

DegreeWorks Banner Considerations Technical Guide

September 30, 2008
Release DW4.0.0



SUNGARD HIGHER EDUCATION

What can we help you achieve?

SunGard Higher Education

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

Customer Support Center website

<http://connect.sungardhe.com>

Documentation Feedback

<http://education.sungardhe.com/survey/documentation.html>

Distribution Services E-mail address

distserv@sungardhe.com

Other services

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

Trademark

Without limitation, SunGard, the SunGard logo, Banner, Campus Pipeline, Luminis, PowerCAMPUS, Matrix, Plus, ediLINK, TreQ, and DegreeWorks are trademarks or registered trademarks of SunGard Data Systems Inc. or its subsidiaries in the U.S. and other countries. Third-party names and marks referenced herein are trademarks or registered trademarks of their respective owners.

Notice of rights

Copyright © SunGard Higher Education 1999, 2008. This document is proprietary and confidential information of SunGard Higher Education Inc. and is not to be copied, reproduced, lent, displayed or distributed, nor used for any purpose other than that for which it is specifically provided without the express written permission of SunGard Higher Education Inc.



Think before you print.

Document Change Log

Version	Date	Change Description
DW4.0.0	September 30, 2008	Update UCX references to reflect new names and naming convention Source of Class Attribute from CFG020BANNER setting Extract uses config file Self Service updated Added Banner Database Issues section List of Database Table access updated
7.7.2.D02.P03	May 2008	Require access to SOBCACT table Selecting UCX tables for extract
7.7.2.D02.P02	March 2008	Added single ID option for bannerextract
7.7.2.D02.P01	February 2008	Correction for sorlcur_program Update and reformat Required Tables Added new View Audit Refresh D20 REFRESH flag and explanatory text; update D20 BANNER SureCode screen shot Added option to use .ids file when running bannerextract Added DGWCPUCOUNT section for bannerextract and RAD30JOB Added deleteid to bannerextract script
7.7.2.D02	December 14, 2007	Initial Version



Table of Contents



Document Change Log	3
Table of Contents	4
Banner Considerations	6
Banner Attributes	7
Banner Class Attributes	7
Banner Course Attributes.....	8
Banner Student Attributes	9
Banner Setup Checklist	10
UCX.....	10
Extract	11
Oracle	11
Scripts	11
Post extract.....	12
DegreeWorks Data Extract.....	13
Configuration and Installation	13
STEP 1 – Banner Database login.....	13
STEP 2 – Setup UCX-CFG020 BANNER.....	14
STEP 3 – Setup SQL select files	15
STEP 4 – Setup bannerextract.config file	16
Batch Data Extract Process	17
bannerextract.....	18
Student	18
Advisor.....	18
Staff	19
Deleting IDs	19
Selected UCX Tables	20
Other Modes	20
Transit – RAD30.....	21
DGWCPUCOUNT.....	22
Batch Extract Flow Diagram	23
Dynamic Data Extract Process	24
Banner Degree Coding Structure	26
Required Access to Banner	28
Database Table Access	28
Function Access.....	29
Banner Database Issues	30
Pointing to different Banner database.....	30
Creating Banner database link in DegreeWorks.....	31
Integration with Portals.....	32

Luminis.....	33
Single Sign-on for DegreeWorks	33
Instructions for integration of DegreeWorks with Luminis GCF.....	33
Overview of single sign-on process.	33
Installation Steps: Luminis system configuration.	34
Installation Steps: Luminis SSO configuration files.....	35
Installation Steps: DegreeWorks server.....	36
Luminis Channel and User Configuration.	36
User Guide.....	37
Load DegreeWorks credentials into Luminis LDAP.	37
Self Service Banner.....	39
Single Sign-on for DegreeWorks	39
Integration of DegreeWorks with Self Service Banner	39
Installation Steps: Self-Service Banner web menu setup.	40
Installation Steps: DegreeWorks server.....	42
Student ID Pass-along to DegreeWorks	43
Overview.....	43
How this works.....	43
User Role Pass-along to DegreeWorks	44
Overview.....	44
How this works.....	44



Banner Considerations



Banner Considerations

To help you use DegreeWorks effectively, there are a variety of special topics that need to be discussed and elaborated. It is assumed that the reader is familiar with the UCX and how it is maintained.

Banner Attributes

Banner Class Attributes

Banner “class attributes” are codes that are not part of the standard database tables passed for class records (current, historic and transfer) from the student system to DegreeWorks. These class attributes are stored in the following Banner database tables and are retrieved using different pieces of a student’s class data:

- SSRATTR - current classes by Crn and Term
- SHRATTR - historic classes by PIDM, Sequence Numbers and Effective Term
- SHRATTC - historic classes by Crn and Effective Term
- SHRTATT - transfer classes by PIDM and Sequence Numbers

Historic class attributes are stored in two tables: SHRATTR and SHRATTC. A UCX-CFG020 BANNER flag “Always Process SHRATTC” controls whether attributes from SHRATTC are extracted if attributes from both tables exist. SHRATTR class attributes are processed first. If any SHRATTR attributes are found for a given class they are written to the rad_attr_dtl. Then the UCX-CFG020 BANNER “Always Process SHRATTC” flag is checked:

- ‘N’ - the SHRATTC attributes will be skipped for a given class if attributes from SHRATTR already exist for that class. This is the default value if this flag is left BLANK.
- ‘Y’ - the SHRATTC class attributes will always be processed and written to the rad_attr_dtl if found for a given class, even if SHRATTR attributes exist for that class. In this case class attributes from both the SHRATTR and SHRATTC Banner tables would be written to the rad_attr_dtl.

These values are available for use with the “WITH” keyword in Scribe and a single entry of “ATTRIBUTE” must be defined in UCX-SCR044. Class Attributes may be used where a requirement varies based on class data. For example, classes in the “Honors” program might have an “HONR” Class Attribute assigned to each “Honors” class. If a student is in the “Honors” program and must complete 5 credits in Honors English, then the requirement could be written as follows:

```
5 Credits in ENGL @ (With Attribute = HONR)
```

The above requirement will only work properly if the “ATTRIBUTE” is added to UCX-SCR044 with the “ATTR” Element assigned. The “ATTR” tells DegreeWorks that this item is defined in the rad_attr_dtl. Be sure to set the Offset and Length fields to “00”. See the UCX-SCR044 screen below.

SCR044

KEY: ATTRIBUTE

◀ ▶ ✖ 📄 ?

Standard WITH codes not needed in SCR044:
 DWTitle, DWTerm, DWSection, DWLocation, DWGrade, DWGradeType
 DWCredits, DWResident, DWTransfer, DWCreditType, DWPassFail

**Attributes bridged to the Rad-Attr-Dtl need a record here but
 the Element field should be ATTR (Offset and Length can be 00).**

Description: Banner attributes

Element: ATTR Must be valid element number in RAD-Class-Dtl or be "ATTR"

UCX Table:

Offset: 00 Starting position in WITH buffer to place this value

Length: 00 Number of bytes this value will consume in the WITH buffer

Banner Course Attributes

Banner “course attributes” are codes that are not part of the standard database tables passed for course records from the student system to DegreeWorks. These course attributes are stored in the **SCRATTR** Banner database table and are retrieved when courses are being pulled into DegreeWorks. These attributes are stored in the rad-crs-attr-dtl linked from the rad-course-mst by the course key. When a planner audit is performed the attributes associated with each course in the plan is sent to the auditor in case they are needed to satisfy requirements using “WITH Attribute=”.

Banner Student Attributes

Banner “student attributes” are codes that are not part of the standard database tables passed for class records (current, historic and transfer) from the student system to DegreeWorks. These student attributes are stored in the **SGRSATT** Banner database table. These attributes are retrieved using the student’s PIDM. These values are available for use in If-statements in Scribe and may be used to control what appears in the student header on the audit worksheets. Student Attributes may be used where a requirement varies based on the presence or absence of an attribute. For example, students in the Honors program might have to take an additional set of classes. The Student Attribute code of “HONR”, for example, will be pulled from Banner into DegreeWorks and will be used to control what requirements the student must meet. Such a requirement might be written as follows:

```
If (Attribute = HONR) then  
  15 Credits in ENGL 4@ Label "15 upper-division credits required for honor students";
```

All Banner Student Attributes are placed into the rad-custom-dtl in DegreeWorks with a custom-code of “ATTRIBUTE” and a custom-value of the attribute code, such as “HONR” for example.

UCX-SCR002 must contain an ATTRIBUTE entry telling DegreeWorks to retrieve all rad-custom-dtl ATTRIBUTE records and send them to the auditor.

KEY:

◀ ◁ ▷ ▶ ✖ 📁 🖨️ ?

**REMINDER: To enable your UCX-SCR002 changes be sure to restart the web jobs.
When using SSGPA or BannerGPA you do not need an entry here.**

Description:

Data Element: Specify the datum to be used in a Scribe IF statement

UCX Table:

Edit Element 1: If Data is on a detail this element is used as a filter

Type: AT / EV / SC / HD / DG / DI / CS

Value: Use ACTV if Type is AT - "Active Term"

Banner Setup Checklist

UCX

Review and change UCX settings using SureCode (see SureCode documentation).

UCX	Action
UCX-SCR001 STATUS	change Description to “Student Type”
UCX-SCR001 SCHOOL	change Description to “Level”
UCX-SCR001 LEVEL	change Description to “Student Class Level”
UCX-CFG020 BANNER	Be sure the Banner Site flag is Y. You can turn on the Search in Banner flag now but it is suggested that you perform all searches against the bridged data at first while getting everything setup. Leave this flag as ‘N’ for now but switch it to ‘Y’ later in your implementation. See the documentation regarding the other flags on this record.
UCX-CFG020 REFRESH	Set the flags to appropriately configure the refresh parameters
UCX-CFG020 SEARCH	Set the Show flags to ‘N’ for items you are not using. Show Specialization and Show Liberal Learning should be ‘N’ since these are not used in Banner. Note - School in DegreeWorks is actually the same as Level in Banner. See the documentation regarding the other flags on this record.
UCX-STU352 DISCIPLINE STATUS	Set the Discipline Status flag to ‘I’ for those disciplines that are Inactive. If a Discipline is Inactive then the class (current, historic and transfer) will be skipped and NOT bridged to the DegreeWorks rad_class_dtl.
UCX-STU385 IN PROGRESS Flag	The default In-Progress flag is ‘N’ for historical classes found in SHRTCKN. If an historical grade combination in UCX-STU385 (key = School/GradeType/Grade) is considered ‘In-Progress’ then set this In-Progress flag to ‘Y’ so that the rad_inprog_flag on the rad_class_dtl gets built appropriately with a ‘Y’ value.
UCX-STU385 OVERRIDE Flags	There is a “master” Override flag and 8 Override values. Refer to the Technical Guide UCX documentation on details on each of these Override fields is used.
UCX-STU385 Transfer Repeat Flags	There are three Transfer Repeat fields that may be used to override the standard Transfer Repeat Pointer and Repeat Policy used for repeated transfer classes. Please refer to the Technical Guide UCX documentation for details on how to load these fields.
UCX-BAN080 Custom Data	For special “if” statements and for any special data desired on the worksheet setup UCX-BAN080. Please refer to the Technical Guide UCX documentation for details on how to setup UCX-BAN080.

Extract

Selecting People

Modify these files as needed to select the desired students, advisors and staff:

```
local/sql/bannerstudents.sql  
local/sql/banneradvisors.sql  
local/sql/bannerstaff.sql
```

Deleting People

Modify this file as needed if ID codes are to be deleted from DegreeWorks:

```
local/sql/bannerdeleteids.sql
```

Selecting UCX Entries If selected UCX tables are to be re-extracted from Banner to DegreeWorks, create a UCX file containing a list of DegreeWorks tables (one table per line) with an “.ids” extension in the local/sql directory. The actual file name can be any valid Unix file name. However, it **MUST** have the “.ids” extension. For example, the file “banucx.ids” may be created: local/sql/banucx.ids. The banner extract command would be:

```
bannerextract ucx banucx.ids
```

If this file is found then only the DegreeWorks tables listed in that file will be re-loaded with Banner data. Make sure to check the setting for the UCX-CFG020 BANNER “Add UCX Entries Only” flag. Make sure it is set to “N” if ALL entries are to be reloaded from Banner. If only NEW entries are to be loaded from Banner make sure this UCX-CFG020 flag is set to “Y”.

Selecting Different Records for Your Institution in the Banner Extracts

Modify the \$ADMIN_HOME/common/bannerextract.config file as needed to select a different set of records than those DegreeWorks is selecting by default. Make sure to review this file and make all appropriate changes for your site.

NOTE: ALL of the Banner tables used by the banner extracts (ban40-ban47 and dap58) are now included in this configuration file.

Oracle

Install/setup Banner views, etc.

Scripts

Setup “bannerextract” script in cron (see Technical documentation).

```
bannerextract student  
bannerextract advisor  
bannerextract staff  
bannerextract course  
bannerextract ucx  
bannerextract equiv  
bannerextract ets
```

Post extract

After UCX has been copied from Banner review each of these tables and setup as needed.
Be sure not to run UCX extract again:

Table	Fields to setup
UCX-AUD027	Filter fields
UCX-STU016	Planner flag
UCX-STU035	Planner flag
UCX-STU307	Short degree field
UCX-STU352	Discipline Status ("I" – Inactive)
UCX-STU385	GPA Calc flag
UCX-STU385	In-Progress flag
UCX-STU385	Override flags
UCX-STU385	Override Transfer Repeat flag/fields

DegreeWorks Data Extract

Configuration and Installation

The DegreeWorks Data Extract process for Banner clients was developed using Oracle's Pro*C embedded SQL tools. These programs execute on the DegreeWorks application server – Linux, UNIX, etc. SunGard has developed several extract programs to not only extract the required data but to also convert the data to the Bridge Interface Format (BIF) so that it is ready to load into DegreeWorks.

Verify or perform the following steps to configure your DegreeWorks Server to perform the Banner DegreeWorks Data Extract:

STEP 1 – Banner Database login

Be sure the DB_LOGIN_BANNER environment variable is set correctly:

```
$ echo $DB_LOGIN_BANNER
```

If this does not show the correct Banner database login you need to edit this variable in dw.start.config and log back into your host session. Here are examples of how to set this variable:

```
export DB_LOGIN_BANNER=userid/userpwd
```

```
export DB\_LOGIN\_BANNER=userid/userpwd@some.other.machine.edu
```

If the database is on a remote machine you may use the "@" option to specify where the database resides.

Be sure the ORACLE_SID environment variable is set correctly:

```
$ echo $ORACLE_SID
```

If this is not set correctly review your setup of dw.start.config – see config_SetupMenuItems.

STEP 2 – Setup UCX-CFG020 BANNER

CFG020

KEY:

Banner Site Y=Y=Banner Site - bannerdb exists

Email Code Email code used to find desired GOREMAL record

Email Override Y=Y=If Email code not found, then use Active Email

Repeat Skip A N Y=Skip Averaged classes

Repeat Policy A 1-6, B or BLANK, Repeat Policy if Averaged classes included

Repeat Skip E N Y=Skip Excluded classes

Repeat Policy E 0-6, B or BLANK, Repeat Policy if Excluded classes included

Repeat Policy I 1-6, B or BLANK, Repeat Policy for Included classes

GPA Type I=Institutional, O=Overall

Add UCX Entries Only N Y=Only ADD UCX entries, NO deletes

Use Term as Catalog Year N Y=Load UCX-STU035 with UCX-STU016 Term Codes

Search in Banner N Y=In web search, search in Banner, N=Search in DegreeWorks

Advisor User Class ADV=Can make exceptions, ADVX=Cannot make exceptions

Staff User Class Standard value is REG

Non Course Score C=NCST Code, I=STVNCST Indicator

Default Current Grade Used on all Current classes

Transfer Repeat Skip A N Y=Skip Averaged classes

Transfer Repeat Policy A 1-6, B or BLANK, Repeat Policy if Averaged classes included

Transfer Repeat Skip E N Y=Skip Excluded classes

Transfer Repeat Policy E 0-6, B or BLANK, Repeat Policy if Excluded classes included

Transfer Repeat Policy I 1-6, B or BLANK, Repeat Policy for Included classes

Advisor Method A=Load 4 Advr Codes, C=Match on Major and Minor Codes, S=Sequential

Advisor Major SGRADVR_CODE that identifies a Major Advisor if Method "C" used

Advisor Minor SGRADVR_CODE that identifies a Minor Advisor if Method "C" used

Load Extra Advisors Y Y=Load if Method "C" and NO SGRADVR_CODE match

Cross List in SCREQIV Y Y=Skip Equivalent if SCBCRKY_TERM_CODE_END all 9's

Repeatable Option N=Limits not checked, L=Repeat Limit, U=Max Rpt Units, B=Both, I=Include Course

Current Course A=STVCSTA_ACTIVE_IND of 'A'; C=SCBCRSE_CSTA_CODE of 'C'; K=SCBCRKY End Term

Follow Gradable Ind N Y=Skip current class if GradableInd = 'N'

Method "A" Advisor Code1 1 SGRADVR_ADVR_CODE to be loaded into RAD Advr1
 Code1 2
 Code1 3

Advisor Code2 1 SGRADVR_ADVR_CODE to be loaded into RAD Advr2
 Code2 2
 Code2 3

Advisor Code3 1 SGRADVR_ADVR_CODE to be loaded into RAD Advr3
 Code3 2
 Code3 3

Advisor Code4 1 SGRADVR_ADVR_CODE to be loaded into RAD Advr4
 Code4 2
 Code4 3

Check Dual Degrees Y Y=Check SGBSTDN Dual Degree fields

Always Process SHRATTC Y=Include SHRATTC attributes even if SHRATTR found

Review all settings before running the extract.

Repeat Policy

Although you may use repeat policies 1-6 telling DegreeWorks to use the class with the best grade or the most recently taken class etc it is recommended that you set the Repeat Policy flags to 'B' for all Repeat Indicators.

When 'B' is used this behavior will occur:

- **Excluded** classes will end up in the Insufficient section of the audit but they will not affect the overall GPA or credits.
- **Averaged** classes will end up in the Insufficient section of the audit and they will affect the overall GPA but will not be counted in the overall credits towards degree.
- **Included** classes will apply to rules as normal classes affecting the GPA and total credits.

By using the 'B' setting you are telling DegreeWorks to handle each class based on the indicator without regard to grades or terms taken as the decision about which classes should be counted and not counted has already been made and recorded using the indicator. When 'B' is in use the normal DegreeWorks repeat logic is skipped simplifying the auditing process greatly.

STEP 3 – Setup SQL select files

Review, modify and/or create the “.sql” files in the local/sql directory using your DegreeWorks user *before* launching the RAD30 processor or the bannerextract script. It is recommended that you copy the sql statement into sqlplus and make sure it executes properly. Also, make sure that the number of ID codes selected by the sql statement is what is expected and that they are the correct ID codes.

Note: If a Banner table is used in one of the sql files identified below that is *not* found in the “Required Access to Banner Tables” list located at the end of this document make sure to have your database administrator add the appropriate access. Otherwise the ID codes will not be extracted correctly and the banner extract will fail.

Review and modify as needed the file used to select the **students** you want bridged to DegreeWorks:

local/sql/bannerstudents.sql

Review and modify as needed the file used to select the **advisors** you want bridged to DegreeWorks:

local/sql/banneradvisors.sql

Review and modify as needed the file used to select the other **staff** you want bridged to DegreeWorks:

local/sql/bannerstaff.sql

Review and modify as needed the file used to delete unwanted **ID** codes you would like removed from DegreeWorks:

local/sql/bannerdeleteids.sql

STEP 4 – Setup bannerextract.config file

Review and modify as needed the **\$ADMIN_HOME/common/bannerextract.config** file to select a different set of records based on your particular needs.

The config file should look like the example you see here – the FROM/WHERE text for your school may need to be changed.

```
# SGBSTDN must be a; AND is required at the end of the WHERE
SGBSTDN-from: FROM SGBSTDN a
SGBSTDN-where: WHERE a.SGBSTDN_TERM_CODE_EFF =
SGBSTDN-where: (SELECT MAX(b.SGBSTDN_TERM_CODE_EFF)
SGBSTDN-where: FROM SGBSTDN b WHERE b.SGBSTDN_PIDM = a.SGBSTDN_PIDM) AND
# a.SGBSTDN_PIDM = <students-pidm>
```

A special set of records with keys of “CALCFCN-from:” and “CALCFCN-where:” have been created for the special Banner function: “F_CLASS_CALC_FCEN”. This function call is made by the banner student extract using the PIDM, LEVL_CODE and TERM_CODE specified in this bannerextract.config file to generate the student’s Class Standing code that is loaded into the rad_stu_level on the rad_degree_dtl. A default set of special records with a key of “CALCFCN” are included in this configuration file and are loaded with the FROM and WHERE clauses from the SGBSTDN default entry. Change this set of “CALCFCN” records as appropriate for your site.

Batch Data Extract Process

Student academic data is required for DegreeWorks to generate a Degree Audit. SunGard extracts student data, advisor information, your course catalog and the list of transfer institutions from the Banner database and stores this data in the Repository for Audit Data (RAD) database tables. This procedure is known as the DegreeWorks Bridge process and consists of the following data extraction processes:

Students – active students’ academic and basic biographic data are extracted based on a specified academic term. Student data can also be extracted individually by the SPRIDEN ID.

Advisors – access records are created for advisors who require access to DegreeWorks

Staff – access records are created for staff who require access to DegreeWorks

Course Catalog – all current courses from your course catalog

Course Equivalents – historic courses and their current equivalent courses

ETS – Transfer institution ETS codes, names and identification data

UCX Validation Codes – Validation tables consisting of data from Banner STV tables

The batch extract programs are executed one of two ways: a script named **bannerextract** or via Transit. When extracting students DegreeWorks automatically runs a new degree audit for each of the students that have changed data since the last time they were extracted. In doing this you will be sure that each student’s degree audit reflects any changes made to their student record.

bannerextract

This script can be used to schedule the extract to run using **cron** or **at** or it can be run directly at any time during the day as needed.

Warning: Be sure you are not in the local/sql directory when running bannerextract.

Student

To run the **student** extract using bannerstudents.sql file in the local/sql directory:

```
$ bannerextract student
```

You may also specify a different sql file in the local/sql directory:

```
$ bannerextract student somestudents.sql
```

You may also specify a file of student IDs in the local/sql, admin/data or current directory:

```
$ bannerextract student somestudents.ids
```

For testing purposes you may also supply a single student ID instead of a file name:

```
$ bannerextract student 1234567
```

Advisor

To run the **advisor** extract using banneradvisors.sql file in the local/sql directory:

```
$ bannerextract advisor
```

You may also specify a different sql file in the local/sql directory:

```
$ bannerextract advisor someadvisors.sql
```

You may also specify a file of advisor IDs in the local/sql, admin/data or current directory:

```
$ bannerextract advisor someadvisor.ids
```

You may also supply a single advisor ID instead of a file name:

```
$ bannerextract advisor 1234567
```

Staff

To run the **staff** extract using bannerstaff.sql file in the local/sql directory:

```
$ bannerextract staff
```

You may also specify a different sql file in the local/sql directory:

```
$ bannerextract staff somestaff.sql
```

You may also specify a file of staff IDs in the local/sql, admin/data or current directory:

```
$ bannerextract staff somestaff.ids
```

You may also supply a single staff ID instead of a file name:

```
$ bannerextract staff 1234567
```

Deleting IDs

To run **deleteid** to remove unwanted ID codes from the DegreeWorks database using bannerdeleteids.sql file in the local/sql directory (**do NOT use with cron**):

```
$ bannerextract deleteid
```

You may also specify a different sql file in the local/sql directory:

```
$ bannerextract deleteid somedeletes.sql
```

You may also specify a file of IDs in the local/sql, admin/data or current directory:

```
$ bannerextract deleteid somedeletes.ids
```

Selected UCX Tables

To run the **ucx** extract using a list of DegreeWorks UCX tables listed in a file in the local/sql directory with an “.ids” extension (**do NOT use with cron**):

```
$ bannerextract ucx someucxtables.ids
```

For example, the “local/sql/someucxtables.ids” file might contain tables:

```
UCX-STU352  
UCX-STU560  
UCX-STU563
```

In this case **ONLY** these 3 UCX tables will be re-extracted from Banner.

Note: The UCX-STU563 table (Concentrations) is re-created from the STVMAJR Banner table which also is used to recreate major tables (UCX-STU023 and UCX-AUD027) and minor tables (UCX-STU024 and UCX-AUD029). However, *only* UCX-STU563 will be extracted with this banner extract.

Make sure to check the setting for the UCX-CFG020 BANNER “Add UCX Entries Only” flag **BEFORE** running the UCX bannerextract. Make sure it is set to “N” if **ALL** entries are to be reloaded from Banner. If only **NEW** entries are to be loaded from Banner make sure this UCX-CFG020 flag is set to “Y”.

In the above example, UCX-STU352 is being re-extracted. It has a “Discipline Status” flag in it which is manually input using SureCode. These updates will need to be made again if the entire UCX-STU352 is re-extracted.

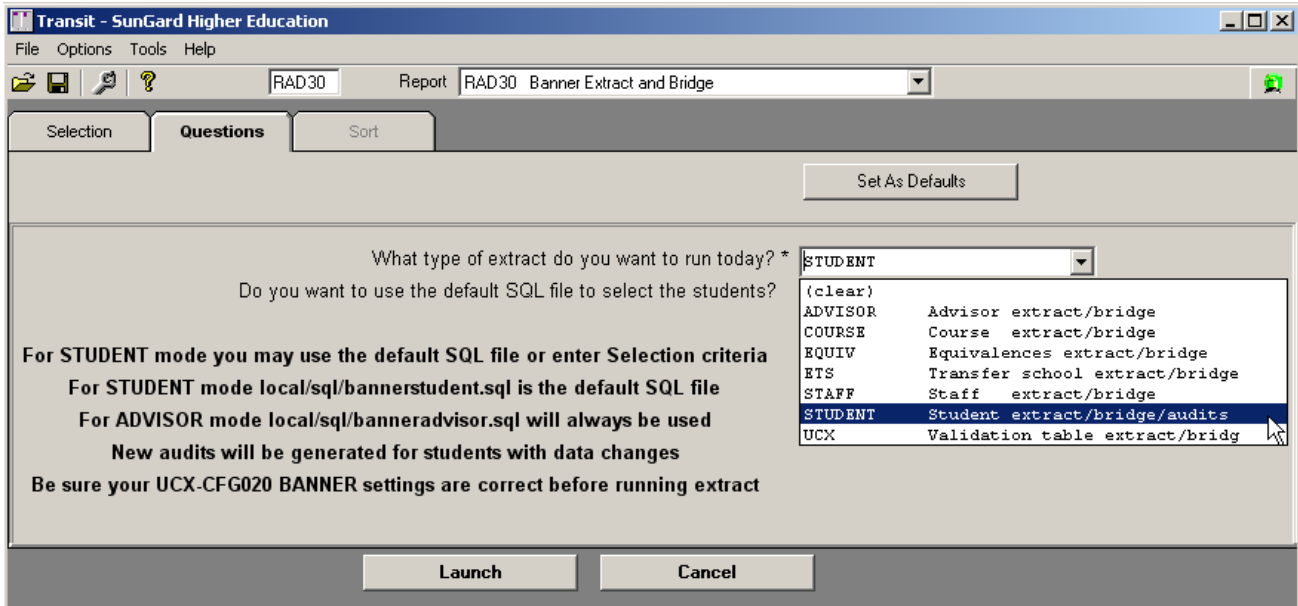
Other Modes

The other extract modes do not have associated sql files – these modes extract **ALL** records from the Banner database and load them into the associated DegreeWorks database tables:

```
$ bannerextract course  
$ bannerextract ucx  
$ bannerextract equiv  
$ bannerextract ets
```

Transit – RAD30

You may use the RAD30 processor in Transit to run the Banner extract. When running the STUDENT extract you may use the default .sql file or you may use the Selection tab to choose the pool of students you want extracted. Your user logon must be given the PTSBANPR key in order for RAD30 to show up as an option for you in Transit - this key can be added to your user using SHPCFG.



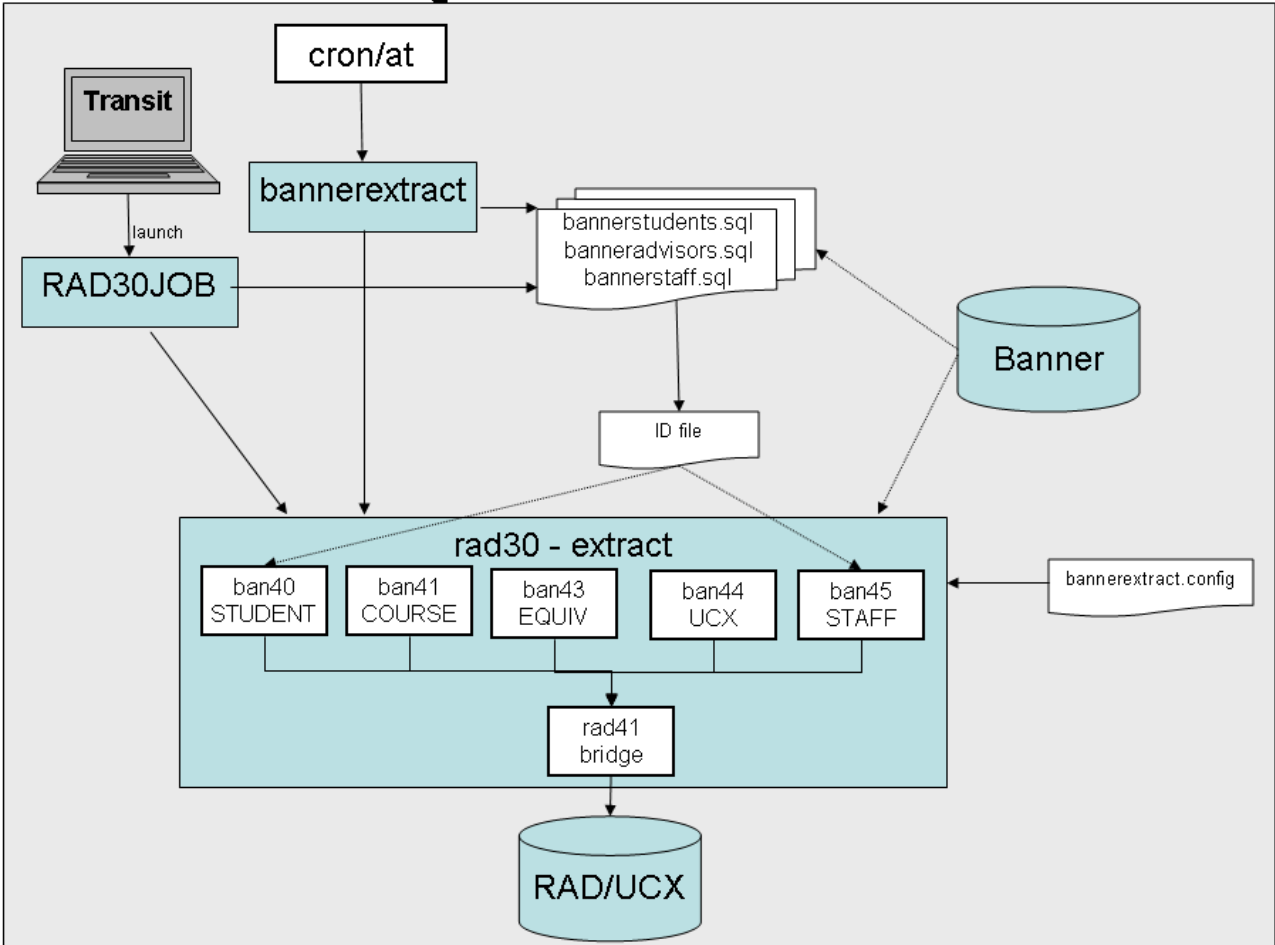
DGWCPUCOUNT

Both the bannerextract script and the RAD30JOB script will use the DGWCPUCOUNT environment variable to split up the file of student, advisor or staff IDs into multiple files. Once multiple ID files are created, the scripts are then able to launch multiple processes on each of the files. In theory, with DGWCPUCOUNT set to 4 the task gets done in a quarter of the time.

See dw.start.config to change your DGWCPUCOUNT setting.

Batch Extract Flow Diagram

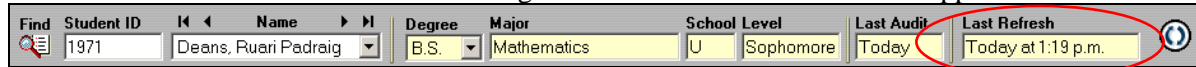
DegreeWorks Banner Extract



Dynamic Data Extract Process

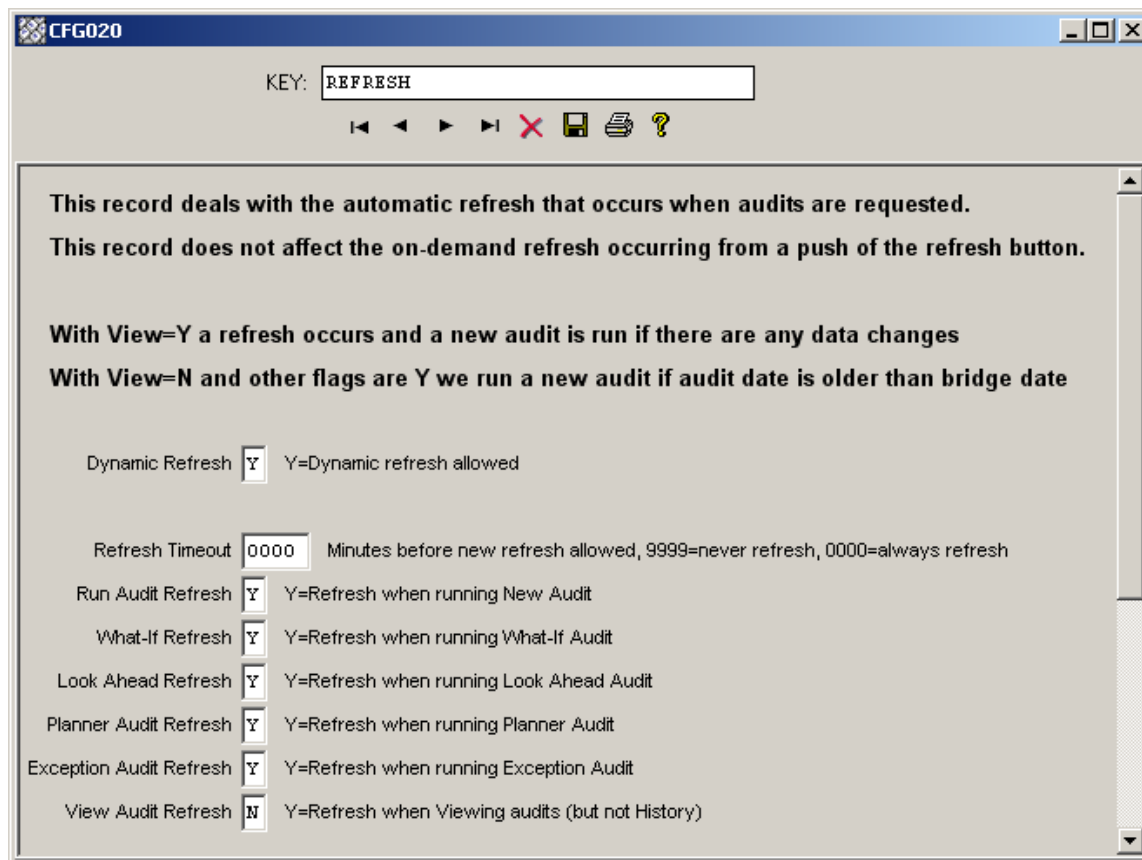
In addition to extracting student data via the batch extract you may extract data for a single student using the dynamic method – there are two ways in which a dynamic extract may take place: the first is available from within the DegreeWorks web application as a button, and the second occurs whenever a triggering event takes place.

The first way of performing a dynamic extract from Banner is to push a button. Users with the SDREFRES key will be shown the Last Refresh field in the student context area at the top of the main web page. When the UCX-CFG020 BANNER “Banner Site” flag is Y the Refresh button will also appear for these users.



The Last Refresh date and time indicate the last time the student’s academic data was copied from Banner into DegreeWorks. At any time users may click the on-demand refresh button to once again pull in this student’s data. Users may want to do this after a major or grade change was made in Banner and they don’t want to wait for the nightly batch extract. Once the extract process completes the user will get a confirmation message and the Last Refresh date/time will be updated.

The second way of performing a dynamic extract involves a triggering event, which may occur when any user requests an audit from the Worksheets, What If, Planner, Exceptions or Look Ahead tabs. The UCX-CFG020 REFRESH record in SureCode is used to control this behavior.



Running Audits

Turning on the Dynamic Refresh with all or some of the other flags turned on will affect performance – both for all users of the system and the particular users attempting to run an audit. Before DegreeWorks runs the requested audit it refreshes the student's academic data from Banner – this takes extra time and will consume additional system resources. When a refresh occurs through running an audit the user is not notified – but the Last Refresh date/time is updated.

The Refresh Timeout is a mechanism the client may use to inhibit repetitive dynamic refreshes for students that are incidental and not necessary, for instance when multiple What-Ifs are being launched within a short time span. The client can specify a length of time during which no additional refreshes of data would be appropriate. This timeout setting is used in conjunction with the refresh date/time stored on the rad-primary-mst to tell DegreeWorks if it should re-read the student's data in Banner.

As with the batch extract process, the ban40/banstudent and rad41/radbridge routines are used to perform the dynamic refresh. The rad41/radbridge does check for changed data and will skip the insertion of records if no changes exist – but will update the refresh date/time on the rad-primary-mst and that shown on the web page either way.

Viewing Audits With the View Audit Refresh flag set to Y the system will check the timeout setting as with running an audit and will execute the extract as needed. If there are data changes a new audit will be run. In addition, the audit and refresh date/times will be updated in the student context area on the web page reflecting what has just occurred. When an extract is processed while in View mode the bridge date/time are not updated if there are no real data changes – this is different from when in Run mode.

When the View Audit Refresh flag is N and one of the other refresh flags is Y the system compares the extract/bridged date/time to that on the most recent audit for the given school/degree. If it is determined that the audit is stale a new audit will be processed.

Banner Degree Coding Structure

For each student, DegreeWorks creates a degree record for each active SORLCUR record it finds with a unique sorlcur_degc_code. One or more SORLFOS records is linked back to the associated SORLCUR(s) by the lcur_seq_no. Each SORLFOS major/minor/concentration is placed on this degree record. Each degree record built in DegreeWorks will result in a discrete degree audit.

Note that DegreeWorks does not utilize the sorlcur_program field to identify the student's degree-major combination. In DegreeWorks, the degree is taken from sorlcur_degc_code and the major is taken from the associated SORLFOS record(s). The sorlcur_program is stored separately, and can be referenced with Scribe to differentiate between degree-major combinations.

Example A

one degree - two majors	
Definition of degree, and location of data, in Banner	Results in DegreeWorks
SORLCUR - sorlcur_degc_code =BS; seq-no=1 SORLFOS - sorlfos_majr_code =CHEM; lcur_seq_no=1 SORLFOS - sorlfos_majr_code =BIOL; lcur_seq_no=1	This will result in one degree record in DegreeWorks - a BS degree with two majors. The degree audit will be run against this degree with two majors sharing or not sharing classes based on the requirements.

Example B

two degrees of different type (same level) - two majors	
Definition of degree, and location of data, in Banner	Results in DegreeWorks
SORLCUR - sorlcur_degc_code=BS; seq-no=1 SORLFOS - sorlfos_majr_code =MATH; lcur_seq_no=1 SORLCUR - sorlcur_degc_code =BS; seq-no=2 SORLFOS - sorlfos_majr_code =PHYS; lcur_seq_no=2	This will result in one degree record in DegreeWorks - a BS degree with two majors. The degree audit will be run against this degree with two majors sharing or not sharing classes based on the requirements.

Example C

two different degrees (same level) - two majors	
Definition of degree, and location of data, in Banner	Results in DegreeWorks
SORLCUR - sorlcur_degc_code =BS; seq-no=1 SORLFOS - sorlfos_majr_code =CHEM; lcur_seq_no=1 SORLCUR - sorlcur_degc_code =BA; seq-no=2 SORLFOS - sorlfos_majr_code =ARTH; lcur_seq_no=2	This will result in two degree records in DegreeWorks - a BS degree with a CHEM major and a BA degree with an ARTH major. Two discrete degree audits will be produced - all classes will be applied to both sets of degrees/majors regardless of sharing policies.

Example D

two degrees (different level) – two majors

Definition of degree, and location of data, in Banner	Results in DegreeWorks
SORLCUR - sorlcur_degc_code =BS; level=UG; seq-no=1 SORLFOS - sorlfos_majr_code =CHEM; lcur_seq_no=1 SORLCUR - sorlcur_degc_code =MA; level=GR; seq-no=2 SORLFOS - sorlfos_majr_code =PHIL; lcur_seq_no=2	This will result in two degree records in DegreeWorks - a BS degree with a CHEM major and an MA degree with a PHIL major. Two discrete degree audits will be produced. The undergraduate classes will be applied to the BS degree/major and the graduate classes will be applied to the MA/PHIL degree/major. (Note - you may change the configuration flag to allow all classes to apply to both degrees thus ignoring the level filter that is in place by default.)

Example E

two different BS degrees (same level) - two majors – Not recommended

Definition of degree, and location of data, in Banner	Results in DegreeWorks
SORLCUR - sorlcur_degc_code=BSMATH; seq-no=1 SORLFOS - sorlfos_majr_code =MATH; lcur_seq_no=1 SORLCUR - sorlcur_degc_code =BSPHYS; seq-no=2 SORLFOS - sorlfos_majr_code =PHYS; lcur_seq_no=2	This will result in two degree records in DegreeWorks - a BSMATH degree with a MATH major and a BSPHYS degree with a PHYS major. Two discrete degree audits will be produced - all classes will be applied to both sets of degrees/majors regardless of sharing policies. This is not a recommended approach. It is best to stick with Example A and link both majors to a single BS degree record. Students normally cannot double-count all classes between the two majors and the only way to prevent the double-counting is to link them to the same degree.

If concurrent curriculum is not used for the student being processed, then this data will be extracted from the fixed columns found on the SGBSTDN table.

Required Access to Banner

Database Table Access

Access to your Banner database is required for the extract process, therefore read access must be provided to the following tables:

Banner Table	Course Catalog	Course Equivalen	ETS (transfer)	Advisors Staff	Students	UCX	Banner General	Description
GOREMAL				X	X			Email Address Data
SCBCRKY		X						Course Start/End Dates
SCBCRSE	X	X						Course Master Data
SCRATTR	X							Course Attributes
SCREQIV		X						Course Equivalents
SCRRTST	X							Course Prerequisites
SFRSTCR					X			Current Class Data
SGBSTDN					X			General Student Degree Data
SGRADVR				X	X			Advisor Data
SGRSATT					X			Student Attributes by PIDM and Current Term
SHRATTC					X			Student Attributes by CRN and Historic Term
SHRATTR					X			Student Attributes by PIDM and Historic Sequence Number
SHRGRDE					X			Grade Codes (UCX-STU385)
SHRGRDO					X			Grade Codes – valid combo of level/gmod/grade (UCX-STU385)
SHRLGPA					X			Summary GPA/Credits Data
SHRNCRS					X			Non Course Data
SHRQPNM					X			Non Course Data – Papers and Exams
SHRTATT					X			Student Attributes by PIDM and Transfer Sequence Number
SHRTCKG					X			Historic Grade Values
SHRTCKL					X			Historic Level (School) Data
SHRTCKN					X			Historic Class Data
SHRTGPA					X			Detail GPA/Credits by Term
SHRTRCE					X			Transfer Equivalent Data
SHRTRCR					X			Transfer Class Data
SHRTRIT					X			Transfer School Data
SOBCACT					X			SORLCUR active indicator validation
SOBSBGI			X					ETS Address Data
SORBTAG			X					ETS Calendar Data
SORDEGR					X			Previous Degree Data
SORLCUR					X			Concurrent Degree Data
SORLFOS					X			Field of Study Data
SORTEST					X			Test Score Data
SPRHOLD					X			Student Holds
SPRIDEN				X	X			Primary Name Data
SSBSECT					X			Schedule Master Data
SSRATTR					X			Student Attributes by CRN and Current Term

Banner Table	Course Catalog	Course Equivalen	ETS (transfer)	Advisors Staff	Students	UCX	Banner General	Description
STVACYR		X				X		Catalog Year Codes (UCX-STU035)
STVASTD					X			Academic Standing Codes
STVCLAS						X		Student Level Codes (UCX-STU305)
STVCOLL						X		College Codes (UCX-STU560)
STVCSTA	X	X						Course Status Codes
STVDEGC						X		Degree Codes (UCX-STU307)
STVGMOD						X		Grade Types (UCX-STU356)
STVHLDD					X			Hold Registration Codes
STVLEVL						X		School Codes (UCX-STU350)
STVMAJR						X		Major Codes (UCX-STU023), Minor Codes (UCX-STU024), Concentration Codes (UCX-STU563)
STVNATN			X					ETS Foreign Country Codes
STVNCRQ					X			Non Course Codes
STVNCST					X			Non Course Status Codes
STVQPTP					X			Non Course Exam/Paper Codes
STVRSTS					X			Course Registration Status
STVSBGI			X		X	X		ETS School Names
STVSTST					X			Student Status (DW student type)
STVSTYP						X		Student Type (DW Student Status Codes UCX-STU306)
STVSUBJ						X		Discipline Codes (UCX-STU352)
STVTERM	X	X			X	X		Term Codes (UCX-STU016)
SURVERS							X	Version of Oracle being used
TWBKBSF							X	Banner Self Service
TWGBWSES							X	Banner Self Service

Function Access

The `f_class_calc_fnc` function must be made available so that student class levels can be calculated.

Banner Database Issues

Pointing to different Banner database

Following are the steps to take in order to point to a different Banner database from DegreeWorks.

- 1) Create a degreeworks user in the new Banner database with select privileges to the list of tables in the DGW_Technical_Guide_Banner_Considerations_772D02_P03.pdf document. You must also grant execute privilege to the Banner function f_class_calc_fnc. If you use Self Service you must grant execute privilege to the function twbkbssf.
- 2) On the DegreeWorks server, update the \$ORACLE_HOME/network/admin/tnsnames.ora entry and add the new Banner database.
- 3) On the DegreeWorks server, edit the file /usr/local/dw/scripts/dw.start.config and modify the following line:
`export DB_LOGIN_BANNER=`
and set the value to
`export DB_LOGIN_BANNER=degreeworks/password@service`
where degreeworks is the user defined in the new Banner database, password is the degreeworks user password, and service is the service name from tnsnames.ora. Keep this change by saving the dw.start.config file.
- 4) Log in again into the DegreeWorks server so that the new DB_LOGIN_BANNER variable is set. To check, issue the command
`env | grep BANNER`
and you should see the entry from dw.start.config.
- 5) Test your new Banner connection by typing (as the dwadmin user logged into the DegreeWorks server):
`dbb`
and SQL*Plus should be launched into Banner. At this point you can query some of the tables the degreeworks user has access to, to verify that you are looking at the correct Banner database.
- 6) Determine if the following tables in the new Banner database are different from the Banner database you originally used to populate DegreeWorks. If so, rerun the associated processes so data from the new Banner database is used to populate DegreeWorks.

SCBCRSE, SCRATTR, SCRRTST	rerun bannerextract course
SCREQIV, SCBCRKY, STVCSTA	rerun bannerextract equiv
SHRGRDO, SHRGRDE	rerun bannerextract ucx for UCX-STU385
STVCLAS	rerun bannerextract ucx for UCX-STU305
STVCOLL	rerun bannerextract ucx for UCX-STU560
STVDEGC	rerun bannerextract ucx for UCX-STU307
STVGMOD	rerun bannerextract ucx for UCX-STU356
STVLEVL	rerun bannerextract ucx for UCX-STU350
STVMAJR	rerun bannerextract ucx for UCX-STU023, UCX-STU024, UCX-AUD027, UCX-AUD029, UCX-STU563
STVSTYP	rerun bannerextract ucx for UCX-STU306
STVTERM	rerun bannerextract ucx for UCX-STU016, UCX-STU035
STVACYR	rerun bannerextract ucx for UCX-STU035

After bannerextract ucx has been run, run ucx12job to update DegreeWorks files and picklists.

Creating Banner database link in DegreeWorks

Once you have finished all of the steps above and confirmed that you can connect to Banner with the dbb command, you should recreate the database link to Banner in the DW database.

IF THE BANNER AND DEGREEWORKS DATABASES ARE ON SEPARATE SERVERS: modify tnsnames.ora on the Banner database server so there is a connection to the DegreeWorks database. This is required for the Banner database link, created in the DegreeWorks database, to connect to Banner.

- 1) On the DegreeWorks server, log in as dwadmin and determine the Banner login information by issuing the following command:

```
env | grep BANNER
```

The result will be similar to the following:

```
DB_LOGIN_BANNER=BANNER_USER/BANNER_PW@BANNER_SERVICE
```

- 2) cd to the app/sql directory and use the vi editor to create a file called dblink.sql. Note that this file may exist if a link had been created to the old Banner database.

```
cd sql
```

```
vi dblink.sql
```

- 3) Add the following text to the dblink.sql file, substituting the values from DB_LOGIN_BANNER for: BANNER_SERVICE, BANNER_USER, and BANNER_PW.

```
create public database link BANNER_SERVICE
connect to BANNER_USER identified by BANNER_PW using 'BANNER_SERVICE';
create or replace public synonym spriden for spriden@BANNER_SERVICE;
create or replace public synonym sobcact for sobcact@BANNER_SERVICE;
create or replace public synonym sorlcur for sorlcur@BANNER_SERVICE;
create or replace public synonym sorlfos for sorlfos@BANNER_SERVICE;
create public synonym f_class_calc_fnc for f_class_calc_fnc@BANNER_SERVICE;
```

- 4) After saving the file and exiting the vi editor, issue the following command so that the dblink.sql script can be run:

```
chmod 755 dblink.sql
```

- 5) Log into SQL*Plus in the DegreeWorks database by issuing the db command. After first dropping the synonym for the function f_class_calc_fnc, execute the dblink.sql script:

```
db
SQL> drop public synonym f_class_calc_fnc;
SQL> @dblink
SQL> exit
```

Integration with Portals

End-User Access to DegreeWorks by Banner customers is typically accommodated through either Banner Self Service, or the Luminis products.

The following chapters provide more information about both these products, and their use with DegreeWorks.

Luminis

Single Sign-on for DegreeWorks

Instructions for integration of DegreeWorks with Luminis GCF

The following instructions will guide the user through configuration of Luminis and DegreeWorks (DW) for single sign-on integration using the Generic Connector Framework (GCF).

Overview of single sign-on process.

Luminis has several options for single sign-on integration and DegreeWorks (DW) uses the Generic Connector Framework (GCF). Configuration of the GCF sets up a Luminis URL API to handle the DW logon process. Once configured and DW SHP user credentials are loaded into Luminis LDAP, then Luminis is able to issue logon requests to DW on behalf of the user. That logon request is sent from Luminis server directly to DW server (not through the browser) when a DW link (bookmark) is accessed within Luminis. The Luminis server will also send a logout request directly to DW when the user logs out of Luminis.

This requires the client to download GCF onto the Luminis server as a “jar file”, install it and configure the connector server. Then, the DegreeWorks configuration files for Luminis must be installed and localized. Finally, the client must load the DegreeWorks user’s credentials into Luminis LDAP using “cptool” and set up a “bookmark” Channel to launch DegreeWorks in a separate window. DegreeWorks is compatible with Luminis 3 and 4.

Installation Steps: Luminis system configuration.

All of the following instructions take place on the Luminis server.

1. Ensure Generic Connector Framework (GCF) is installed on Luminis server. See documentation for installation instructions and GCF installation files can be acquired using SungardHE CRM (customer support center). Note the port number that the GCF virtual server, usually named “cpiconnector”, is running on (usually port 8008).
2. Review GCF documentation `lsdk0907im.pdf`
3. Set up the configuration for Luminis to use the connector for DegreeWorks. We will use the alias “degreeworks” to identify the external system connector. It is important to recognize this alias is case-sensitive.
 - a. Get `es.systems` and store it in a backup file:
`configman -g es.systems > es.systems.configman`
 - b. Edit that file (`es.systems.configman`) to add a space and then the text “degreeworks”. For example if the original contents of the file contains the one line:
`sct cal is`
Then edit the file to add a space and the text “degreeworks” for example:
`sct cal is degreeworks`
 - c. Import the changes:
`configman -i es.systems.configman`
 - d. Verify the changes have taken effect. The following command should show “degreeworks” at the end of the list just as in the file that was edited above.
`configman -g es.systems`
4. Set configuration items in Luminis LDAP site directory using the following commands:
 - a. `configman -s es.degreeworks.autosync false`
 - b. `configman -s es.degreeworks.configsleeptime 10000`
 - c. `configman -s es.degreeworks.configattempts 60`
 - d. `configman -s es.degreeworks.shortcircuitlogin false`
 - e. `configman -s es.degreeworks.configURL`
`“http://your.luminis.server:8008/cpiconnector/degreeworks/GetConfigVersion2”`
(Replace “your.luminis.server” with the actual address of your luminis web server. If your cpiconnector server was configured on a port different from 8008 then replace that with the actual port in use)
5. Verify that all the previous configuration are set as shown above:
`configman -g es.systems`
`configman -g es.degreeworks.configURL`
`configman -g es.degreeworks.autosync`
`configman -g es.degreeworks.configsleeptime`
`configman -g es.degreeworks.configattempts`
`configman -g es.degreeworks.shortcircuitlogin`
6. If the DegreeWorks server uses secure https (SSL) then set the following configuration:
 - a. View the current configuration: `configman -g es.systems.secure.login`
 - b. If the output of that command contains other systems, then add a space “degreeworks” to the end of that output. If not, then set “degreeworks” only as follows:
`configman -s es.systems.secure.login “degreeworks”`

Installation Steps: Luminis SSO configuration files.

All of the following instructions take place on the Luminis server.

1. Copy files `degreeworks.properties` and `degreeworks.xml` to the `cpipconnector` config directory. The location of the `cpipconnector` config directory depends on local configuration. For Luminis 3.3 this directory is usually at `$CP_ROOT/products/sso/config`. For Luminis 4 the directory is commonly at `$CP_ROOT/webapps/cpipconnector/WEB-INF/config`.
2. Ensure that appropriate access permissions are set on those files:

```
chmod 755 degreeworks.properties
chmod 755 degreeworks.xml
```
3. Edit `cpipconnector.properties` which should be already located in the `sso/config` directory. Add the name of the `degreeworks.properties` file to the `property.files` property which is a comma-delimited list.
 - a. If DegreeWorks is the only external system then it should look like this:

```
property.files=degreeworks.properties
```
 - b. But if the original configuration is `property.files=comexp.properties`, then add `degreeworks` like this:

```
property.files=comexp.properties,degreeworks.properties
```
4. Edit `degreeworks.properties` file to make the following localizations. Replace “`your.degreeworks.server`” with the actual address of your `degreeworks` server/url. Where appropriate, replace `http` with `https` if SSL is used.
 - a. `http://your.degreeworks.server/pickup.html`
 - b. `degreeworks.externalSystemURL = http://your.degreeworks.server`
 - c. Use the actual location of `sso/config` directory in the following setting instead of “`/opt/luminis/products/sso/config`”
`degreeworks.operations = /opt/luminis/products/sso/config/degreeworks.xml`
5. Still in `degreeworks.properties` file: Luminis version 3 requires the following settings that Luminis version 4 does not require. If using Luminis version 4, these settings should be removed.
 - a. `degreeworks.license.issued = Luminis Platform Generic Framework Connector Implementation Key`
 - b. `degreeworks.license.key = T211-XD3Q-2WGW-AEAD-E2E1-T2XP-SA21-9Q4P`
 - c. `degreeworks.coursemap.enabled = false`
 - d. `degreeworks.coursemap.cp.00001.200410 = _28_1`
6. Edit `degreeworks.xml` to replace the absolute url paths with the actual url paths in use on the DegreeWorks web server. For instance if the login page is located at `http://your.degreeworks.server/degreeworks/default.html`, then replace “`/dw2/default.html`” in this xml file on line 5 with “`/degreeworks/default.html`” and replace “`/dw2/IRISLink.cgi`” on lines 11 and 25 with “`/degreeworks/IRISLink.cgi`”.
7. Restart Luminis and `cpipconnector` servers. Whenever any of the above configuration changes are made, both servers must be restarted in order to take effect.

Installation Steps: DegreeWorks server.

All of the following instructions take place on the DegreeWorks server.

1. Upload the file pickup.html to the DegreeWorks web server document root directory.
2. Ensure that appropriate access permissions are set on that file:
`chmod 755 pickup.html`
3. If necessary update IRISLink.cgi to version 1.4 be sure not to lose local configuration settings.
4. In IRISLink.cgi set the following configuration setting:
`$IGNORE_LOGONADDRESS = $TRUE;`

Luminis Channel and User Configuration.

All of the following instructions take place on the Luminis server.

1. Set up a user to test single sign-on to DegreeWorks. In the following command replace luminis_username with an actual luminis user name, access_id with a DegreeWorks user name and access_code with the DegreeWorks user's password.
`cptool set user luminis_username
ExternalAccount="degreeworks|access_id|accesscode"`
2. For instructions on bulk loading of DegreeWorks users in to Luminis LDAP, review Luminis GCF Implementation Guide (lsdk0907im.pdf) section titled "SET THE CPIP EXTERNAL ACCOUNT FOR EXTERNAL SYSTEM USERS"
3. In order to set up a Luminis Channel for access to DegreeWorks log in with a user who has administrative access to control content/layout. Create a "bookmark" channel using a title of "DegreeWorks" and using a url similar to the following while replacing your.luminis.server and your.degreeworks.server/dw2/ locations with appropriate values. Also, if SSL is used, replace http in http%3A//your.degreeworks.server/dw2/ with https.

<http://your.luminis.server/cp/ip/login?sys=degreeworks&url=http%3A//your.degreeworks.server/dw2/IRISLink.cgi%3FSCRIPT%3DSD2WORKS>

Optionally set up multiple channels to control access for different user classes by changing the url arguments contained in the channel bookmark link. For instance, send a user class of "STU" or "ADV" as in the following example:

<http://your.luminis.server/cp/ip/login?sys=degreeworks&url=http%3A//your.degreeworks.server/dw2/IRISLink.cgi%3FSCRIPT%3DSD2WORKS%26USERCLASS%3DSTU>

Load DegreeWorks credentials into Luminis LDAP.

1. Take a csv file listing Luminis ID and SPRIDEN for each user that you want to load/enable for single sign on from Luminis to DegreeWorks. Upload it to a working directory on the DegreeWorks server. The format should look like this for example:

```
adam.red,911199999
adam_blue,911199994
adelina.green,911199997
```

2. Open that file in vi and prepare to do some find/replace on it. Use the following vi substitution commands to make an sql script out of it.

- a. `:%s/^/SELECT 'set user /g`
- b. `:%s/,/ ExternalAccount="dw|' || TRIM(shp_access_id) || '|' || TRIM(shp_access_code) || "' FROM shp_user_mst WHERE shp_access_id='/g`
- c. `:%s/$/';/g`

3. Save the file as `luminis_user_dw_creds.sql`.

4. Create another file that you will run through sqlplus to execute the file you created above. Note that you may want to change or use your own names for the files “`sptool_cmd_input.txt`” and “`luminis_user_dw_creds.sql`” and “`thisfile.sql`”.

-Run this file using this command:

```
SQLPLUS /@ @thisfile.sql
set termout off
set feedback off
set verify off
set echo off
set pagesize 0
set linesize 200
set trimspool on
column dt new_Value mydate noprint
select to_char(sysdate, 'YYYYMMDD') dt from dual;

spool cptool_cmd_input.txt
@luminis_user_dw_creds.sql;
spool off
exit;
```

5. Run that file using sqlplus or db for example:
`db /@ @thisfile.sql`

6. That should run sqlplus and output the file that you named in the spool command, for example cptool_cmd_input.txt. Inspect that file, it should have a format that looks like this:

```
set user adam.red ExternalAccount="dw|911199999|password"  
set user adam_blue ExternalAccount="dw|911199994|password"  
set user adelina.green ExternalAccount="dw|911199997|password"
```

7. Take that spool file and transfer it to the Luminis server. You will now execute the commands in that file using cptool (a Luminis program) that will load the user credentials into Luminis LDAP. Run the cptool command on it as follows:

```
cptool process file cptool_cmd_input.txt
```

Self Service Banner

Single Sign-on for DegreeWorks

Integration of DegreeWorks with Self Service Banner

Introduction.

These instructions will guide the user through configuration of Self-Service Banner (SSB) and DegreeWorks (DW) for single sign-on integration.

Overview of single sign-on process.

The purpose of these installation steps is to make a new menu item available in Self Service under the tab where it is configured. When that menu item is selected the link will send a single sign-on request to DegreeWorks then cause the DW application to be displayed if the single sign-on request is validated. If it is not validated then an error will be displayed.

Single sign-on is accomplished by means of a pl/sql SSB package that builds an html form containing the encoded arguments that DW needs to identify a user and validate the user's SSB session. The arguments are posted to DW cgi in an encoded format. An argument labeled SSBDATA contains a value that changes with every page view or refresh of the DW link in SSB. DW decodes that value then validates the session ID and spriden_pidm associated with that session in Banner. However DW does not update/change the session ID value like SSB normally does with each page view. The reason is to leave the SSB session intact so that the user can go back to SSB by means of a link and find that their SSB session is still valid and active as long as it has not timed out during their usage of DW.

Installation Steps: Self-Service Banner web menu setup.

Menu setup for Student role

1. Find dwssbstudent.sql file from app/sql directory on the DW host server. Use sqlplus to connect to your Banner database and then run the file. Run sqlplus using the shortcut script dbb to connect to your Banner database. This will create the package called “DW_Student” in the database. In sqlplus this command should run the file if you are running sqlplus from the app/sql directory: @dwssbstudent.sql
2. Log in to SSB with a user who has access to WebTailor Administration. Select the WebTailor tab, then [“Web Menus and Procedures”](#) from the menu.
3. Click the “Create” button to add a new web menu or procedure. Enter data for the following required fields. Enter data for the other fields according to your preference.
 - c. Page Name: DW_Student.P_SignOn
 - d. Description: DegreeWorks. Or, enter whatever text you would like to appear as the description.
 - e. Module: Student Self-Service. Or, select the appropriate module for your site.
 - f. Enter page title, header text, and header graphic as you prefer.
 - g. Enter back link settings as you prefer. If you are adding the DegreeWorks menu item to the student main menu then the back link url would be:
twbkwbis.P_GenMenu?name=bmenu.P_StuMainMnu.
 - h. Choose Associated Roles according to your preference.
4. Numbers 4 and 5 are options dependant on how you want to display or call the DW package from within Self Service. Display the DegreeWorks link on a main menu by calling the new web procedure from your main menu package: DW_Student.P_SignOn (term, pidm, 0). The first two arguments, term and pidm, can be sent as null. The third argument, show_headers, should equal 0 (zero) when you are calling this procedure from another package. If you choose to do this step, you can skip #5 below.

5. Display the DegreeWorks package link on a main menu. Do this if you skipped #4 above. This option will allow you to display a full page containing the DW link and any additional information you wish to provide.
 - i. Go back to WebTailor > Web menus and Procedures.
 - j. Select the menu where you want to add DegreeWorks (for example the main menu [bmenu.P_GenMnu](#) or the student main menu [bmenu.P_StuMainMnu](#))
 - k. Click “Customize Menu Items”
 - l. Click “Add a New Menu Item” (If you do not see that button yet, then you will need to click “Copy Baseline to Local” first).
 - m. Enter the URL: DW_Student.P_SignOn
 - n. Enter the link text, description and sequence number according to your preference.
 - o. Check the box next to Database Procedure.

6. Configure WebTailor Parameters
 - p. DWLINKTEXT – The text you want to display as the link to DegreeWorks
 - q. DWURL – The url to the DegreeWorks cgi. For example <https://yourserver.edu/degreeworks/IRISLink.cgi>.
 - r. DWDISPLAYBUTTON – Set to “1” to have a button displayed instead of an automatic redirect.

Menu setup for Faculty role

Setting up a menu item for a Faculty role is similar to the steps above with some differences listed as follows:

1. The sql file `app/sql/dwssbfaculty.sql` contains another package `DW_Faculty` with a procedure called `P_SignOn`. The default `USERCLASS` in this package is “ADV”. If you are providing this link to users who are `ADVX` or `REG` instead of `ADV`, then modify the hardcoded value for `USERCLASS` in this file. Insert the package into the Banner database using `dbb`.
2. The “Page Name” in step 3a above will be: `DW_Faculty.P_SignOn`.
3. Choose a module and associated roles that appropriate for faculty members.
4. Add the menu item for faculty members to a menu that is appropriate for your site. One option is to add it to the `bmenu.P_FacStuMnu` menu.

Installation Steps: DegreeWorks server.

1. Use SureCode to set CFG020 WEBPARAMS “Ignore Access Code” = “Y”
2. Change your header frame to include a link back to self-service. You should modify the **SD_HeaderFrame.html** file to add a link back to Self Service. We suggest you remove the Portal link and replace it with the call to the Self Service function defined in the SD2WORKS shpscript.

```
DrawLink ("Back to Self-Service", "top.BackToSelfServiceBanner('Main')");
DrawLink ("Transcript", "top.BackToSelfServiceBanner('Transcript')");
//DrawLink ("Portal", "", "_blank");
```

3. In SD2WORKS the BackToSelfService function is defined.
Be sure you change “myschool” and “sOMEMACHINE” shown below to be appropriate for your setup.

```
////////////////////////////////////
// You can pass in an sMode to be used here to control where the link
// should go within Self-Service.
////////////////////////////////////
function BackToSelfServiceBanner(sMode)
{
  // assume user is not a student (or not logged on as one anyway)
  var bStudent = false;
  <$ILMASK Service=SDSTUME>
  bStudent = true; // this is a student
  </$ILMASK>
  if (bStudent)
  {
    if (sMode=="Transcript")
      sMenuName="bwskotrn.P_ViewTermTran";
    else // normal mode
      sMenuName="twbkwbis.P_GenMenu?name=bmenu.P_AdminMnu";
  }
  else // not a student
  {
    if (sMode=="Transcript")
      sMenuName="bwlkftrn.P_FacDispTran";
    else // normal mode
      sMenuName="twbkwbis.P_GenMenu?name=bmenu.P_FacStuMnu";
  }
  window.location.href="http://myschool.edu/sOMEMACHINE/" + sMenuName;
} // backtoselfservicebanner
```

4. In the same HeaderFrame files you need to disable the DoLogout function. This function gets triggered when the DegreeWorks pages are unloaded when the user clicks on “Back to Self Service”. Odd behavior occurs when DoLogout actually tries to log the user out. In the DoLogout function add the “return;” as shown below:

```
function DoLogout()
{
  return; // disable because of Self Service
}
```

5. Issue a webrestart.

Student ID Pass-along to DegreeWorks

Overview

An advisor working in SSB must select a student before clicking on DegreeWorks. When the user does choose DegreeWorks the student being reviewed is passed to the student context area within DegreeWorks. As soon as the student appears the Worksheet tab is automatically selected and the student's most recent degree audit is displayed.

Allowing an advisor user coming from SSB to switch to another student in DegreeWorks would cause much confusion when the user then switched back to SSB since the new student ID is not then pushed back to SSB - they would be surprised to see the old student ID still sitting there in SSB.

To prevent such confusion it is best to take away the ability for these users to switch to another student within DegreeWorks. To remove this ability you should do a RemKey in common/SHPCFG on the SDSTUANY and SDFIND keys.

```
If (DGWUserClass = ADV) then  
  RemKey = SDSTUANY, SDFIND # Disallow changing student IDs in DW
```

How this works

The link from SSB requests the SD2WORKS DegreeWorks script. When the request is made the student ID from SSB is sent to DegreeWorks as PORTALSTUID=<someid>. The SD2WORKS script grabs this student ID and passes it along to the SD2STUCON student context area script. Once the SD2STUCON script sees that an ID was passed in it checks to see that the user is not a student and then immediately loads that student's name, degree, etc. If the user does not have the SDFIND or SDSTUANY keys they will not be able to switch to a new student; switching students must occur in SSB.

User Role Pass-along to DegreeWorks

Overview

An advisor working in SSB may be reviewing data on her advisees or may be looking at her own student records. The advisor would have two DegreeWorks links – one on the SSB student tab and one on the SSB faculty tab. When the advisor is working on one of her advisees and clicks the DegreeWorks link she needs to be able to continue to play the role of an advisor when in DegreeWorks. Conversely, when she is examining her own student record in SSB and clicks the DegreeWorks link she needs to play the role of a student when in DegreeWorks.

The above section on passing the Student ID from SSB to DegreeWorks is tightly related to this topic and thus the same adjustments to the advisor's keys need to be made.

How this works

The faculty tab containing the link to DegreeWorks must pass the USERCLASS of ADV (or ADVX) to DegreeWorks. The student tab must pass the USERCLASS of STU to DegreeWorks.

The user must have been extracted from Banner as an advisor at some point. This advisor user-class is stored in the shp-user-mst in DegreeWorks as the primary/overall user-class. When the user connects to DegreeWorks from SSB the USERCLASS passed from SSB is stored in the dap-user-mst table as the dynamic one to use for the current session. The primary user-class in the shp-user-mst is used to prevent a user with a primary user-class of STU from being changed to another other user-class. Similarly, a primary user-class of REG cannot be overwritten with any other user-class.