EDW	August 8, 2008 12:02:46 PM	More...

Last refresh time: February 13, 2009 12:06:46 PM

8. Click the **Signon** tab.



Set properties - EDW

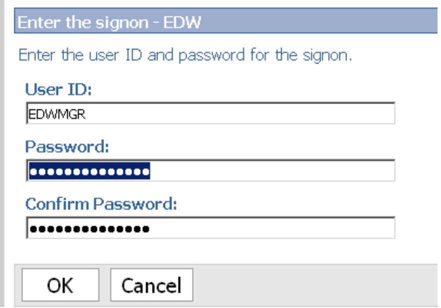
Edit the signon and select the users, groups, and roles that can use it.

Signon:
Edit the signon...

Entries: 1 - 1

□	...> Name	Type
□	...> Everyone	Group

9. Click the **Edit the signon. . .** link to view or change the Oracle username and password for this connection.



Enter the signon - EDW

Enter the user ID and password for the signon.

User ID: EDWMGR

Password: ••••••••••••••

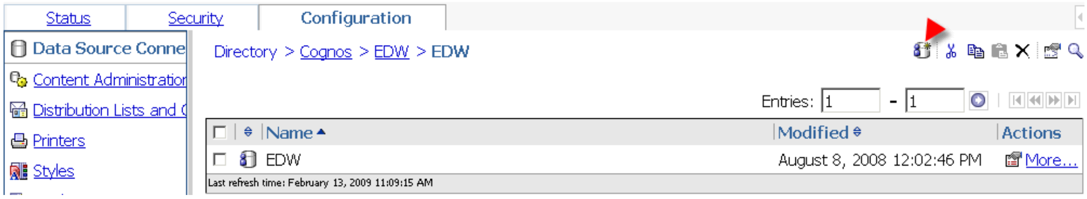
Confirm Password: ••••••••••••••

In this case you'll see the warehouse data source connection defined with a username of EDWMGR, which would have access to all data.

Let's say, for example, that your institution has two Cognos users: John Doe and Bob Smith. You would like to make use of the Oracle fine-grained access (FGA) rules that are already defined for these two users in your Administrative User Interface. Accomplishing this is a simple matter of defining two different logins to the warehouse data source that is proprietary to each user.

10. To create a new connection for the warehouse data source, return to the user connection screen within the Cognos Administration.

11. Click the **New Signon** icon.



Directory > Cognos > EDW > EDW

Entries: 1 - 1

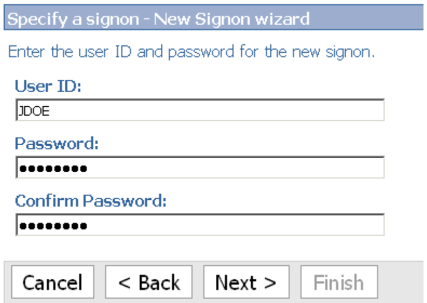
□	Name	Modified	Actions
□	EDW	August 8, 2008 12:02:46 PM	More...

Last refresh time: February 13, 2009 11:09:15 AM

12. Create a signon for John Doe and call the signon “JDOE”.

13. Click **Next**.

    You are prompted for the Oracle username and password that will be used for this signon.



Specify a signon - New Signon wizard

Enter the user ID and password for the new signon.

User ID: JDOE

Password: •••••••••

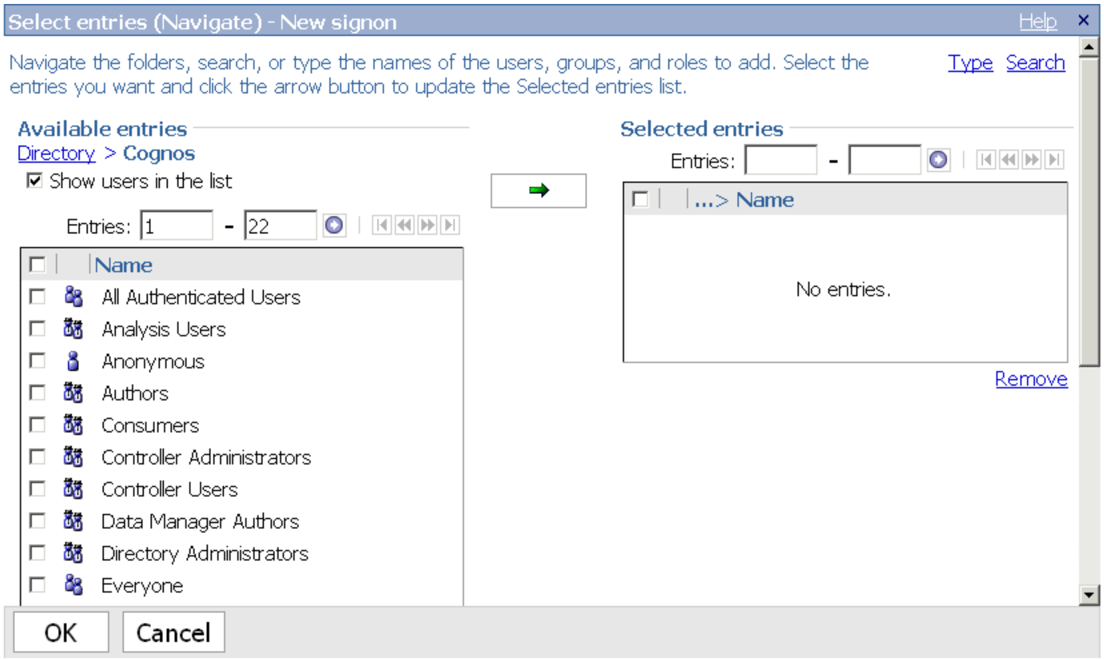
Confirm Password: •••••••••

14. Enter the information, then click **Next**.

    You are prompted for which Cognos users can access this signon.

15. Add JDOE to the list of users able to use this signon.

16. Click **OK**.



Select entries (Navigate) - New signon

Help ×

Navigate the folders, search, or type the names of the users, groups, and roles to add. Select the entries you want and click the arrow button to update the Selected entries list.

Type Search

Available entries

Directory > Cognos

☑ Show users in the list

Entries: 1 - 22

□ Name
□ All Authenticated Users
□ Analysis Users
□ Anonymous
□ Authors
□ Consumers
□ Controller Administrators
□ Controller Users
□ Data Manager Authors
□ Directory Administrators
□ Everyone

Selected entries

Entries: -

□ ...> Name

No entries.

Remove

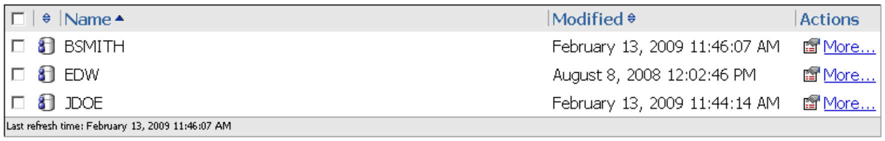
OK Cancel

17. Click **Finish.**

    You’ll see that now there is a second signon for the warehouse data source.

18. Repeat the above steps for Bob Smith.

    You will view three distinct signons for the warehouse.

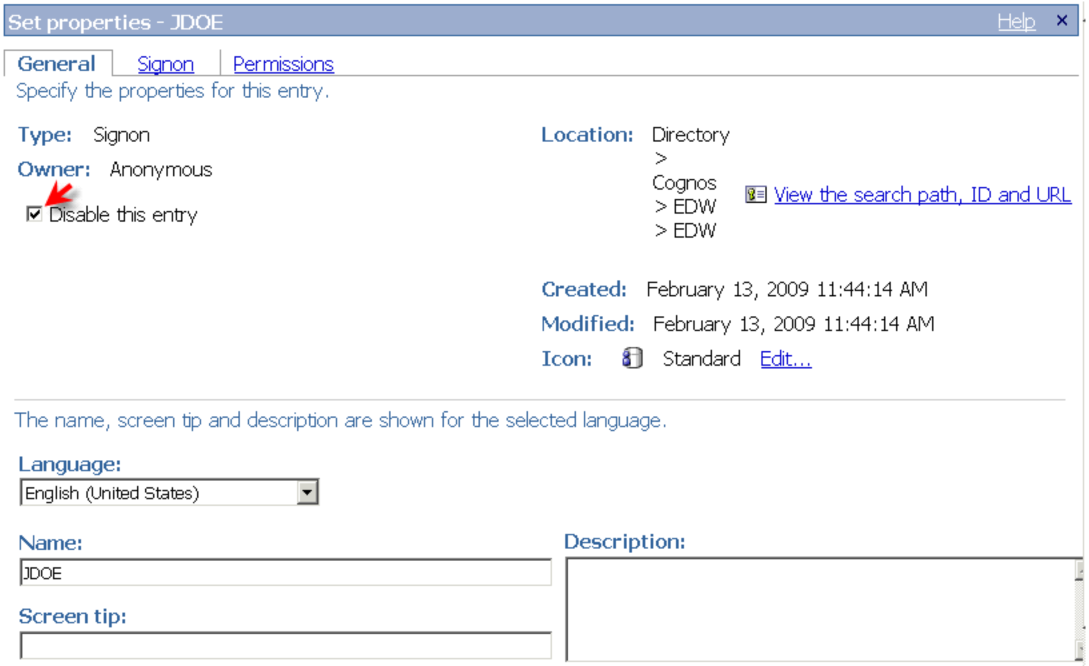


□	Name ▲	Modified	Actions
□	BSMITH	February 13, 2009 11:46:07 AM	More...
□	EDW	August 8, 2008 12:02:46 PM	More...
□	JDOE	February 13, 2009 11:44:14 AM	More...

Last refresh time: February 13, 2009 11:46:07 AM

At this point, if you logged in as John Doe, and ran a query within Cognos, you would be prompted for what signon to use. (John or Bob) This would not be an ideal situation, because you would be prompted for which connection to use each time you accessed Cognos, and the warehouse signon is not FGA secured. You, therefore, would want to remove John or Bob’s access to the warehouse signon, delete the signon, or disable it.

How to view or change what users have access to a signon was detailed previously. Deleting a signon is a straight forward activity. You select a signon and delete it. Disabling a signon is most likely the preferred method so that the overall warehouse signon is retained, but simply not active. This is a simple matter of checking the **Disable this entry** check box within the general properties of the signon.



Specify the properties for this entry.

Type: Signon

Owner: Anonymous

☑ Disable this entry

Location: Directory > Cognos > EDW > EDW

Created: February 13, 2009 11:44:14 AM

Modified: February 13, 2009 11:44:14 AM

Icon: Standard

The name, screen tip and description are shown for the selected language.

Language: English (United States)

Name: JDOE

Description:

Screen tip:

Once this signon is disabled, the signons John and Bob will be the only two active signons. Therefore, if John Doe now signs into Cognos and performs a query, he will no longer be prompted to choose a signon (because he does not have permission to use the Bob signon) and his FGA rules would be enforced on his query because his signon is using his actual Oracle username of JDOE to access the database. Similarly, if Bob Smith signs into Cognos and performs a query, his FGA rules will be enforced because his signon is using his Oracle username of BSMITH.

To put this into more practical application, one might set up more general Oracle users within the data warehouse whose associated FGA rules might apply more broadly to a type of report writer as opposed to a single person. Multiple Cognos users or roles could then be set up to secure the actual Cognos content (reports, dashboards, etc.), and matched with data source signons which would provide the means to secure the actual data contained in the database.

For additional detailed information on Cognos security, see the *Cognos Administration and Security Guide*.

Luminis authentication (single sign-on)

Authentication is the process of logging into a secured application. This section describes integrating Authentication considerations when using Cognos BI with BPRA solutions using the Luminis portal.

Usually Luminis and Cognos are configured to require users to enter a username and password to access their content. And usually, these credentials are stored and maintained separately. This requires users to log in once for Luminis and then again for Cognos every time you use Cognos within the Luminis Portal. However, this dual log-in problem can be avoided by configuring Luminis to perform Single Signon (SSO) into Cognos. Luminis provides various techniques to accomplish SSO with external applications, but the simplest is their Generic Connector Framework (GCF). (This is documented extensively in the Luminis SDK / Generic Connector Framework Implementation Guide), but basically what happens after setting up a GCF is this:

- The user sees a Cognos BI link in a Luminis page and clicks it.
- The first time a user clicks a Cognos link within Luminis they are prompted for their Cognos username/password.
- Luminis passes that through to Cognos. If it authenticates, Luminis redirects that link to the appropriate Cognos page.
- Luminis also stores that Cognos username/password, so that for future attempts, the user doesn't have to enter anything. Luminis automatically passes through the username/password and authenticates the user for them.

An important consideration regarding Cognos security is that, unlike other applications, Cognos does not have its own security infrastructure. That is, it does not have its own "user store" (where it stores usernames/passwords). Instead, it interfaces with standard security providers (such as LDAP, NTLM, Windows Active Directory, etc.) so that users can re-use existing security setups without having to duplicate them. This is fully documented in the Cognos Setup/Install documentation, as well as various other Cognos extensibility documents. So this provides an opportunity to re-use an existing user store, so that clients only have to enter/remember a single username/password.

Combining reusing an existing user store for Cognos authentication with the Luminis GCF construct simplifies SSO because users can re-use existing usernames/passwords and (after an initial Luminis session) not have to re-enter credentials to access Cognos from Luminis. The only exception to this is when their password changes. They will have to re-enter it in Luminis once.

Luminis also supports different user stores as well. By default, Luminis uses its default LDAP implementation (the SunOne Directory server) as the location where it stores security credentials, but it can also be configured to use other external systems (such as Windows Domain, or other LDAP implementations). This flexibility regarding authentication storage between Luminis and Cognos provides the client the ability to centralize their authentication processes, which can further help with the SSO process.

Determining where to store security credentials is a client-specific choice, but for SSO illustration purposes, this documentation describes how to implement that using the default Luminis LDAP implementation. Some of the concepts are applicable to other configurations as well and are noted.

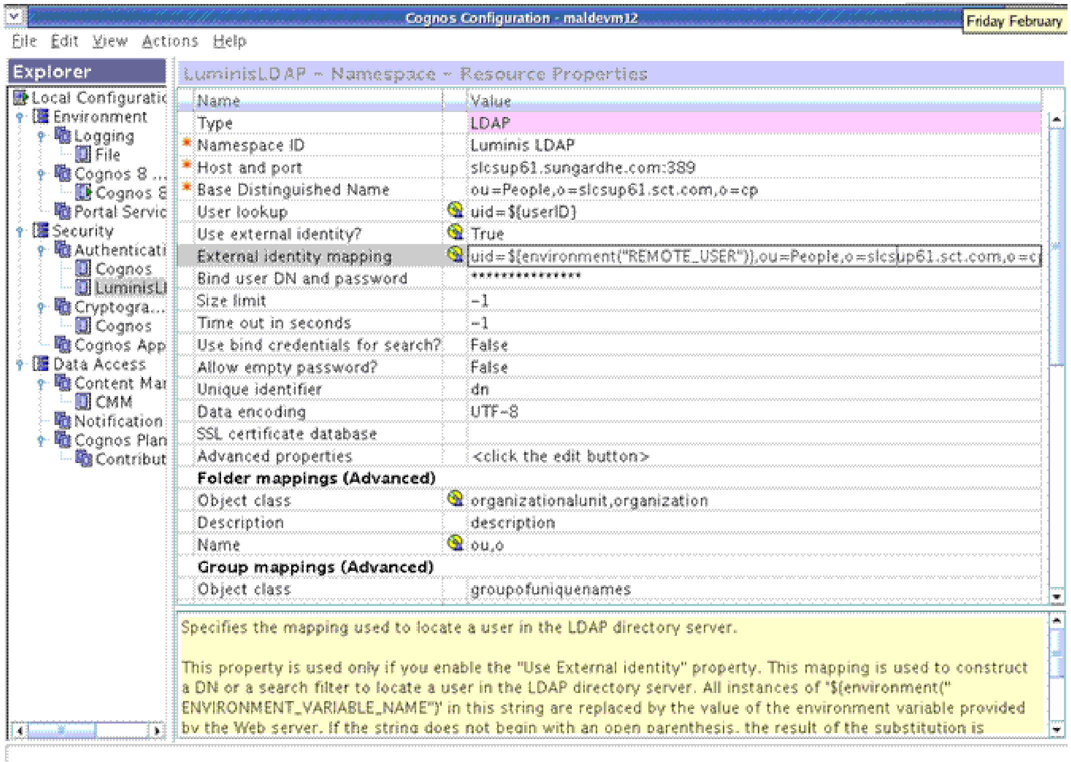
Setting up Luminis single sign-on to Cognos using Luminis LDAP authentication

These steps were written for Cognos Business Intelligence 8.3 and Luminis 4.0.2. Later releases may follow the same steps. Refer to the release-specific versions of each product's associated documentation for more details.

All sample configuration files referenced can be found in the `luminis_sso` folder, under the `ods\reports\cognos_8` folder in the ODS source tree.

In Cognos:

1. Configure an LDAP authentication namespace in Cognos to point to the Luminis LDAP instance. The properties page for the new namespace should look similar to the screen capture:



LuminisLDAP - Namespace - Resource Properties

	Name		Value
	Type		LDAP
*	Namespace ID		Luminis LDAP
*	Host and port		slcsup61.sungardhe.com:389
*	Base Distinguished Name		ou=People,o=slcsup61.sct.com,o=cp
	User lookup	●	uid=${userID}
	Use external identity?	●	True
	External identity mapping	●	uid=${environment("REMOTE_USER")},ou=People,o=slcsup61.sct.com,o=c
	Bind user DN and password		***************
	Size limit		-1
	Time out in seconds		-1
	Use bind credentials for search?		False
	Allow empty password?		False
	Unique identifier		dn
	Data encoding		UTF-8
	SSL certificate database		
	Advanced properties		<click the edit button>
	Folder mappings (Advanced)		
	Object class	●	organizationalunit,organization
	Description		description
	Name	●	ou,o
	Group mappings (Advanced)		
	Object class		groupofuniquenames

Specifies the mapping used to locate a user in the LDAP directory server.

This property is used only if you enable the "Use External identity" property. This mapping is used to construct a DN or a search filter to locate a user in the LDAP directory server. All instances of '${environment(" ENVIRONMENT_VARIABLE_NAME")}' in this string are replaced by the value of the environment variable provided by the Web server. If the string does not begin with an open parenthesis, the result of the substitution is

The majority of the default settings for an LDAP namespace can be retained with the following exceptions (as noted in the screen above, either with a red asterisk or a yellow circle icon):

NameValue

Property Name	Property Value
Namespace ID	A unique name for the namespace - can be whatever you choose
Host and port	Needs to point to the Luminis machine and LDAP listener port
Base Distinguished Name	Ou=People, o=<machine>,o=cp
User lookup	uid=$(userID)
Use external identity?	True
External identity mapping	uid=${environment("REMOTE_USER")},ou=People, o=<machine>,o=cp
Bind user DN and password	user="cn=Directory Manager" password=<Luminis LDAP pw>

Under folder mappings (advanced):

Property Names	Property Value
Object Class	organizationalunit, organization
Name	ou,o

2. Once configured, disable the Anonymous Login property in the default Cognos namespace. (Your Cognos content now requires login.)

3. Place a copy of the Luminis `pickup.html` file in the document root location of the Cognos web/application server, where it can be accessed from the Luminis machine.

In Luminis:

1. Place a copy of the `cognos.xml`, `cognos.properties` and `cognos.config` files from the distribution in the GCF connector configuration folder, specifically:

   Luminis IV

   `$CP_ROOT/webapps/cpipconnector/WEB-INF/config`

   Luminis III

   `$CP_ROOT/products/sso(`*or gcf*`)/config`

2. Edit the `cognos.properties` file and update the values of the following fields to represent your Cognos installation:

Field	Description
cognos.externalSystemURL	Point to the main URL for your Cognos environment.
cognos.pickup.remoteurl	Point to the copy of the file you placed in the Cognos environment in step 3 of the previous section.

3. Edit the `cognos.properties` file only if Cognos and Luminis are not authenticating to the same LDAP to allow the credentials to be entered the first time a person selects the link:

```
cognos.cpipconnector.getconfig.createonlogin      = 0
cognos.cpipconnector.getconfig.usePDSCredentials  = false
```

4. Edit the `cognos.config` file and make sure the property:

```
es.cognos.configURL
```

   points to your Luminis installation.

5. Edit the `cpipconnector.properties` file and append `cognos.properties` to the end of the `property.files` line toward the top of the file.

6. Perform the following configuration. Import the configuration parameters within `cognos.config` into the Luminis configuration:

```
configman –i cognos.config
```

7. Alter the `es.systems` parameter to include the cognos connector:

```
configman –g es.systems  This gets the current list of connectors

configman –s es.systems “<current list> cognos”
```

8. Restart Luminis to reload the cache with the new configuration values.

9. Build a channel using a portal admin account and the following URL:

```
http://<Site Luminis Server>/cp/ip/login?sys=cognos&url=${urlPass}
```

   or

   refer to the next section “Cognos channels in Luminis” on page 5-35.

10. Once these changes have been made, restart Luminis, or at least the cpipconnector service.

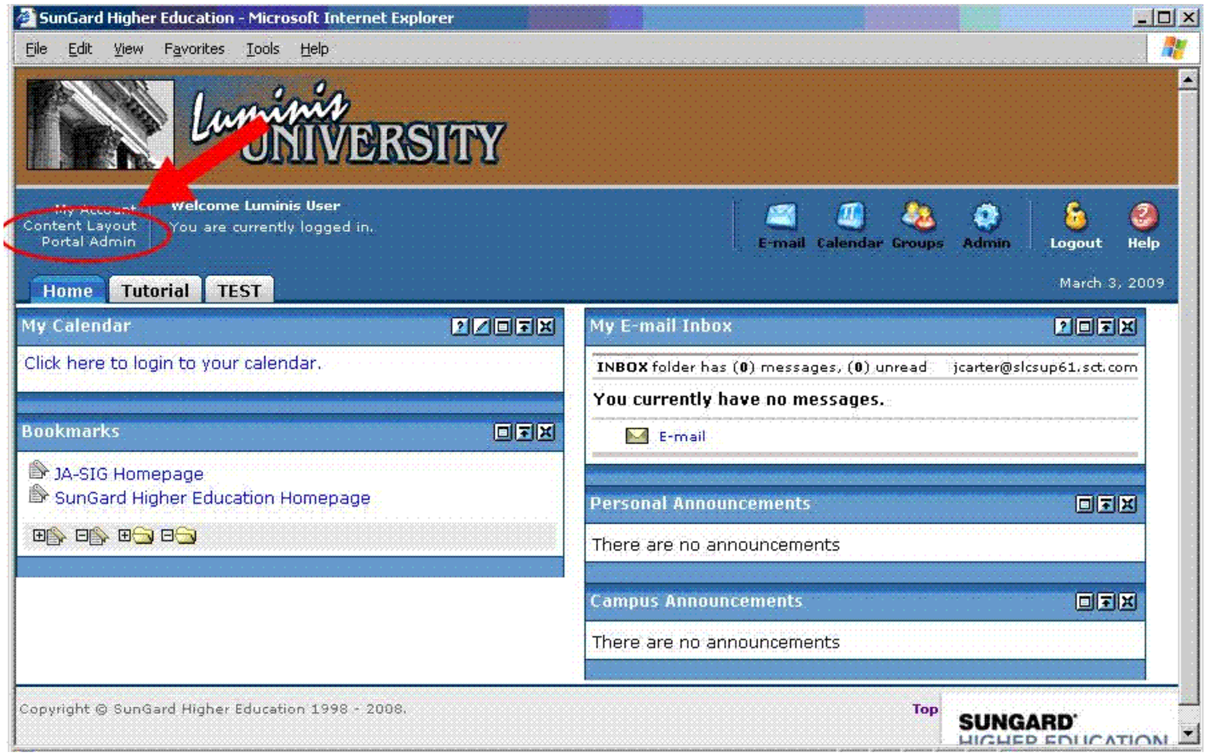
This explains how to configure Luminis and Cognos to share a username/password using Luminis’s LDAP implementation. However, both Luminis and Cognos can be configured to use other authentication sources, even potentially different ones. When they are configured to use the same source, then the password information can be maintained in a single place. If they point to different sources, Luminis can store the username/password information and then it can be configured to prompt the user to re-enter the password whenever it changes.

For more information on configuring a Luminis GCF implementation, refer to the Luminis SDK / Generic Connector Framework Implementation Guide.

You can also find more information about configuring Cognos security in the *Cognos Configuration and Installation Guide*.

Cognos channels in Luminis

Once SSO has been established, you can create links to Cognos within Luminis. Typically, this is accomplished using channels within the Luminis tabs.This process is documented in the “Luminis SDK/Channel Developer Guide”. The end result is to be able to display Cognos content within Luminis, such as in the example screen below:



Welcome Luminis User
You are currently logged in.

March 3, 2009

Click here to login to your calendar.

INBOX folder has (0) messages, (0) unread jcarter@slcsup61.sct.com

You currently have no messages.

There are no announcements

There are no announcements

Copyright © SunGard Higher Education 1998 - 2008.

The simplest way to set up the links is using CPIP Inline Frames, which can then be defined for an entire tab, or as a column (portion) of a specific tab. These tabs can then be associated with a Luminis fragment definition, which can then be rolled out to specific Luminis users, or audiences (based on Luminis role). The key is to define which Cognos content should be displayed within a channel. That is done by capturing the actual URL used to access the Cognos content, and defining that in the cognos.xml file as a variable, which can then be referenced in the Luminis channel definition URL.

For example, consider the following URL definition that is delivered in the `cognos.xml` file delivered (in the `luminis_sso` folder in the data store source tree):

```
<SET a:symbol="urlPass" a:value=
"${properties.externalSystemURL}/${properties.cognosSystemID}/
cgi-bin/cognos.cgi?b_action=xts.run&amp;m=portal/
cc.xts&amp;gohome=&amp;ui=" />
```

This defines a CPIP variable called urlPass which points to the base Cognos URL for Cognos Connection viewer.

Note

Notice the use of the es.externalSystemURL and es.cognosSystemID variables, which are defined in the properties file for the cognos CPIP definition. This convention allows you to parameterize commonly used portions of URL definitions.

Also notice the conversion of all ampersand characters to the URL-encoding equivalent. This is required for proper parsing of the URL in the XML syntax.

This variable “urlPass” can now be used when referencing Cognos via a Luminis channel definition, as per:

```
http://<Site Luminis Server>/cp/ip/
login?sys=cognos&url=${urlPass}
```

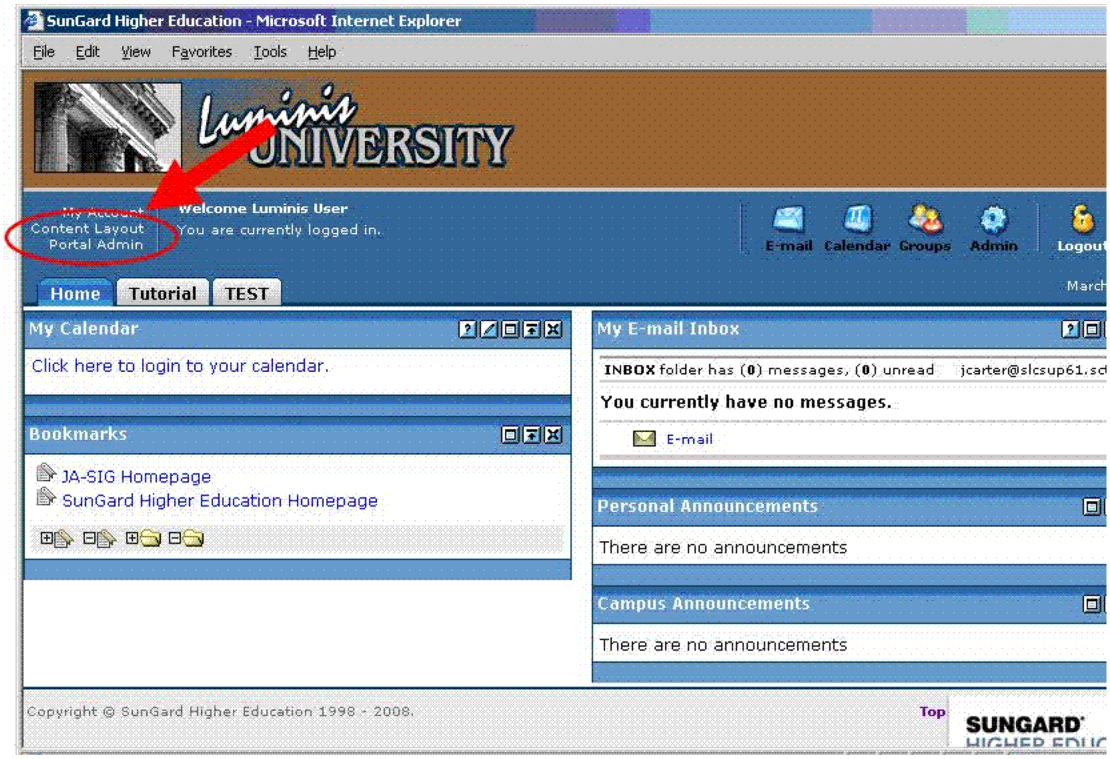
which points to the Cognos Connection viewer (based on the definition of urlPass). By defining CPIP variables in the XML file to point to the desired Cognos reports/pages you wish to expose in Luminis, you can then create Luminis channels using those variables.

A series of example Cognos URLs are delivered as variables in the `cognos.xml` file (urlPass, cogURL1, cogURL2, cogDash1 - cogDash4). These demonstrate the ability to define various Cognos content (reports) that can be viewed specifically using a Luminis channel, and these can be modified/updated/deleted as needed. Note that these URL values need to be URL-encoded when they are stored in the XML file for proper parsing by Luminis.

Setting up Cognos channels in Luminis

Now we will use all the pieces of what we have defined so far to create a basic Luminis channel to display the standard Cognos Connection viewer application. To start, assume a new user is defined in Luminis (who has Luminis Administrative privileges, in order to administer the portal and content layout).

1. Click the **Portal Admin** link to define the channel.



Welcome Luminis User
You are currently logged in.

My Calendar

Click here to login to your calendar.

Bookmarks

JA-SIG Homepage
SunGard Higher Education Homepage

My E-mail Inbox

INBOX folder has (0) messages, (0) unread jcarter@slcsup61.sd

You currently have no messages.

E-mail

Personal Announcements

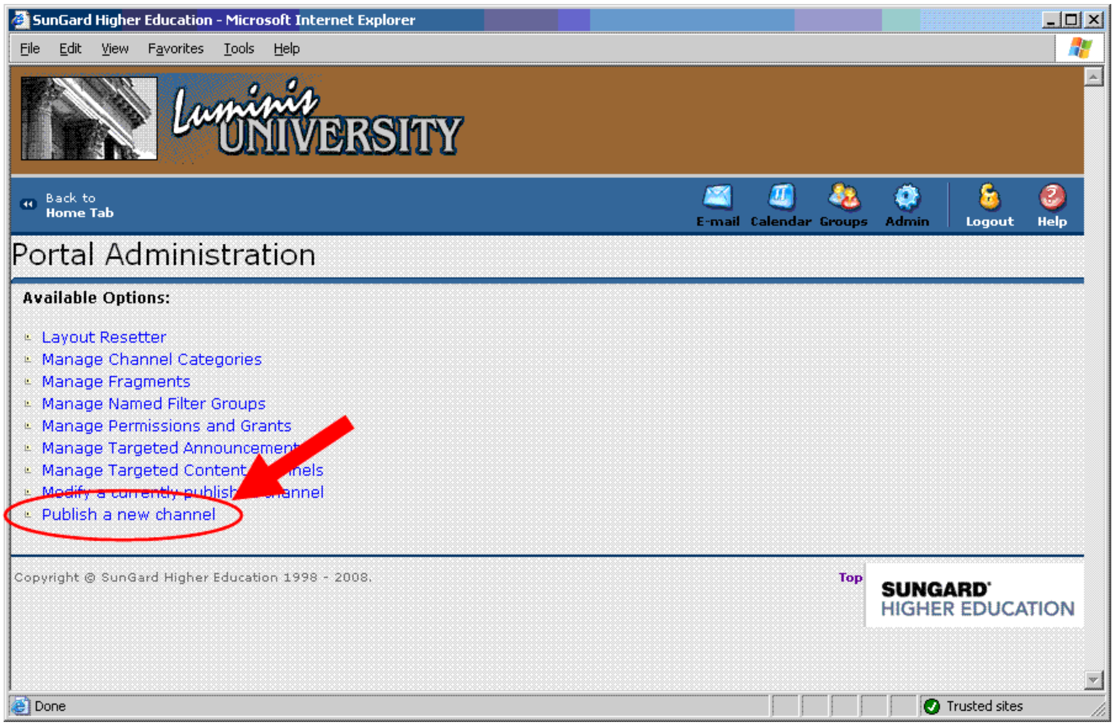
There are no announcements

Campus Announcements

There are no announcements

Copyright © SunGard Higher Education 1998 - 2008.

2. Select **Publish a new channel**.


SunGard Higher Education - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Luminis UNIVERSITY
Back to Home Tab
E-mail Calendar Groups Admin Logout Help
Top
SUNGARD HIGHER EDUCATION
Done
Trusted sites

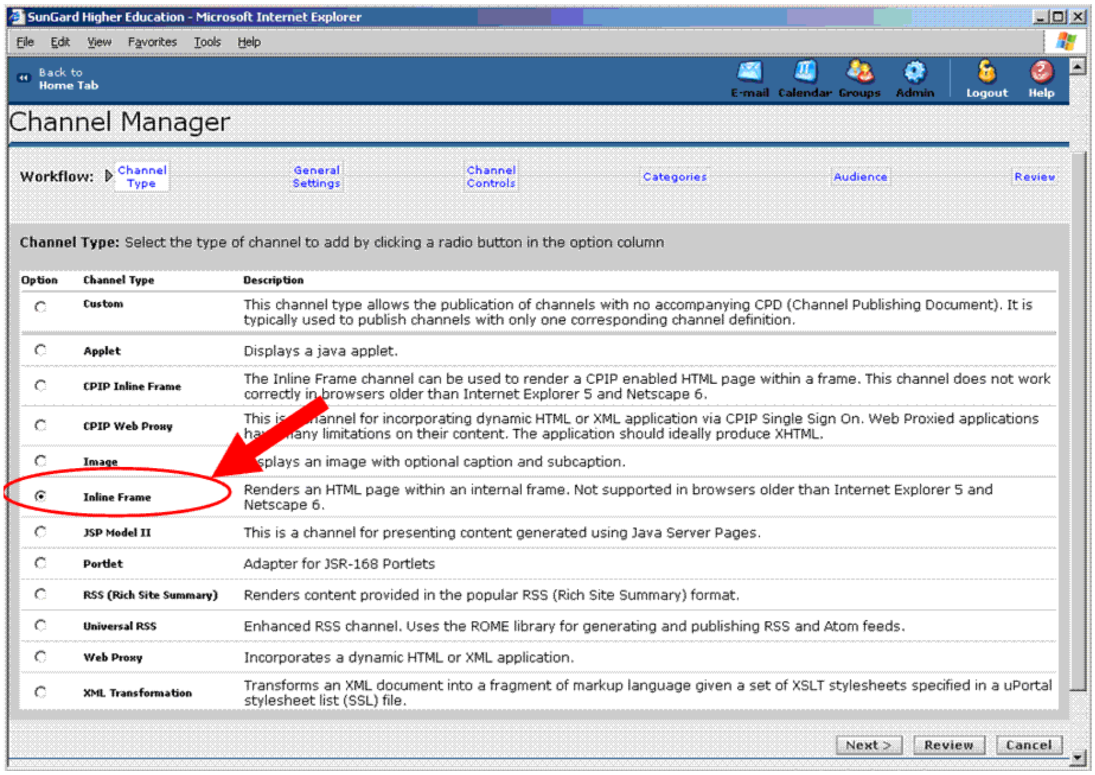
Portal Administration

Available Options:

- Layout Resetter
- Manage Channel Categories
- Manage Fragments
- Manage Named Filter Groups
- Manage Permissions and Grants
- Manage Targeted Announcements
- Manage Targeted Content Channels
- Modify a currently published channel
- Publish a new channel

Copyright © SunGard Higher Education 1998 - 2008.

3. Select **Inline Frame** as the Channel type.


SunGard Higher Education - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back to Home Tab
E-mail Calendar Groups Admin Logout Help
Workflow:
Channel Type
General Settings
Channel Controls
Categories
Audience
Review
Next >
Review
Cancel

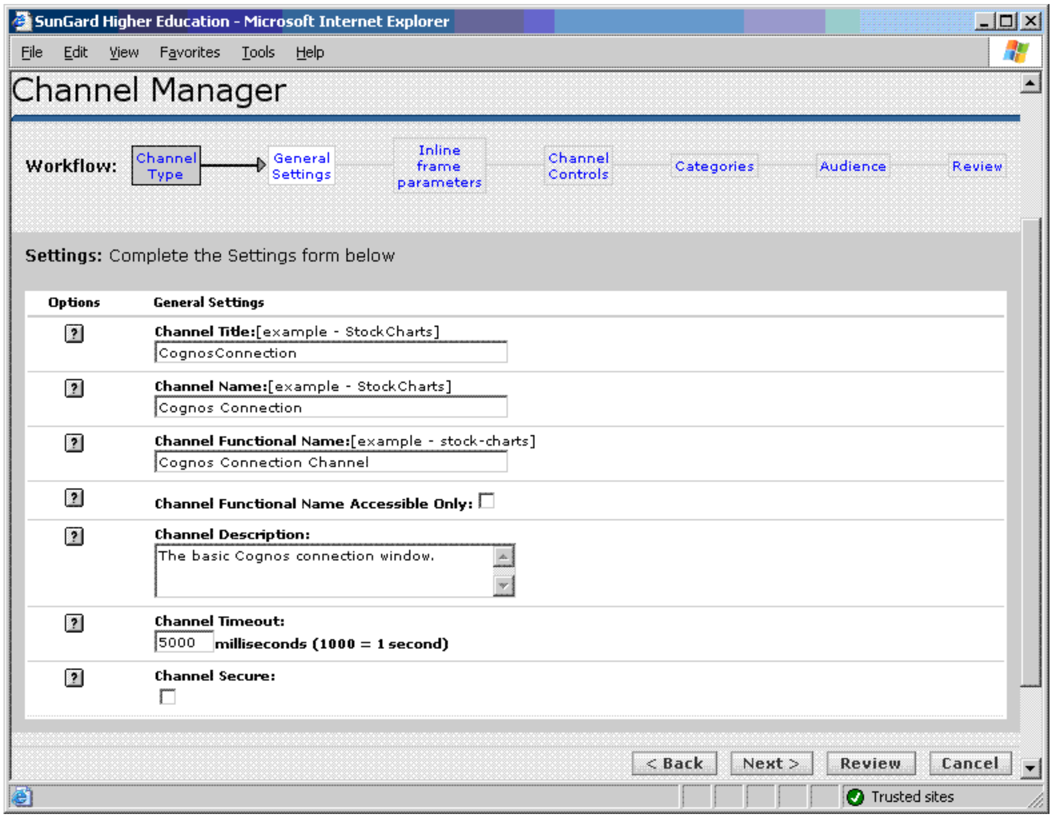
Channel Manager

Channel Type: Select the type of channel to add by clicking a radio button in the option column

Option	Channel Type	Description
○	Custom	This channel type allows the publication of channels with no accompanying CPD (Channel Publishing Document). It is typically used to publish channels with only one corresponding channel definition.
○	Applet	Displays a java applet.
○	CPIP Inline Frame	The Inline Frame channel can be used to render a CPIP enabled HTML page within a frame. This channel does not work correctly in browsers older than Internet Explorer 5 and Netscape 6.
○	CPIP Web Proxy	This is a channel for incorporating dynamic HTML or XML application via CPIP Single Sign On. Web Proxied applications have many limitations on their content. The application should ideally produce XHTML.
○	Image	Displays an image with optional caption and subcaption.
●	Inline Frame	Renders an HTML page within an internal frame. Not supported in browsers older than Internet Explorer 5 and Netscape 6.
○	JSP Model II	This is a channel for presenting content generated using Java Server Pages.
○	Portlet	Adapter for JSR-168 Portlets
○	RSS (Rich Site Summary)	Renders content provided in the popular RSS (Rich Site Summary) format.
○	Universal RSS	Enhanced RSS channel. Uses the ROME library for generating and publishing RSS and Atom feeds.
○	Web Proxy	Incorporates a dynamic HTML or XML application.
○	XML Transformation	Transforms an XML document into a fragment of markup language given a set of XSLT stylesheets specified in a uPortal stylesheet list (SSL) file.

4. Click **Next**

5. Enter the title, names, and description information for the channel.



Channel Manager

Settings: Complete the Settings form below

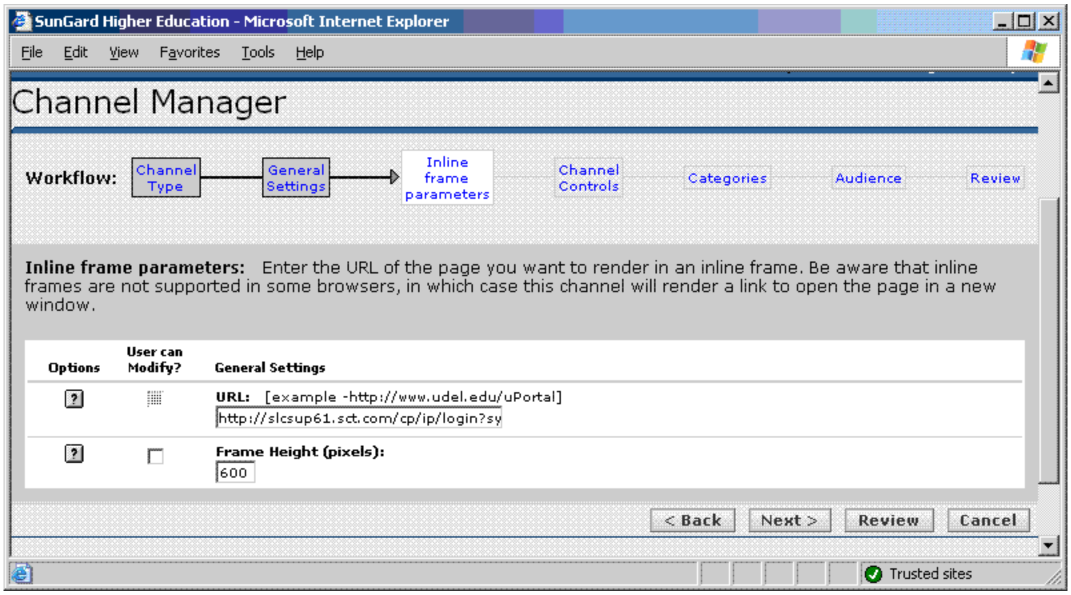
Options	General Settings
?	**Channel Title:**[example - StockCharts] CognosConnection
?	**Channel Name:**[example - StockCharts] Cognos Connection
?	**Channel Functional Name:**[example - stock-charts] Cognos Connection Channel
?	**Channel Functional Name Accessible Only:** □
?	**Channel Description:** The basic Cognos connection window.
?	**Channel Timeout:** 5000 **milliseconds (1000 = 1 second)**
?	**Channel Secure:** □

6. Click **Next**.

7. Enter the URL for this channel, which is the CPIP definition described earlier:

```
http://<Site Luminis Server>/cp/ip/
login?sys=cognos&url=${urlPass}
```

Include the CPIP variable “urlPass” which points to the desired Cognos content.


SunGard Higher Education - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Workflow: Channel Type → General Settings → Inline frame parameters · Channel Controls · Categories · Audience · Review
< Back
Next >
Review
Cancel
Trusted sites

Channel Manager

Inline frame parameters: Enter the URL of the page you want to render in an inline frame. Be aware that inline frames are not supported in some browsers, in which case this channel will render a link to open the page in a new window.

Options	User can Modify?	General Settings
?		URL: [example -http://www.udel.edu/uPortal] http://slcsup61.sct.com/cp/ip/login?sy
?	□	Frame Height (pixels): 600

8. Click **Next**.

9. Click **Next** to accept the default values for Channel Controls.

10. Select a category (or categories) for the channel to be associated with.

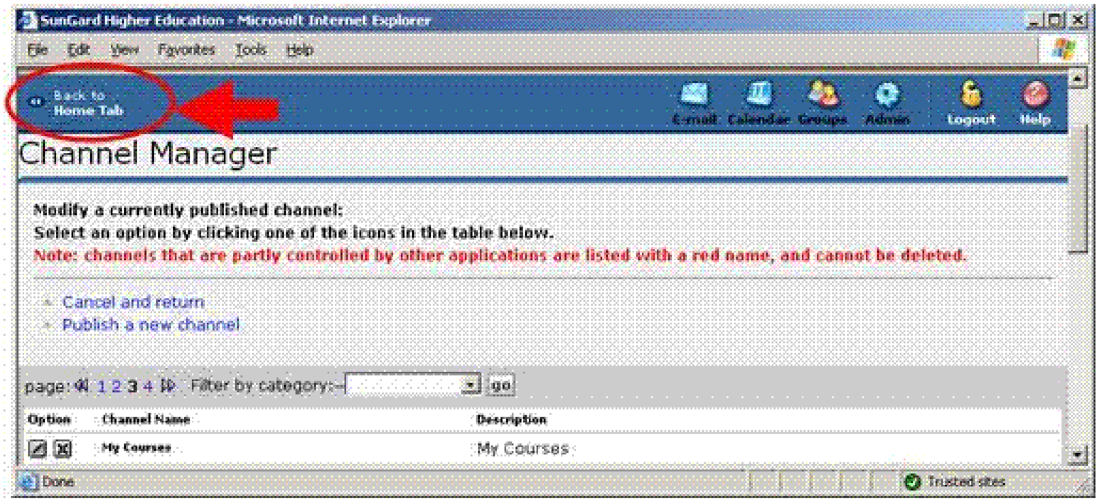
11. Click **Next**.

    (The category is used to locate channels when searching for them later)

12. Click **Next** to accept the default values for Audience.

13. Click **Finished** to publish this channel.

14. Click the **Back to Home Tab** link in the upper left-hand side of the screen to return to the main Luminis page.


SunGard Higher Education - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back to Home Tab
E-mail Calendar Groups Admin Logout Help

Channel Manager

Modify a currently published channel:
Select an option by clicking one of the icons in the table below.
Note: channels that are partly controlled by other applications are listed with a red name, and cannot be deleted.

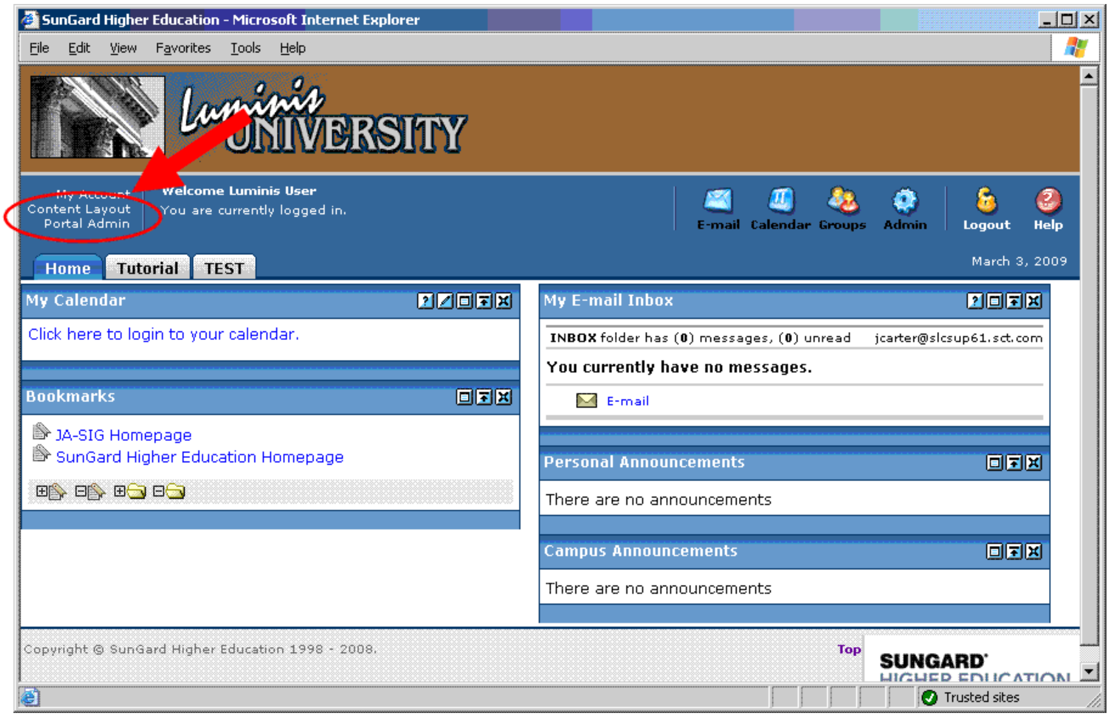
- Cancel and return
- Publish a new channel

page: 1 2 3 4 Filter by category: go

Option	Channel Name	Description
	My Courses	My Courses

The next step is to associate this channel with a tab on the portal.

15. Click the **Content Layout** link


SunGard Higher Education - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Luminis UNIVERSITY
My Account
Content Layout
Portal Admin
E-mail Calendar Groups Admin Logout Help
Home Tutorial TEST

Welcome Luminis User
You are currently logged in.

March 3, 2009

My Calendar
Click here to login to your calendar.

Bookmarks
JA-SIG Homepage
SunGard Higher Education Homepage

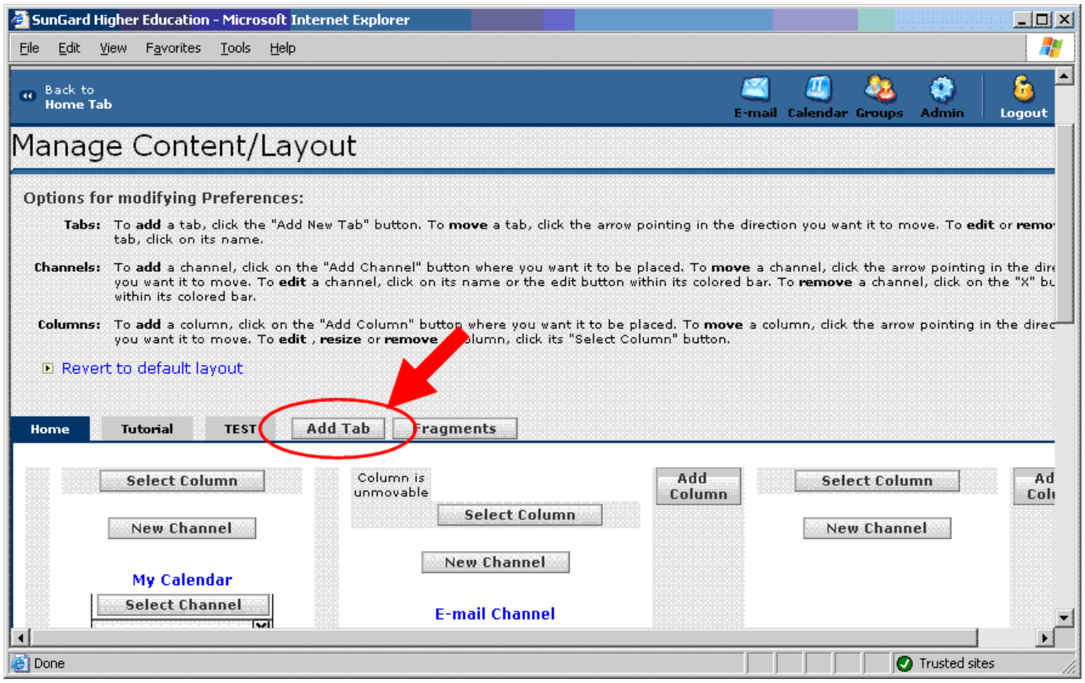
My E-mail Inbox
INBOX folder has (0) messages, (0) unread jcarter@slcsup61.sct.com
You currently have no messages.
E-mail

Personal Announcements
There are no announcements

Campus Announcements
There are no announcements

Copyright © SunGard Higher Education 1998 - 2008.

16. Click the **Add Tab** button to create a new tab.



Manage Content/Layout

Options for modifying Preferences:

Tabs: To **add** a tab, click the "Add New Tab" button. To **move** a tab, click the arrow pointing in the direction you want it to move. To **edit** or **remo** tab, click on its name.

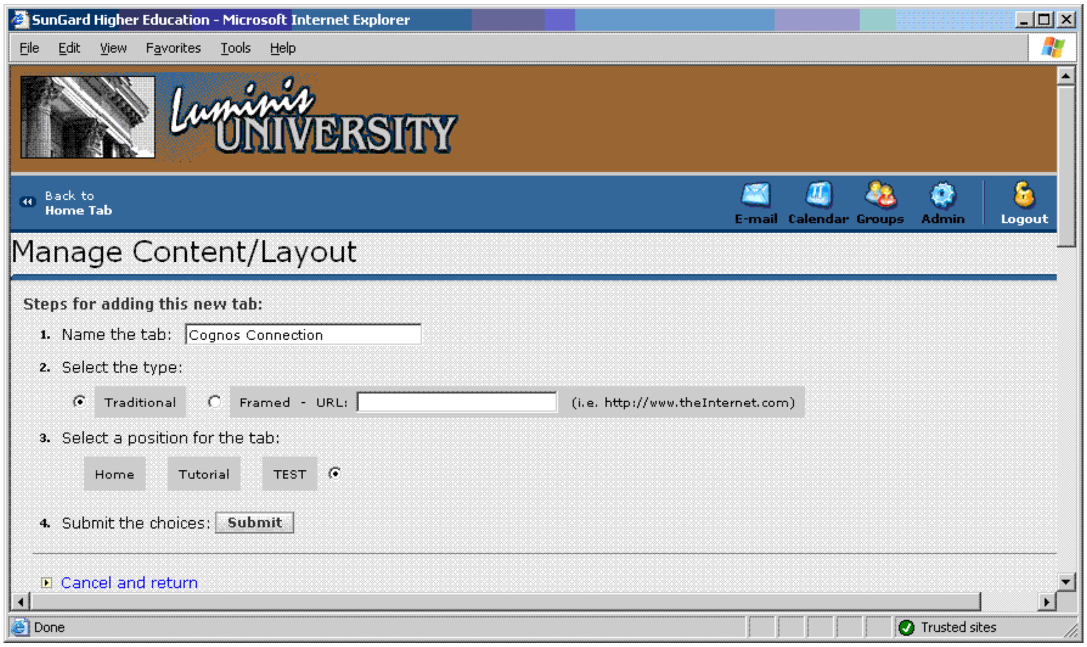
Channels: To **add** a channel, click on the "Add Channel" button where you want it to be placed. To **move** a channel, click the arrow pointing in the dire you want it to move. To **edit** a channel, click on its name or the edit button within its colored bar. To **remove** a channel, click on the "X" bu within its colored bar.

Columns: To **add** a column, click on the "Add Column" button where you want it to be placed. To **move** a column, click the arrow pointing in the direc you want it to move. To **edit** , **resize** or **remove** olumn, click its "Select Column" button.

Revert to default layout

17. Enter the name for this Tab as **Cognos Connection.**

18. Click **Submit**.


SunGard Higher Education - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Luminis UNIVERSITY
Back to Home Tab
E-mail Calendar Groups Admin Logout
Done
Trusted sites

Manage Content/Layout

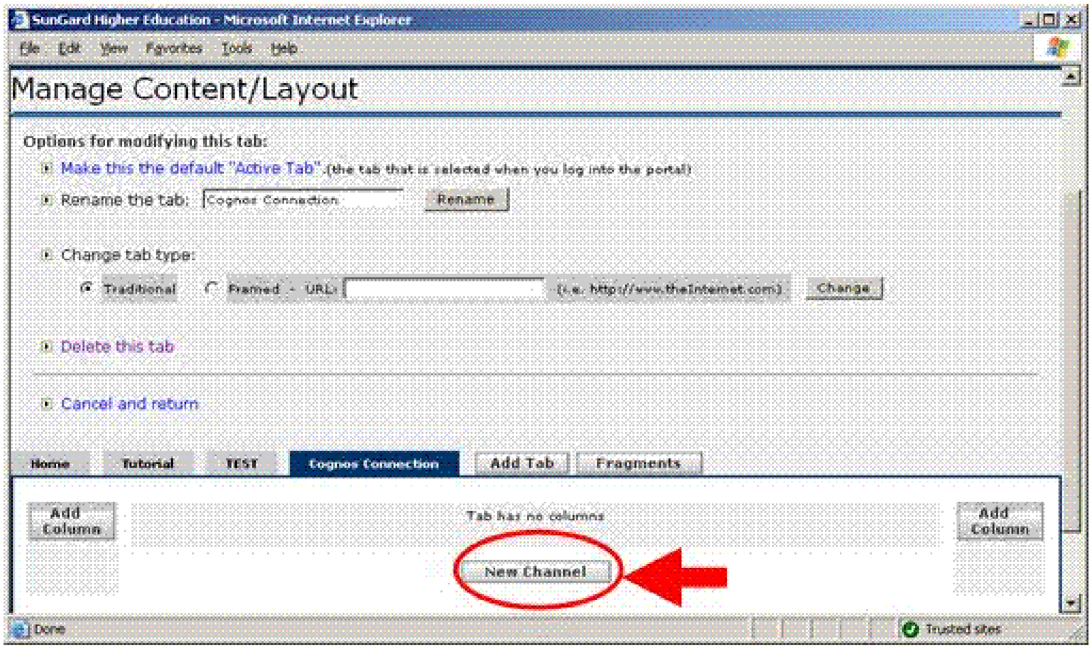
Steps for adding this new tab:

1. Name the tab: Cognos Connection
2. Select the type:
   ● Traditional ○ Framed - URL: (i.e. http://www.theInternet.com)
3. Select a position for the tab:
   Home Tutorial TEST ●
4. Submit the choices: Submit

Cancel and return

19. Select the new Cognos Connection tab.

20. Click the **New Channel** button:



Manage Content/Layout

Options for modifying this tab:

- Make this the default "Active Tab" (the tab that is selected when you log into the portal)
- Rename the tab: Cognos Connection
- Change tab type:
  ● Traditional ○ Framed - URL: (i.e. http://www.theInternet.com)
- Delete this tab
- Cancel and return

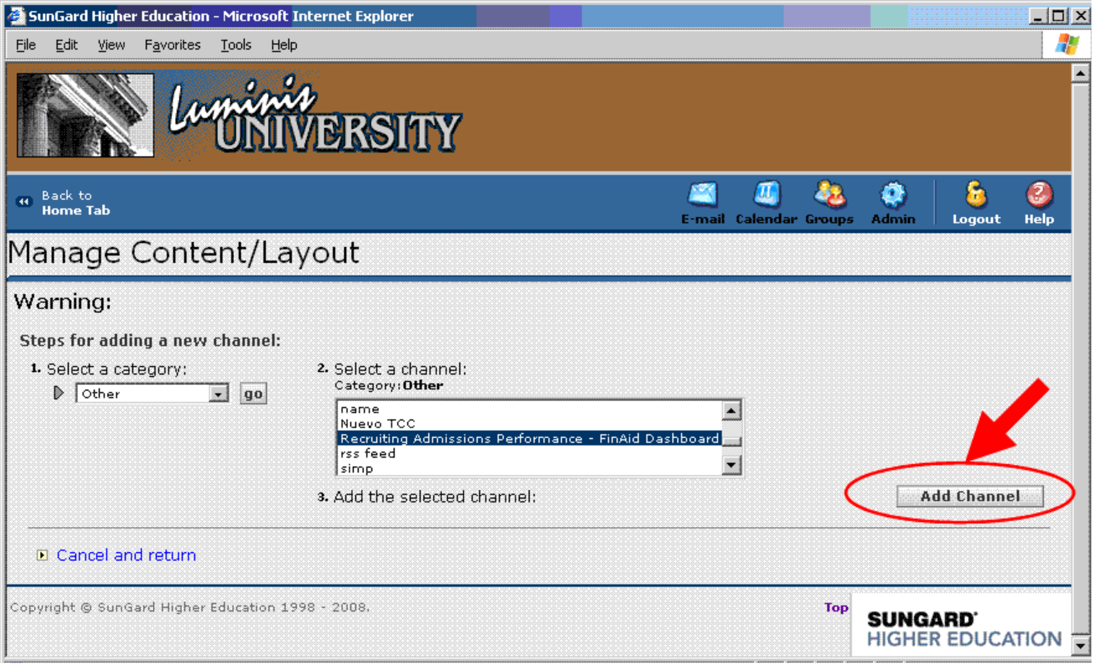
Tab has no columns

21. Select the channel by first entering the category (or **Select All**)

22. Click **Go**.

23. Select the channel from the listbox.

24. Click **Add Channel**:



Manage Content/Layout

Warning:

Steps for adding a new channel:

1. Select a category: Other

2. Select a channel:
Category: Other
name
Nuevo TCC
Recruiting Admissions Performance - FinAid Dashboard
rss feed
simp

3. Add the selected channel:

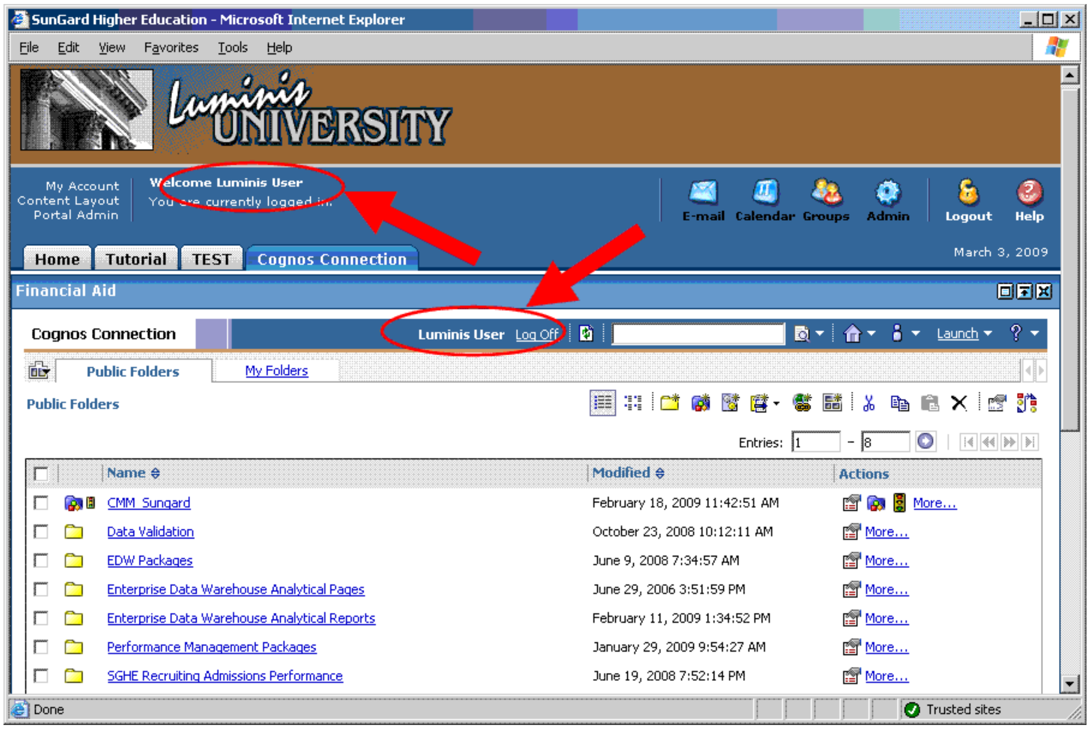
Cancel and return

Copyright © SunGard Higher Education 1998 - 2008.

25. Click the **Back to Home Tab** link to return to the main Luminis page,

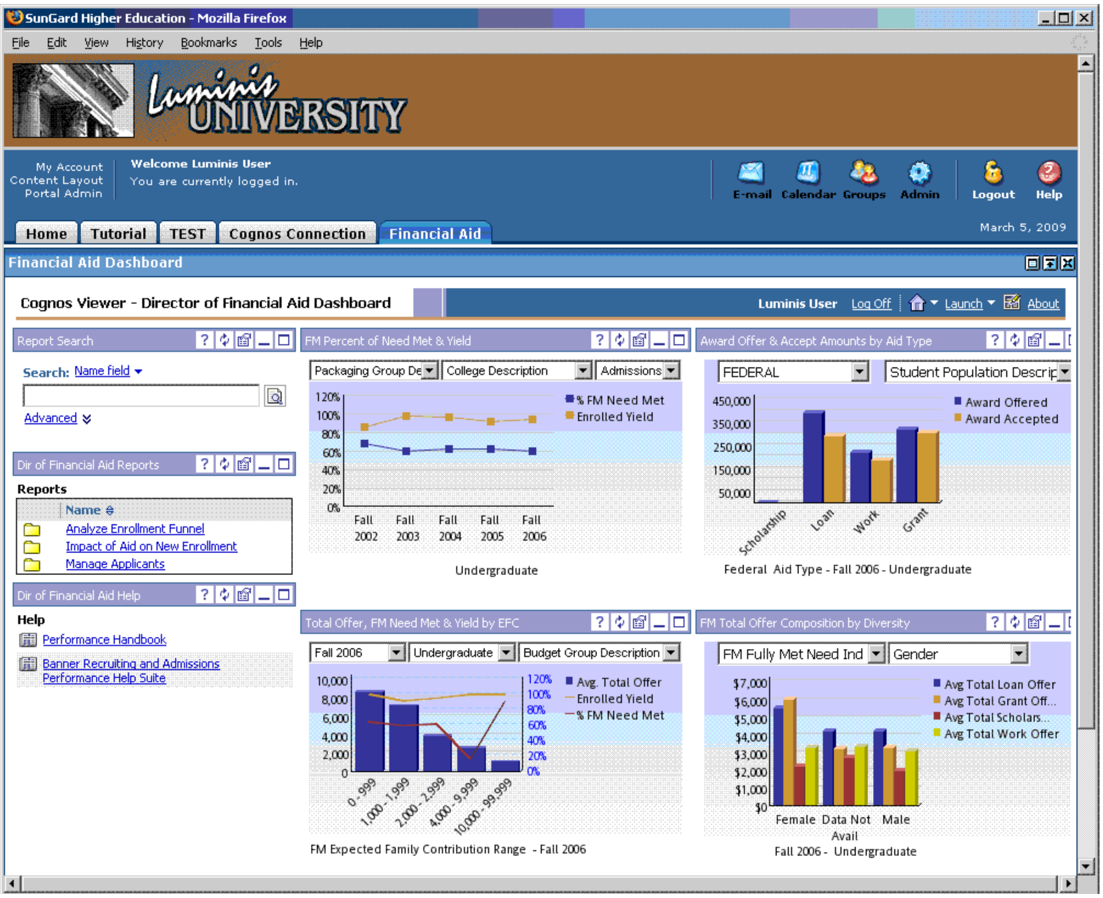
26. Click the new **Cognos Connection** tab to see the new channel.

You will see the authenticated user in both Luminis and Cognos, with the name coming from the common user store (Luminis LDAP):


SunGard Higher Education - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Luminis UNIVERSITY
My Account
Content Layout
Portal Admin
Welcome Luminis User
You are currently logged in.
E-mail Calendar Groups Admin Logout Help
March 3, 2009
Home Tutorial TEST Cognos Connection
Financial Aid
Cognos Connection
Luminis User Log Off
Launch
Public Folders My Folders
Public Folders
Entries: 1 - 8
Done
Trusted sites

□	Name	Modified	Actions
□	CMM_Sungard	February 18, 2009 11:42:51 AM	More...
□	Data Validation	October 23, 2008 10:12:11 AM	More...
□	EDW Packages	June 9, 2008 7:34:57 AM	More...
□	Enterprise Data Warehouse Analytical Pages	June 29, 2006 3:51:59 PM	More...
□	Enterprise Data Warehouse Analytical Reports	February 11, 2009 1:34:52 PM	More...
□	Performance Management Packages	January 29, 2009 9:54:27 AM	More...
□	SGHE Recruiting Admissions Performance	June 19, 2008 7:52:14 PM	More...

Following the same basic process, any Cognos pages can be deployed within Luminis, such as in the example screen below:


SunGard Higher Education - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Luminis UNIVERSITY
My Account
Content Layout
Portal Admin
Welcome Luminis User
You are currently logged in.
E-mail Calendar Groups Admin Logout Help
Home Tutorial TEST Cognos Connection Financial Aid
March 5, 2009
Financial Aid Dashboard
Cognos Viewer - Director of Financial Aid Dashboard
Luminis User Log Off Launch About
Report Search
Search: Name field
Advanced
FM Percent of Need Met & Yield
Packaging Group De College Description Admissions
% FM Need Met
Enrolled Yield
120% 100% 80% 60% 40% 20% 0%
Fall 2002 Fall 2003 Fall 2004 Fall 2005 Fall 2006
Undergraduate
Award Offer & Accept Amounts by Aid Type
FEDERAL Student Population Descrip
Award Offered
Award Accepted
450,000 350,000 250,000 150,000 50,000
Scholarship Loan Work Grant
Federal Aid Type - Fall 2006 - Undergraduate
Dir of Financial Aid Reports
Reports
Name
Analyze Enrollment Funnel
Impact of Aid on New Enrollment
Manage Applicants
Dir of Financial Aid Help
Help
Performance Handbook
Banner Recruiting and Admissions Performance Help Suite
Total Offer, FM Need Met & Yield by EFC
Fall 2006 Undergraduate Budget Group Description
Avg. Total Offer
Enrolled Yield
% FM Need Met
10,000 8,000 6,000 4,000 2,000 0
120% 100% 80% 60% 40% 20% 0%
0 - 999 1,000 - 1,999 2,000 - 2,999 4,000 - 9,999 10,000 - 99,999
FM Expected Family Contribution Range - Fall 2006
FM Total Offer Composition by Diversity
FM Fully Met Need Ind Gender
Avg Total Loan Offer
Avg Total Grant Off...
Avg Total Scholars...
Avg Total Work Offer
$7,000 $6,000 $5,000 $4,000 $3,000 $2,000 $1,000 $0
Female Data Not Avail Male
Fall 2006 - Undergraduate

This screen is using the Cognos URL for the “Director of Financial Aid Dashboard”, which is defined in the cognos.xml file as:

```
<SET a:symbol="cogURL2"
a:value="${properties.externalSystemURL}/
${properties.cognosSystemID}/cgi-bin/
cognos.cgi?b_action=dashboard&amp;pathinfo=/cm&amp;frag-
header=true&amp;path=storeID(%22i04AD276242AF47B680223538F724B0
6C%22)&amp;ui=h1h3h4" />
```

so that the channel definition of this is then:

```
http://<Site Luminis Server>/cp/ip/
login?sys=cognos&url=${cogURL2}
```

In the Cognos URL definition, note the use of the `path=storeID` parameter to refer to the Cognos object (the dashboard report) to display. This ID number is unique within a given Cognos installation so it can be advisable to use the actual search path for the object instead of the object ID when referencing it in the URL. The search path for a page/object is found in the Properties dialog, which is available in the Cognos Connection navigator interface.

Further Cognos UI customization

While the Cognos applications (such as Cognos Connection viewer or the Studio applications) can be embedded within a Luminis page using the channel concepts discussed, some of the Cognos UI features may be unnecessary and distract from the overall usability of the page. To address this issue, Cognos provides various URL-based parameters which can control some aspects of the UI for these applications. This section describes those parameters and describes a few examples of setting these up.

Consider again the Cognos URL used to launch the Cognos Connection viewer previously, that was defined in the cognos.xml file:

```
<SET a:symbol="urlPass" a:value=
"${properties.externalSystemURL}/${properties.cognosSystemID}/
cgi-bin/cognos.cgi?b_action=xts.run&amp;m=portal/
cc.xts&amp;gohome=&amp;ui=" />
```

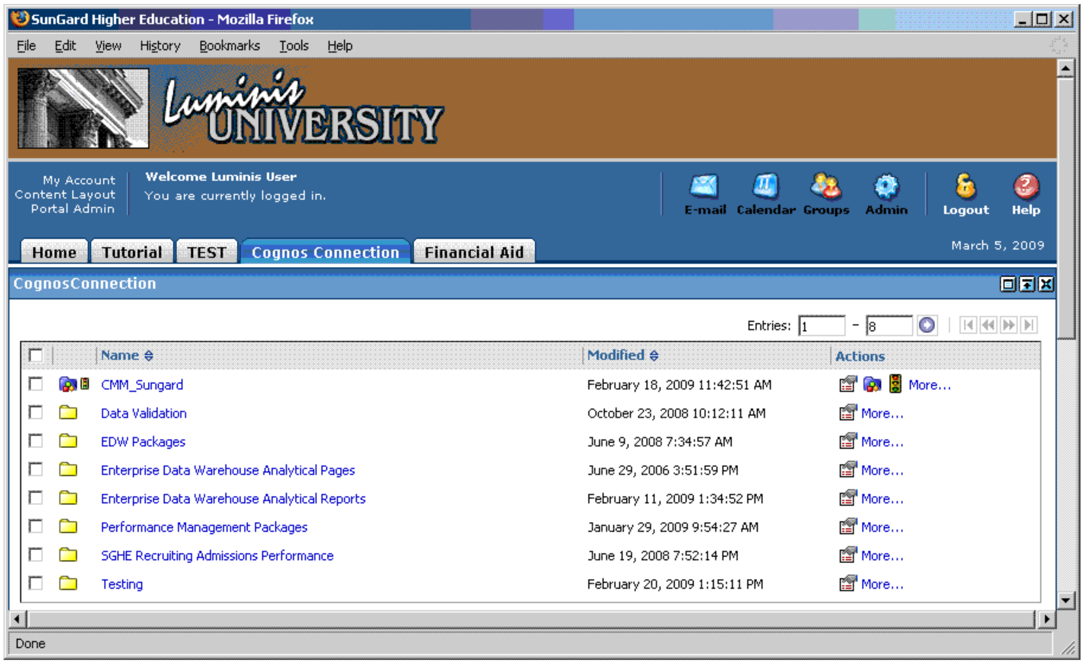
Note

Consider the use of the trailing “ui=” parameter in the URL above. Cognos supports using URL parameters to customize the appearance and functionality of the web pages displayed by the Cognos Connection/ Viewer interface. The “ui” parameter can take different values to display (or hide) various parts of the page. For example, ui=h1h2h3h4 will display all 4 header bars on a Cognos Connection page, whereas “ui=h1” would only display the first header bar. Similarly, the “frag-header” dparameter (=true/false) can be used to customize the appearance of Cognos dashboard reports displayed in Cognos connection. Following the technique described here, these values would get added to the URLs defined in the cognos.xml file, so they could then be referenced in Channel definitions.

For more information on using these parameters, see information about customizing the functionality of Cognos in the *Cognos Administration and Security Guide*. ■

Putting this into action, you can modify the Cognos Connection URL used earlier (defined in the cognos.xml as urlPass) as:

```
<SET a:symbol="urlPass" a:value=
"${properties.externalSystemURL}/${properties.cognosSystemID}/
cgi-bin/cognos.cgi?b_action=xts.run&amp;m=portal/
cc.xts&amp;gohome=&amp;ui=h1h3h4" />
```



Welcome Luminis User
You are currently logged in.

March 5, 2009

Entries: 1 - 8

□		Name	Modified	Actions
□		CMM_Sungard	February 18, 2009 11:42:51 AM	More...
□		Data Validation	October 23, 2008 10:12:11 AM	More...
□		EDW Packages	June 9, 2008 7:34:57 AM	More...
□		Enterprise Data Warehouse Analytical Pages	June 29, 2006 3:51:59 PM	More...
□		Enterprise Data Warehouse Analytical Reports	February 11, 2009 1:34:52 PM	More...
□		Performance Management Packages	January 29, 2009 9:54:27 AM	More...
□		SGHE Recruiting Admissions Performance	June 19, 2008 7:52:14 PM	More...
□		Testing	February 20, 2009 1:15:11 PM	More...

Note

Luminis can be configured to cache certain internal configuration data (such as channel definitions) so you may need to restart Luminis in order for channel definition changes to take effect. ■

For additional details on defining Luminis channels and UI elements, see the Luminis SDK/Channel Development Guide.

Transaction history tracking process

In Framework Manager, you can view and play back actions performed on the project. An action log is an XML file that contains a set of transactions. Each transaction has a sequence number and one or more actions. The action log file is stored in the project folder.

For example, you make changes to a project in a test environment. When it is time to move the project to production, you can use log files to play back every action, or series of actions, that you performed in the test environment to create an identical project in the production environment. Similarly, as an alternative to branching and merging projects one might want to track a series of customizations applied to a project to enable the identical customizations to be applied to an upgraded version of that model.

There are two action log files. The `log.xml` file contains all the transactions that have been run and saved in the project. This file is created the first time you save the project and exists until you delete the project. The temporary file contains transactions that have been run during the current session, but not saved. The temporary file is deleted when you close the project.

Note

Previously you had the option to use the Cognos Branch Merge functionality to retain your customizations. It is recommended using the Cognos Transaction History Tracking functionality instead to maintain institution-specific changes and upgrade your Framework Manager model. ■

View and save transaction history

You can view the transaction history in an action log file and then save it as a script.

1. From the **Project** menu, click **View Transaction History**.

Tip

To make the dialog box larger, double-click the caption. Double-click again to restore the dialog box to its original size. ■

2. Click the transaction numbers that you want.

Tip

To view the details of a transaction, click the plus sign (+) next to a transaction number. ■

3. Click **Save as Script**.
4. Type a name for the file.
5. Click **Save**. Do not save the file in the **logs** folder.
6. Click **Close**.

Play back transactions from a log file

You can choose to play back a specific transaction or a combination of transactions in a project or segment action log file.

When you play back transactions from a log file, the script player applies the commands in the log file to the contents of the existing model. Errors appear if objects created by the log file already exist in the model.

After the script in a log file has run successfully, a backup of the original project is created in the parent directory of the project. If you want to undo the transactions performed in the script, you can use the backup to restore the project to its original state.

You must disable or clear any commands that will conflict with the contents of the model. You can then run the script again.

1. From the **Project** menu, click **Run Script**.
2. Select the script you want, and click **Open**.
3. If you want to view the details of a transaction, click the transaction.
4. Set the starting or stop point that you want.
    - To set the starting point for running the script, select the script and then click **Set the starting point**. You can do this at any time to skip an instruction or run instructions that have already been executed .
    - To set a stop point for the script, select the script and then click **Set the stop point** .

      You can stop the script to make a manual fix and then start it again.
    - To remove the stop point, click **Remove the stop point**
5. Using the toolbar buttons, choose the run action that you want.

Buttons	Description
	Runs the script. After an error is encountered, clicking this button attempts to re-execute the failed instruction.
	Skips to the next transaction and runs the script to the end
	Runs the selected transaction only
	Skips to the next transaction and stops, but does not run any transactions

6. The project window is updated as the script is run.
7. Fix any errors encountered by the script either by retargeting objects or modifying the temporary project as required.
8. When the script has completed, click **Accept** to accept the changes or click **Revert** to undo the changes.

Note

After you click **Accept** or **Revert**, you cannot use **Undo** and **Redo** for the current session.

Brand Cognos Connection page

You can brand the Cognos Connection page to meet your institution's needs by customizing the banner, display text fonts, gradient, display color, and other format aspects.

To brand the Cognos Connection, perform the following steps.

1. Go to the `[install root]\cognos\c#\webcontent\skins` folder in your Cognos installation server (where # refers to the Cognos Release number you are currently running).

2. Modify the following style sheets and XML files to reflect the desired display settings:

   - fonts.css
   - default.css
   - banner.css
   - system.xml

   For detailed instructions on how to modify these files, refer to the *Customizing the IBM Cognos UI* document provided with Cognos.

Customize the welcome splash screen

You can customize the Welcome splash screen to reflect the desired look and feel.

To modify the default splash screen, perform the following steps.

1. Go to the `[install root]\cognos\c#\webcontent\skins` folder in your Cognos installation server (where # refers to the Cognos Release number you are currently running).

2. Replace the following image files with your branded files:

   - cognos_product_label.gif
   - portal_splash.gif

For additional information on customizing the splash screen, refer to the *Administration and Security Guide* document delivered with your Cognos product.