DegreeWorks Banner Considerations Technical Guide

Software Patch DW4.0.6 April 15, 2010



SUNGARD HIGHER EDUCATION

What can we help you achieve?

Baseline Release DW4.0.0 September 30, 2008

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

SunGard or its subsidiaries in the U.S. and other countries is the owner of numerous marks, including "SunGard," the SunGard logo, "Banner," "PowerCAMPUS," "Advance," "Luminis," "UDC," "Unified Digital Campus," "TreQ," and "DegreeWorks." Other names and marks used in this material are owned by third parties.

© 1995-2010 SunGard. All rights reserved. The unauthorized possession, use, reproduction, distribution, display or disclosure of this material or the information contained herein is prohibited.

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.



Think before you print.

Document Change Log

Version	Date	Change Description
DW4.0.6	April 15, 2010	Update to Banner Workflow Integration procedure
DW4.0.4	September 14, 2009	Miscellaneous text cleanup
	-	Updated required tables
		Text cleanup for Luminis (UCX-CFG020WEBPARAMS)
		Text cleanup for SSB (menu setup for student role)
		Addition of Special Topics
DW4.0.3	May 1, 2009	Class Attribute for transfer classes applied to Program
		Updated required tables
DW4.0.2	March 31, 2009	WebTreQer update notations
		Updated required tables
		Luminis Channels (Bookmark, CPIP Inline)
		Luminis LDAP Credentials
		Applicant Processing
		Workflow Integration
DW4.0.1	December 19, 2008	List of Database Table Access updated (SSRMEET)
		Updated text about Program Field
DW4.0.0	October 22, 2008	Added SCBDESC to Database Table Access
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	
Table of Contents	
Banner Considerations	
Banner Attributes	7
Banner Class Attributes	7
Transfer Classes Applied to Program	9
Banner Course Attributes	10
Banner Student Attributes	11
Banner Setup Checklist	
UCX	
Banner Data Extracts	13
Selecting People	
Deleting People Selecting LICX Entries	
Selecting Different Records for Your Institution	
Oracle	14
Scripts	14
Post extract	14
DegreeWorks Data Extract	
Configuration and Installation	
STEP 1 – Banner Database login	15
STEP 2 – Setup UCX-CFG020 BANNER	
STEP 4 – Setup SQL select files	
Batch Data Extract Process	20
hannerevtract	21
Student	21
Applicant	21
Advisor Staff	
Deleting IDs	
Selected UCX Tables	
Other Modes Transit – RAD30	
DGWCPUCOUNT	
Batch Extract Flow Diagram	
Dynamic Data Extract Process	
Banner Degree Coding Structure	
Banner Applicant Processing	
Required Access to Banner	
Database Table Access	

Function Access	
Banner Database Issues	
Pointing to different Banner database	
Creating the Banner Database Link in DegreeWorks	
Banner Workflow Integration (optional)	
Installation Guide	
Log in to the DegreeWorks host server	
Log in to the Banner Workflow host server	
Log in to the Banner Workflow web environment as admin user	
Managing Plan/Petition Approval – two tools	45
Integration with Portals	
Luminis	47
Single Sign-on for DegreeWorks	47
Instructions for integration of DegreeWorks with Luminis GCF	
Overview of single sign-on process.	
Installation Steps: Luminis system configuration.	
Installation Steps: DegreeWorks server	
Luminis User Configuration	
Luminis Bookmark Channel	51
Luminis CPIP Inline Channel	52
User Guide	
I wo Luminis SSO options	
Using the Luminis LDAP Gredentials (option 1)	
Solf Sorvice Banner	
Single Sign-on for DegreeWorks	
Integration of Degreeworks with Sen Service Barner	
Installation Steps: DegreeWorks server	62
Student ID Pass-along to DegreeWorks	
Overview	63
How this works	
User Role Pass-along to DegreeWorks	
Overview	
Special Topics	
Applicants in DegreeWorks	
How to extract applicants:	
Configuration Flags:	
Extract Process	
Bannerextract.config file	
Using Banner Data to create Scribe Custom Data	70
Scribing against Test scores	
Using Bannar data as Transit aslastics	
Using Danner data as Transit Selection	80

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.

🗱 5CR044	
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).	<u> </u>
Description Banner attributes Element ATTR Must be valid element number in RAD-Class-Dtl or be "ATTR" UCX Table	

Transfer Classes Applied to Program

(As of DW4.0.3)

In addition to class attributes generated from the standard Banner attribute tables, it is also possible to generate a rad_attr_dtl record for Transfer Classes which have been applied to a Program in Banner. Such classes are identified by linking the Banner Transfer Equivalence record, SHRTRCE, to the Transfer Course Degree Applied table SHRTRCD where SHRTRCD_APPLIED_IND = 'Y'. The rad_attr_value will be populated with the student's Program Code, which is extracted from the associated Degree table SHRDGMR. To enable this feature, set the UCX-CFG020BANNER flag "Transfer Program Attr" to 'Y'.

You could then scribe the Program value in the related requirement blocks so that the Transfer Classes are applied appropriately. For instance, "do not accept any transfer classes for this requirement that are not applied to the AA program (coded attribute of PRAA)" could be scribed as:

```
MaxClasses 0 in @ (With DWTransfer = Y and Attribute \langle \rangle PRAA)
```

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.

🐹 5CR002	- 🗆 🗵
KEY: ATTRIBUTE	
REMINDER: To enable your UCX-SCR002 changes be sure to restart the web jobs. When using SSCPA or BannorGPA you do not need an entry here	-
Description Banner Student Attributes Data Element R323 Specify the datum to be used in a Scribe IF statement UCX Table	
Edit Element 1 R322 If Data is on a detail this element is used as a filter Type EV AT / EV / SC / HD / DG / DI / CS Value ATTRIBUTE 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
	load these fields
LICV PANO80 Custom Data	Total litese fields.
UCA-DAINUOU CUSIOIN Dala	the worksheet setup LICX RANO80 Please refer to the
	Technical Guide LICY documentation for datails on how to
	actum LICX DAMONO
	SELUP UCA-BAINUOU.

Banner Data Extracts

Selecting People

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

```
local/sql/bannerstudents.sql
local/sql/bannerapplicants.sql (As of DW4.0.2)
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. For example, load one table per line:

STU356 STU385 STU560

Add a ".ucx" extension to this file in the local/sql directory. The actual file name can be any valid Unix file name. However, it MUST have the ".ucx" extension. For example, the file "bantables.ucx" may be created: local/sql/bantables.ucx. The banner extract command would be:

bannerextract ucx bantables.ucx

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

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 applicants (As of DW4.0.2)
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 - unless the UCX-CFG020 BANNER Add UCX Entries Only is set to Y so that none of the records you changed will be deleted – only new records in Banner will be added to the UCX.

Table	Fields to setup
UCX-AUD027	Filter fields
UCX-STU016	Planner flag
UCX-STU035	Planner flag
UCX-STU307	Short degree field
UCX-STU346	Calendar codes used in WebTreQer (As of DW4.0.2)
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

If this does not show the correct Banner database login you need to edit this variable in dwenv.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.

If this is not set correctly, review your setup of dwenv.config - see config_SetupMenuItems.

STEP 2 – Setup UCX-CFG020 BANNER

Review all settings BEFORE running the extract.

💥 CFG020	
KEY: BANNER	
Banner Site Y =Banner Site - bannerdb exists	_
Email Code SCHL Email code used to find desired GOREMAL record	
Email Override Y Y=If Email code not found, then use Active Email	
Repeat Skip A 👖 V=Skip Averaged classes	
Repeat Policy A 5 1-6, B or BLANK, Repeat Policy if Averaged classes included	
Repeat Skip E V=Skip Excluded classes	
Repeat Policy E 0-6, B or BLANK, Repeat Policy if Excluded classes included	
Repeat Policy I 5 1-6, B or BLANK, Repeat Policy for Included classes	
GPA Type 0 I=Institutional, O=Overall	
Add UCX Entries Only T Y=Only ADD UCX entries, NO deletes	
Use Term as Catalog Year 🛛 Y=Load UCX-STU035 with UCX-STU016 Term Codes	
Search in Banner 🕅 Y=In web search, search in Banner, N=Search in DegreeWorks	
Advisor User Class ADVX ADV=Can make exceptions, ADVX=Cannot make exceptions	
Staff User Class REG Standard value is REG	
Non Course Score C=NCST Code, I=STVNCST Indicator	
Default Current Grade NR Used on all Current classes	
Transfer Repeat Skip A 🕎 🛛 Y=Skip Averaged classes	
Transfer Repeat Policy A s 1-6, B or BLANK, Repeat Policy if Averaged classes included	
Transfer Repeat Skip E Y=Skip Excluded classes	
Transfer Repeat Policy E 0-6, B or BLANK, Repeat Policy if Excluded classes included	
Transfer Repeat Policy I s 1-6, B or BLANK, Repeat Policy for Included classes	
	-

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.

ČFG020	
ĸ	EY: BANNER
	H - F F 🗙 🖬 🎒 🖇
	_
Advisor Method	S A=Load 4 Advr Codes, C=Match on Major and Minor Codes, S=Sequential
Advisor Major	MAJR SGRADVR_CODE that identifies a Major Advisor if Method "C" used
Advisor Minor	MINR SGRADVR_CODE that identifies a Minor Advisor if Method "C" used
Load Extra Advisors	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	L N=Limits not checked; L=Repeat Limit; U=Max Rpt Units; B=Both; I=Include Course
Current Course	R A=STVCSTA_ACTIVE_IND of 'A'; C=SCBCRSE_CSTA_CODE of 'C'; K=SCBCRKY End Term
Follow Gradable Ind	Y=Skip current class if GradableInd = 'N'
Method "A" Advisor Code1 1	HAJR SGRADVR_ADVR_CODE to be loaded into RAD Advr1
Code1 2	
Code1 3	
Advisor Code2 1	MINR SGRADVR_ADVR_CODE to be loaded into RAD Advr2
Code2 2	CONC
Code2 3	
Advisor Code3 1	THES SGRADVR_ADVR_CODE to be loaded into RAD Advr3
Code3 2	
Code3 3	
Advisor Code4 1	PEER SGRADVR_ADVR_CODE to be loaded into RAD Advr4
Code4 2	MAJR
Code4 3	
Check Dual Degrees	Y Y=Check SGBSTDN Dual Degree fields
Always Process SHRATTC	N Y=Include SHRATTC attributes even if SHRATTR found
GPA Report	N Y=Create rad_report_dtls for GPAxx, GPACREDITSxx for "IN", "OV" and "TR"
Program as Degree	N Y≑Create DW Degree using Banner Program. Default=N
Inactive in SCBCRKY	Y =Skip Inactive Courses in SCBCRKY in EQUIV (End Term NOT all 9's). Default=Y
Process Applicants	Y =Look for "ADMISSIONS" applicant data in SORLCUR/SORLFOS and load if LevI/Degree unique
Process Both Goals	Y =If Student Goal data exists, still load Applicant Goal data if LevI/Degree unique
Load SARADAP Goals	Y =Load SARADAP Goal data if no "ADMISSIONS" SORLCUR record found and Levi/Degree unit
Cross-listed Term	s Load CFG073 Cross-listed Term, B=Blanks, L=Lowest Term, S=Start Term

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 **applicants** you want bridged to DegreeWorks: local/sql/bannerapplicants.sql
- Review and modify as needed the file used to select the **advisors** you want bridged to DegreeWorks: local/sql/banneradvisors.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 SQL FROM/WHERE clauses for every Banner table used by the banner extract programs are included in this configuration file.

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_FCN". 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_goal_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.

Here is an example of the SORLCUR/SORLFOS entries in this configuration file:

SORLCUR must be a; AND is required at the end of the WHERE SORLCUR-from: FROM SORLCUR a SORLCUR-where: WHERE a.SORLCUR_CACT_CODE = 'ACTIVE' SORLCUR-where: AND a.SORLCUR_SEQNO = SORLCUR-where:(SELECT MAX(b.SORLCUR_SEQNO) FROM SORLCUR bSORLCUR-where:WHERE b.SORLCUR_PIDM= a.SORLCUR_P = a.SORLCUR_PIDM AND b.SORLCUR_PRIORITY_NO = a.SORLCUR_PRIORITY_NO SORLCUR-where: AND b.SORLCUR_LMOD_CODE = 'LEARNER') SORLCUR-where: SORLCUR-where: AND a.SORLCUR_PIDM = <students-pidm> łŁ # SORLFOS must be a; AND is required at the end of the WHERE SORLFOS-from: FROM SORLFOS a, SORLCUR b SORLFOS-where: WHERE b.SORLCUR CACT CODE = 'ACTIVE' SORLFOS-where: AND b.SORLCUR_SEQNO = SORLFOS-where: (SELECT MAX(c.SORLCUR_SEQNO) FROM SORLCUR c WHERE c.SORLCUR_PIDM = b.SORLCUR_PIDM SORLFOS-where: AND c.SORLCUR_PRIORITY_NO = b.SORLCUR_PRIORITY_NO SORLFOS-where: SORLFOS-where:AND c.SORLCUR_LMOD_CODE= 'LEARNSORLFOS-where:AND a.SORLFOS_CSTS_CODE= 'INPROGRESS' AND c.SORLCUR_LMOD_CODE = 'LEARNER') SORLFOS-where: AND a.SORLFOS_CACT_CODE = 'ACTIVE' SORLFOS-where: AND a.SORLFOS_PIDM = b.SORLCUR_PIDM SORLFOS-where: AND a.SORLFOS_LCUR_SEQNO = b.SORLCUR_SEQNO SORLFOS-where: AND a.SORLFOS_PIDM = <students-pidm>

Batch Data Extract Process

Student and/or applicant academic data is required for DegreeWorks to generate a Degree Audit. SunGard extracts student data, applicant 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 the SQL specified in the \$ADMIN_HOME/common/bannerextract.config file. Student data can also be extracted individually by the SPRIDEN ID.
- Applicants admissions applicants academic and basic biographic data may be extracted based on the SQL specified in the \$ADMIN_HOME/common/bannerextract.config file. However, only unique combinations of the Level (School) and Degree may be extracted (this assumes the UCX-CFG020 BANNER configuration flags are set appropriately and/or the APPLICANT data extract is performed). Applicant data may also be extracted individually by the SPRIDEN ID. (As of DW4.0.2)

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

Mappings – Transfer Articulation Mappings for import into TreQ and WebTreQer This mapping information is also used by a CourseLink display. (As of DW4.0.4)

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.

The bannerextract for ETS must be run before bannerextract mappings.

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
```

Sometimes a need arises to force a student to be bridged from Banner into DegreeWorks thereby ignoring or overriding the hash value that is normally checked. An environment variable can be set to force this extract. This variable must be set at the command line. The extract must also be run from the command line. Once the user does not want to force extracting, the user can either log off or reset the environment variable. Make sure the command is typed exactly as below (case Does matter). Only extracts performed from the command line by this user will be affected by this environment variable. Transit extracts will not be affected. Extracts done during a cron job will not be affected unless explicitly denoted in the cron job.

```
For Banner sites the commands are:

$export RAD11FORCE=ALL
$bannerextract student listofstudents
```

Applicant

```
(As of DW4.0.2)
```

- To run the **applicant** extract using bannerapplicants.sql file in the local/sql directory: \$ bannerextract applicant
- You may also specify a different sql file in the local/sql directory: \$ bannerextract applicant someapplicants.sql
- You may also specify a file of applicant IDs in the local/sql, admin/data or current directory: \$ bannerextract applicant someapplicants.ids
- For testing purposes you may also supply a single applicant ID instead of a file name:
 - \$ bannerextract applicant 2468642

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, you must save a file of staff IDs in the local/sql directory. The file can have any name but it must have the .ids extension, for example "staff.ids". Exit the local/sql directory before running the following command (you must NOT be in local/sql when running bannerextract):

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

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

```
$ bannerextract staff 1234567
```

Deleting IDs

To delete unwanted IDs from the DegreeWorks database, you can create a query to select those individuals, and save it in a file named local/sql/bannerdeleteids.sql. To run the delete function, issue the following command after exiting 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 ".ucx" extension (**do NOT use with cron**):

\$ bannerextract ucx someucxtables.ucx

WARN: DO NOT put the UCX_prefix on the table names even though the actual database table names are: UCX_STU352, UCX_STU560 and UCX_STU563)

For example, the "local/sql/someucxtables.ucx" file might contain tables:

STU352

STU560

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
```

Only adds/updates rad_course_mst records, but deletes and re-adds rad_crs_attr_dtl records.

\$ bannerextract ucx

If UCX-CFG020 BANNER Add UCX Entries Only = "N" all records are deleted and re-added. If UCX-CFG020 BANNER Add UCX Entries Only = "Y" only new records will be added (no updates).

\$ bannerextract equiv

The dap_eqv_crs_mst and UCX-CFG070 are first both deleted and all equivalencies re-added.

```
$ bannerextract ets
```

Only adds/updates rad_ets_mst records (no deletes).

\$ bannerextract mappings

The old mappings are first deleted and then new mappings are re-added.

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.

🚻 Transit - SunGard Higher Education					-O×
File Options Tools Help					
🗃 🖬 🦻 😵 🛛 RAD30	Report RAD30 Banner B	Extract and Bridge			
Selection Questions	Sort				
			Set As	Defaults	
	What type of extract do y	ou want to run today? *	STUDENT		
Do you want '	to use the default SQL file to	o select the students?	ADVISOR APPLICANT	Advisor extract/bridge Applicant extract/bridge/audits	-
For STUDENT mode you may use th For STUDENT mode local/sql/b For APPLICANT mode local/sql/	e default SQL file or ente cannerstudent.sql is the do bannerapplicants.sql will	er Selection criteria efault SQL file always be used	EQUIV ETS MAPPINGS STAFF STUDENT	Equivalences extract/bridge Transfer school extract/bridge Transfer Articulation extract/bridge Staff extract/bridge/audits	
For ADVISOR mode local/sql/	banneradvisor.sql will alv	ways be used			
New audits will be genera	ited for students with data	i changes			
New audits will be generat Be sure your CFG020 BANNER sett	ed for applicants with dat lings are correct before ru	a cnanges inning the extract			
	- torot	Grand			
	Launch	Cancel			

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 dwenv.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.

Find Student ID I∢ ◀ Name I	N Degree Major	School Level	Last Audit	Last Refresh	
🔍 1971 Deans, Ruari Padraig	B.S. V Mathematics	U Sophomore	Today	Today at 1:19 p.m.	

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.

🕅 CFG020							
KEY: REFRESH							
·							
This record deals with the automatic refresh that occurs when audits are requested.	^						
This record does not affect the on-demand refresh occurring from a push of the refresh button.							
With View=Y a refresh occurs and a new audit is run if there are any data changes							
With View=N and other flags are Y we run a new audit if audit date is older than bridge date							
Dynamic Refresh Y=Dynamic refresh allowed							
Refresh Timeout 00000 Minutes before new refresh allowed, 9999=never refresh, 0000=always refresh							
Run Audit Refresh Y Y=Refresh when running New Audit							
What-If Refresh V=Refresh when running What-If Audit							
Look Ahead Refresh 🕎 Y=Refresh when running Look Ahead Audit							
Planner Audit Refresh 🕎 Y=Refresh when running Planner Audit							
Exception Audit Refresh Y = Refresh when running Exception Audit							
View Audit Refresh N Y=Refresh when Viewing audits (but not History)	•						

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 and extract the student's degreemajor combination. In DegreeWorks, the degree is specifically taken from sorlcur_degc_code and the major is specifically taken from the associated SORLFOS record(s). However, the sorlcur_program is extracted and stored separately and can be referenced with Scribe to create a Program Requirement block, or differentiate between degree-major combinations by using the Program as a secondary tag. (As of DW4.0.1)

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	This will result in one degree record in
SORLFOS - sorlfos_majr_code =CHEM; lcur_seq_no=1	DegreeWorks - a BS degree with two majors.
SORLFOS - sorlfos_majr_code =BIOL; lcur_seq_no=1	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	This will result in one degree record in
SORLFOS - sorlfos_majr_code =MATH; lcur_seq_no=1	DegreeWorks - a BS degree with two majors.
SORLCUR - sorlcur_degc_code =BS; seq-no=2	The degree audit will be run against this
SORLFOS - sorlfos_majr_code =PHYS; lcur_seq_no=2	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	This will result in two degree records in					
SORLFOS - sorlfos_majr_code =CHEM; lcur_seq_no=1	DegreeWorks - a BS degree with a CHEM					
SORLCUR - sorlcur_degc_code =BA; seq-no=2	major and a BA degree with an ARTH major.					
SORLFOS - sorlfos_majr_code =ARTH; lcur_seq_no=2	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	This will result in two degree records in
SORLFOS - sorlfos_majr_code =CHEM; lcur_seq_no=1	DegreeWorks - a BS degree with a CHEM
SORLCUR - sorlcur_degc_code =MA; level=GR; seq-no=2	major and an MA degree with a PHIL major.
SORLFOS - sorlfos_majr_code =PHIL; lcur_seq_no=2	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 re	commended
Definition of degree, and location of data, in Banner	Results in DegreeWorks
Definition of degree, and location of data, in Banner 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	Results in DegreeWorks 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.

Banner Applicant Processing

(As of DW4.0.2)

DegreeWorks now allows the import of admissions data from Banner, allowing applicants to view degree audit worksheets. This can be a powerful recruiting tool for applicants who have transfer, test scores, custom records or other appropriate data in DegreeWorks.

The Banner extract (RAD30/BAN40) may now be run with the new **APPLICANT** mode to create the appropriate Goal and Goal Data records for each unique Level (school) and Degree combination (as well as extract all other appropriate Banner data for an applicant that may be used by DegreeWorks). The picture below outlines how the Goal Data may be loaded into DegreeWorks by the ban40 data extract with three STUDENT paths and two APPLICANT paths:



Basically there are 5 paths that ban40 may take to load student/applicant Goal data:

- 1) Student Goal Data may be loaded from 'LEARNER' SORLCUR/SORLFOS curriculum records
- 2) Student Goal Data may be loaded from SGBSTDN curriculum data
- 3) Student Goal Data may be loaded from SGBSTDN DUAL degree data
- 4) Applicant Goal Data may be loaded from 'ADMISSIONS' SORLCUR/SORLFOS curriculum records
- 5) Applicant Goal Data may be loaded from SARADAP curriculum data

The rules for determining how ban40 decides what Goal Data to extract from Banner are defined below.

Three new configuration flags in the UCX-CFG020 BANNER record control how the applicant data is to be extracted and which rules will be followed:

Process Applicants - A Y/N flag. Set to "Y" if applicant processing by the ban40 extract is to be performed. Set to "N" if NO applicant data is to be imported into DegreeWorks. If this flag is set to "Y" and the REFRESH button on the web is clicked, then Banner applicant data will be looked up **in addition to** student data.

Process Both Goals - A Y/N flag. Set to "Y" if goal (degree) data from Banner Student (LEARNER) as well as Applicant (ADMISSIONS) data is to be imported into DegreeWorks. Set to "N" if the ADMISSIONS SORLCUR/SORLFOS records are NOT to be imported into DegreeWorks if 'LEARNER' SORLCUR/SORLFOS data is found. For example, if a student is working toward an undergraduate degree, but has applied to graduate school at the same institution applicant data could also exist at the same time. Thus, this flag would need to be set to 'Y' so that both the undergraduate student data as well as the graduate applicant data is imported into DegreeWorks.

Load SARADAP Goals - A Y/N flag. Set to "Y" if SARADAP is to be processed if NO SORLCUR ADMISSIONS data is found. Set to "N" if NO SARADAP data is to be loaded into DegreeWorks.

The ban40 Banner data extract is now a STUDENT "and" APPLICANT extract. When processing begins, the software will determine whether any of the following conditions exist and set the configuration flags appropriately:

- The LEARNER SORLCUR/SORLFOS records are retrieved using the SORLCUR query in \$ADMIN_HOME/common/bannerextract.config (where the SORLCUR query specifies SORLCUR_LMOD_CODE = 'LEARNER' along with all other required SQL). If found then set LEARNER_FOUND = "Y".
- 2) If the UCX-CFG020 BANNER Process Applicants = "Y" (for Refresh over the web) or the bannerextract script is run with mode **APPLICANT**, then two additional checks will be made:
 - 1. No LEARNER SORLCUR/SORLFOS records were found OR
 - 2. The UCX-CFG020 BANNER Process Both Goals is set to "Y"

If either condition is met then ADMISSIONS SORLCUR/SORLFOS records are retrieved using the SORLCUR2 query in \$ADMIN_HOME/common/bannerextract.config (where the SORLCUR query specifies SORLCUR_LMOD_CODE = 'ADMISSIONS' along with all other required SQL). If found then set ADMISSIONS_FOUND = "Y".

- 3) The SGBSTDN General Student record is looked up regardless of whether any SORLCUR/SORLFOS records are found. If an SGBSTDN record is found, SGBSTDN_FOUND will be set to "Y".
- 4) If ADMISSIONS_FOUND = Y then the associated SARADAP record will be identified using the SORLCUR_PIDM, SORLCUR_TERM_CODE and SORLCUR_KEY_SEQNO (linked to SARADAP_APPL_NO). If ADMISSIONS_FOUND = N, and the UCX-CFG020 BANNER Process Applicants flag = "Y" (or the APPLICANT mode is found in batch) and the UCX-CFG020 BANNER Load SARADAP Goals = "Y" the SARADAP record will be looked up using the SARADAP query in \$ADMIN_HOME/common/bannerextract.config. If a valid record is found, SARADAP_FOUND = "Y".

Once the appropriate records have been read, the four flags (LEARNER_FOUND, ADMISSIONS_FOUND, SGBSTDN_FOUND and SARADAP_FOUND) will be checked. If data was NOT retrieved for ANY of these four conditions then an error message will be written to the extract log and processing will quit for this ID Code.

Next, the software processes the curriculum data into using the following five rules or paths:

- 1) If LEARNER_FOUND = "Y" then the appropriate rad_goal_dtl and rad_goaldata_dtl records will be created from LEARNER SORLCUR/SORLFOS.
- 2) If LEARNER_FOUND = "N" and ADMISSIONS_FOUND = "N" and SGBSTDN_FOUND = "Y" then the appropriate rad_goal_dtl and rad_goaldata_dtl records will be created from SGBSTDN.
- 3) If the UCX-CFG020 BANNER Check Dual Degree = "Y" and SGBSTDN_FOUND = "Y" then the Dual Degree fields will be checked. If the Dual Degree contains data in the Dual Level (school) and Dual Degree fields, and the combination is unique (does not match any LEARNER combinations previously extracted) then the appropriate rad_goal_dtl and rad_goaldata_dtl.records will be created from the Dual Degree.
- 4) If ADMISSIONS_FOUND = "Y" and the Level (school)/Degree combination is unique (does not match any LEARNER combinations previously extracted), then the appropriate rad_goal_dtl and rad_goaldata_dtl records will be created from the ADMISSIONS SORLCUR/SORLFOS data.
- 5) If ADMISSIONS_FOUND = "N" and UCX-CFG020 BANNER Process Applicants = "Y" and UCX-CFG020 BANNER Load SARADAP Goals = "Y" and SARADAP_FOUND = "Y" and the Level (school)/Degree combination is unique (does not match any LEARNER or SGBSTDN combinations previously extracted) then the appropriate rad_goal_dtl and rad_goaldata_dtl records will be created from SARADAP.

Processing will then continue so that other Banner data for the individual, such as transfer classes, test scores (e.g., AP tests), etc., are extracted and imported into DegreeWorks.

Example of APPLICANT SQL in \$ADMIN_HOME/common/bannerextract.config:

SORLCUR must be a; AND is required at the end of the WHERE SORLCUR2-from: FROM SORLCUR a, STVTERM t, SARADAP c SORLCUR2-where: WHERE (SELECT COUNT(*) FROM SHRTRCR SORLCUR2-where: WHERE SHRTRCR_PIDM = c.SARADAP_PIDM) > 0 SORLCUR2-where: AND t.STVTERM_START_DATE > SYSDATE SORLCUR2-where: AND ((SELECT COUNT(*) FROM SGBSTDN SORLCUR2-where: WHERE SGBSTDN_PIDM = c.SARADAP_PIDM) < 1</pre> SORLCUR2-where: OR (SELECT COUNT(*) FROM STVSTST, SGBSTDN p SORLCUR2-where: WHERE STVSTST_CODE = p.SGBSTDN_STST_CODE SORLCUR2-where: AND STVSTST REG IND = 'Y' SORLCUR2-where: AND p.SGBSTDN_TERM_CODE_EFF = (SELECT MAX (o.SGBSTDN_TERM_CODE_EFF) FROM SGBSTDN o SORLCUR2-where: SORLCUR2-where: WHERE o.SGBSTDN_PIDM = p.SGBSTDN_PIDM SORLCUR2-where: AND o.SGBSTDN_TERM_CODE_EFF <</pre> SORLCUR2-where: c.SARADAP_TERM_CODE_ENTRY)) < 1)</pre> SORLCUR2-where:ANDa.SORLCUR_CACT_CODE = 'ACTIVE'SORLCUR2-where:ANDa.SORLCUR_LMOD_CODE = 'ADMISSIONS' SORLCUR2-where: AND a.SORLCUR SEQNO = (SELECT MAX(b.SORLCUR SEQNO)

```
SORLCUR2-where:FROM SORLCUR bSORLCUR2-where:WHERE b.SORLCUR_PIDM = a.SORLCUR_PIDMSORLCUR2-where:AND b.SORLCUR_PRIORITY_NO = a.SORLCUR_PRIORITY_NOSORLCUR2-where:AND b.SORLCUR_LMOD_CODE = 'ADMISSIONS')SORLCUR2-where:AND a.SORLCUR_PIDM = c.SARADAP_PIDMSORLCUR2-where:AND a.SORLCUR_TERM_CODE = c.SARADAP_TERM_CODE_ENTRYSORLCUR2-where:AND a.SORLCUR_KEY_SEQNO = c.SARADAP_APPL_NOSORLCUR2-where:AND t.STVTERM_CODE = c.SARADAP_TERM_CODE_ENTRYSORLCUR2-where:AND t.STVTERM_CODE = c.SARADAP_TERM_CODE_ENTRYSORLCUR2-where:AND t.STVTERM_CODE = c.SARADAP_TERM_CODE_ENTRYSORLCUR2-where:AND t.STVTERM_CODE = c.SARADAP_TERM_CODE_ENTRY
```

∦ SORLFOS must	be a; AND is required at the end of the WHERE	
SORLFOS2-where:	FROM SORLFOS a, SORLCUR b, STVTERM t, SARADAP d	
SORLFOS2-where:	WHERE (SELECT COUNT(*) FROM SHRTRCR	
SORLFOS2-where:	WHERE SHRTRCR_PIDM = d.SARADAP_PIDM) > 0	
SORLFOS2-where:	AND t.STVTERM_START_DATE > SYSDATE	
SORLFOS2-where:	AND ((SELECT COUNT(*) FROM SGBSTDN	
SORLFOS2-where:	WHERE SGBSTDN_PIDM = d.SARADAP_PIDM) < 1	
SORLFOS2-where:	OR (SELECT COUNT(*) FROM STVSTST, SGBSTDN p	
SORLFOS2-where:	WHERE STVSTST_CODE = p.SGBSTDN_STST_CODE	
SORLFOS2-where:	AND STVSTST_REG_IND = 'Y'	
SORLFOS2-where:	AND p.SGBSTDN_TERM_CODE_EFF = (SELECT MAX	
SORLFOS2-where:	(o.SGBSTDN_TERM_CODE_EFF) FROM SGBSTDN o	
SORLFOS2-where:	WHERE o.SGBSTDN_PIDM = p.SGBSTDN_PIDM	
SORLFOS2-where:	AND o.SGBSTDN_TERM_CODE_EFF <	
SORLFOS2-where:	<pre>d.SARADAP_TERM_CODE_ENTRY)) < 1)</pre>	
SORLFOS2-where:	AND b.SORLCUR_CACT_CODE = 'ACTIVE'	
SORLFOS2-where:	AND b.SORLCUR_LMOD_CODE = 'ADMISSIONS'	
SORLFOS2-where:	AND b.SORLCUR_SEQNO = (SELECT MAX(f.SORLCUR_SEQNO)
SORLFOS2-where:	FROM SORLCUR f	
SORLFOS2-where:	WHERE f.SORLCUR_PIDM = b.SORLCUR_PIDM	
SORLFOS2-where:	AND f.SORLCUR_PRIORITY_NO = b.SORLCUR_PRIORITY_NO	
SORLFOS2-where:	AND f.SORLCUR_LMOD_CODE = 'ADMISSIONS')	
SORLFOS2-where:	AND b.SORLCUR_PIDM = d.SARADAP_PIDM	
SORLFOS2-where:	AND b.SORLCUR_TERM_CODE = d.SARADAP_TERM_CODE_ENT	5
SORLFOS2-where:	AND b.SORLCUR_KEY_SEQNO = d.SARADAP_APPL_NO	
SORLFOS2-where:	AND t.STVTERM_CODE = d.SARADAP_TERM_CODE_ENTRY	
SORLFOS2-where:	AND a.SORLFOS_CSTS_CODE = 'INPROGRESS'	
SORLFOS2-where:	AND a.SORLFOS_CACT_CODE = 'ACTIVE'	
SORLFOS2-where:	AND a.SORLFOS_PIDM = b.SORLCUR_PIDM	
SORLFOS2-where:	AND a.SORLFOS_LCUR_SEQNO = b.SORLCUR_SEQNO	
SORLFOS2-where:	AND a.SORLFOS_SEQNO =	
SORLFOS2-where:	(SELECT MAX(1.SORLFOS_SEQNO) FROM SORLFOS 1	
SORLFOS2-where:	WHERE 1.SORLFOS_PIDM = b.SORLCUR_PIDM	
SORLFOS2-where:	AND 1.SORLFOS_PRIORITY_NO = a.SORLFOS_PRIORITY_NO	
SORLFOS2-where:	AND l.sorlfos_csts_code = 'INPROGRESS'	
SORLFOS2-where:	AND 1.SORLFOS_LCUR_SEQNO = b.SORLCUR_SEQNO)	
SORLFOS2-where:	AND	
#	a.SORLFOS_PIDM = <students-pidm></students-pidm>	

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 tables listed in the chart below.

SPECIAL NOTE: If your site uses the custom SQL in the UCX-BAN080 table to extract non-standard pieces of data from Banner, those database tables will NOT be listed here. However, SELECT access must be provided for those tables as well.

Banner Table	Course Catalog	Course Equivalents	ETS (transfer)	Advisors / Staff	Students	Applicants	UCX	Banner General	TreQ / WebTreQer	Description
GOREMAL				Х	Х	Х				Email Address Data
SARADAP						Х				Applicant Degree Data (As of DW4.0.2)
SCBCRKY		Х								Course Start/End Dates
SCBCRSE	Х	Х								Course Master Data
SCBDESC	Х									Course Description
SCRATTR	Х									Course Attributes
SCREQIV		Х								Course Equivalents
SCRRTST	Х	1								Course Prerequisites
SFRSTCR		1			Х	Х				Current Class Data
SGBSTDN					Х	Х				General Student Degree Data
SGRADVR				Х	Х	Х				Advisor Data
SGRSATT					Х	Х				Student Attributes by PIDM and Current Term
SHBTATC									Х	Transfer Institution Transfer Catalog Data (As of DW4.0.2)
SHRATTC					Х	Х				Student Attributes by CRN and Historic Term
SHRATTR					Х	Х				Student Attributes by PIDM and Historic Sequence Number
SHRDGMR					Х					Student Degree table (As of DW4.0.3)
SHRGRDE					Х	Х				Grade Codes (UCX-STU385)
SHRGRDO					Х	Х				Grade Codes – valid combo of level/gmod/grade (UCX-STU385)
SHRICMT									Х	Transfer Articulation Institution Course Comment (As of DW4.0.2)
SHRLGPA					Х	X				Summary GPA/Credits Data
SHRNCRS					X	X				Non Course Data
SHROPNM					Х	Х				Non Course Data – Papers and Exams
SHRTATC									Х	Transfer Institution Catalog Equivalent Data (As of DW4.0.2)
SHRTATT					Х	X				Student Attributes by PIDM and Transfer Sequence Number
SHRTCKG					X	X				Historic Grade Values
SHRTCKL					X	X				Historic Level (School) Data
SHRTCKN					X	X				Historic Class Data
SHRTGPA					X	X				Detail GPA/Credits by Term
SHRTRCD					X					Transfer Course Degree Applied table (As of DW4.0.3)
SHRTRCE					Х	Х				Transfer Equivalent Data
SHRTRCR					Х	Х				Transfer Class Data
SHRTRIT					Х	Х				Transfer School Data
SOBCACT					Х	Х				SORLCUR active indicator validation
SOBSBGI			Х							ETS Address Data
SORBTAG			Х							ETS Calendar Data
SORDEGR					Х	Х				Previous Degree Data
SORLCUR					Х	Х				Concurrent Degree Data
SORLFOS					Х	Х				Field of Study Data
SORTEST					Х	Х				Test Score Data
SMRPRLE					Х	Х				Program codes (As of DW4.0.1)
SPRIDEN				Х	X	X				Primary Name Data
SSBSECT					X	X				Schedule Master Data
SSRATTR			1	1	X	X	1			Student Attributes by CRN and Current Term
SSRMEET	Х		1	1	1	1	1			Section meeting times (As of DW4.0.1)
STVACCL			1	1	1		Х			Calendar Codes (UCX-STU346) used by WebTreOer (As of DW4.0.2)
	1									
Banner Table	Course Catalog	Course Equivalents	ETS (transfer)	Advisors / Staff	Students	Applicants	UCX	Banner General	TreQ / WebTreQer	Description
-----------------	----------------	--------------------	----------------	------------------	----------	------------	-----	----------------	------------------	--
STVACYR										Catalog Year Codes (UCX-STU035)
STVATTR	Х									Attribute Codes - used by CourseLink (As of DW4.0.4)
STVCLAS							Х			Student Level Codes (UCX-STU305)
STVCOLL							Х			College Codes (UCX-STU560)
STVCSTA	Х	Х								Course Status Codes
STVDEGC							Х			Degree Codes (UCX-STU307)
STVGMOD							Х			Grade Types (UCX-STU356)
STVLEVL							Х			School Codes (UCX-STU350)
STVMAJR							Х			Major Codes (UCX-STU023),
										Minor Codes (UCX-STU024),
										Concentration Codes (UCX-STU563)
STVNATN			Х							ETS Foreign Country Codes
STVNCRQ					Х	Х				Non Course Codes
STVNCST					Х	Х				Non Course Status Codes
STVQPTP					Х	Х				Non Course Exam/Paper Codes
STVRSTS					Х	Х				Course Registration Status
STVSBGI			Х		Х	Х	Х			ETS School Names
STVSTST					Х	Х				Student Status (DW student type)
STVSTYP					Х	Х	Х			Student Type (DW Student Status Codes UCX-STU306)
STVSUBJ							Х			Discipline Codes (UCX-STU352)
STVTAST									Х	Transfer Articulation Course Status (As of DW4.0.4)
STVTERM	Х	Х			Х	Х	Х			Term Codes (UCX-STU016)
SURVERS								Х		Version of Oracle being used
TWGBWSES								Х		Banner Self Service

TreQ Customers: If you have purchased TreQ, write access is required for the following Banner tables:

Banner Table	Description
SHRTRIT	Transfer Institution (As of DW4.0.4)
SHRTRAM	Attendance Period by Transfer Institution (As of DW4.0.4)
SHRTRTK	Transfer Institution Transfer Course Taken (As of DW4.0.4)

Function Access

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

If using Banner Self-Service single sign-on for DegreeWorks, grant execute on the following packages/procedures to the DB_LOGIN_BANNER user. The DB_LOGIN_BANNER user must also be granted create public synonym and drop public synonym privileges in order to install dwssbfaculty.sql and dwssbstudent.sql (See the Self-Service Banner section in this document).

TWBKWBIS TWBKFRMT BWLKOIDS BWLKOSTM BWCKFRMT BWLKILIB BWCKLIBS

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 identified in the DGW_Technical_Guide_Banner_Considerations document. You must also grant execute privilege to the Banner function f_class_calc_fnc and package twbkbssf.
- 2) On the DegreeWorks server, update the \$ORACLE_HOME/network/admin/tnsnames.ora entry and add the new Banner database.
- On the DegreeWorks server, edit the file dwenv.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 dwenv.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

```
echo $DB_LOGIN_BANNER
```

and you should see the new entry from dwenv.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 in the Banner database. To verify that you are looking at the correct Banner database issue:

SQL> select * from global_name;.

- 6) Restart the servers using the webrestart and daprestart commands
- 7) Determine if the data in 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 the 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 the model that the model of 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:

echo \$DB_LOGIN_BANNER

The result will be similar to the following:

DB_LOGIN_BANNER=BANNER_USER/BANNER_PW@BANNER_SERVICE

- 2) cd to the \$DGWHOME/app/sql directory
- 3) Log into SQL*Plus in the DegreeWorks database by issuing the db command. Execute the bannerlink.sql script:

db SQL> @bannerlink BANNER_SERVICE BANNER_USER BANNER_PW SQL> exit

Note that the bannerlink script will attempt to drop the database link, so you may ignore the Oracle warning ORA-02024: database link not found.

Banner Workflow Integration (optional)

(As of DW4.0.2)

Installation Guide

Follow these steps in order to install and enable Banner Workflow integration with DegreeWorks for the sample models: exception petition process and planner approval process. We provide guidance and suggestions on Banner Workflow configuration here. But, it is assumed you are familiar, in general, with setting up Models, Business Events, Business Processes and Business Components in Workflow and you should refer to Banner Workflow documentation for more information.

Once the sample Workflow models are installed and set up you should inspect and customize the models before testing or using them. In particular you should inspect the activities, notifications and emails defined in each of these workflow models to customize Roles, Performers and email addresses. The "from" address in all notifications and emails used in these models is the email set for the Workflow "admin" user. These samples also use Workflow Roles: Approver, Academic Advisor and Academic Dean. You should be aware the users and email addresses associated with these Roles because they will be the ones notified to approve DegreeWorks requests if you do not modify the models. Most likely, you will want to customize the models to change which Roles/Users are responsible for performing the approval actions and receiving Workflow notifications.

Log in to the DegreeWorks host server

- 1. Load Workflow related packages into the DegreeWorks database.
 - a. cd app/sql
 - b. db
 - c. @wf_parameters_pkg.sql

This file contains package wf_parameters and procedures F_PetitionParams and F_PlannerParams. This package can be customized in order to change the parameters that are sent to Banner Workflow model DW_PETITION and DW_PLANNER. The parameters that are extracted from the DegreeWorks database by these procedures needs to match up with the parameters that the Workflow models expect. So, if the "context parameters" of the model are customized, then these procedures probably need to be customized too.

- d. @wf_updates_pkg.sql
 This file contains a package named wf_updates and procedures P_UpdatePetition and P_UpdatePlanner. These procedures would probably not need to be customized.
- Enable Banner Workflow integration using SureCode.
 For exception petitions: CFG020 WORKFLOWPETITION "Enable Petition Workflow"
 For SEP Planner: CFG020 WORKFLOWPLANNER "Enable Planner Workflow"
- 3. Review settings in CFG020 WORKFLOWWSDL to set the url to your Banner Workflow server WSDL for web services location.
- 4. Review settings in CFG020 WORKFLOWCREDENTIALS to set a username and password for your Banner Workflow server. Commonly, "wfwebservices" would be the username.

Log in to the Banner Workflow host server

- 5. These task needs to be completed by a system administrator who can reconfigure and restart the Workflow server. Upload these two files from your DegreeWorks environment to your Workflow server. They will be located in sis_Banner/export/Workflow directory under the updates (installer root) directory.
 - a. dw_petition.zip
 - b. dw_planner.zip
- 6. Execute these two commands to import the sample DegreeWorks workflow models. Substitute a valid workflow username and password (wfroot for example) on your system as well the actual location of the files you uploaded in the previous step.
 - a. \$WFHOME/bin/import username password dw_petition.zip
 - b. \$WFHOME/bin/import username password dw_planner.zip
- 7. Edit \$WFHOME/config/configuration.xml to add a DataSource for the DegreeWorks database the the "DataSources" section. An example DataSource setup follows in which you should substitute: your actual DegreeWorks server url for "your.degreeworks.server.edu", the actual port number the DegreeWorks oracle listener is running on for "1521" and your actual oracle listener handler for "YOUR DEGREEWORKS SID".

```
<DataSources>
<!-- ... other existing data sources should not be modified... -->
<DataSource name="DegreeWorks">
</url>jdbc:oracle:thin:@your.degreeworks.server.edu:1521:YOUR_DEGREEWORKS_SID<//Url>
</username>yourusername</Username>
</Password>yourpassword</Password>
</DataSource>
</DataSources>
```

- a. After editing configuration.xml run: \$WORKFLOW_HOME/bin/wftool uploadconfig
- b. Use the engineconsole and startengine scripts to restart the engine node(s).
- c. Restart the OC4J instance(s).

Log in to the Banner Workflow web environment as admin user

- 8. Create the Product Type needed for connecting to the DegreeWorks database. As an administrator using Workflow in your web browser, navigate to Workflow System Administration > Product Types. Click "Add Product Type". Give it the name "DegreeWorks", Version 1 (one), and choose the Data Source "DegreeWorks" from the drop-down list. This depends on the naming of the DataSource and successful restart of the Workflow server in step 7.
- 9. Set up the Business Components that the sample Workflow models depend on for calling the stored procedures installed earlier (in wf_updates_pkg.sql).
 - a. Select "Business Component Catalog"
 - b. Click "Add New Category"
 - c. Name the category anything you like, we suggest using the name "DegreeWorks". Save Category.
 - d. Click "Add Component"
 - e. The name must be "dw_updatepetition" (without quotes, of course).
 - f. Select the category "DegreeWorks" from the drop-down list.
 - g. Set Component Type to "Internal".
 - h. Set Product Type to "DegreeWorks" (this depends on setup of the Product Type step 8.)
 - i. Set Technology Type to "Stored Procedure".
 - j. Set Status to "Active".
 - k. Set Release ID to "1".
 - Add a client launch parameter named "procedure" and set the value to: wf_updates.P_UpdatePetition(@DW_UNIQUEID, @APPROVAL_DECISION)
 - m. Add two parameters, named "DW_UNIQUEID" and "APPROVAL_DECISION".
- 10. Repeat step 11 for another Business Component.
 - a. The name must be "dw_updateplanner" (without quotes, of course).
 - b. Select the category "DegreeWorks" from the drop-down list.
 - c. Set Component Type to "Internal".
 - d. Set Product Type to "DegreeWorks" (this depends on setup of the Product Type step 8.)
 - e. Set Technology Type to "Stored Procedure".
 - f. Set Status to "Active".
 - g. Set Release ID to "1".
 - h. Add a client launch parameter named "procedure" and set the value to: wf_updates.P_UpdatePlanner(@DW_UNIQUEID, @APPROVAL_DECISION)
 - i. Add two parameters, named "DW_UNIQUEID" and "APPROVAL_DECISION".
- 11. Create/modify business events for the two DegreeWorks models: Business Events > Business Event Definitions > Click "Add Business Event Definition".
 - a. Set "Name:" to anything you like, we suggest "DW_PETITION". Whatever name you set, it will also have to be set to the same name on CFG020 WORKFLOWPETITION "Petition Event Name". Click "Save".
 - b. Add Event Parameters for "DW_PETITION" as follows and set them all to Type="Text" and Guaranteed="No". On the topic of "Guaranteed" options: if you change any data that you want to be "optional" in dw_parameters_pkg.sql, then set those parameters to Guaranteed="No". STUDENT_EMAIL, USER_EMAIL, CREATE_DATE, NOTE_TEXT, STUDENT_ID, STUDENT_NAME, UNIQUE_ID, USER_NAME. (As of DW4.0.6)
 - c. There are two optional parameters that you can define and modify dw_parameters_pkg.sql to implement. They control whether either or both approvers in the sample model should be ignored.

Define and set APPROVAL1_REQUIRED=0 to disable approver1. Define and set APPROVAL2_REQUIRED=0 to disable approver2.

- d. Still on Business Event Definition, click "Add Workflow Association". Select the value from the drop-down list provided for your workflow model and version.
- e. After you click "Save" you will then map all of the business event parameters to the context parameters within that version of the workflow model. You must map all "guaranteed" parameters. When finished click "Save Parameter Mappings".
- 12. Repeat step 11. for another event named "DW_PLANNER" which has to match on CFG020 WORKFLOWPLANNER "Planner Event Name" and have parameters: DESCRIPTION, STUDENT_EMAIL, STUDENT_GOALS, USER_EMAIL, CREATE_DATE, NOTE_TEXT, STUDENT_ID, STUDENT_NAME, UNIQUE_ID, USER_NAME. Again, parameters named APPROVAL1_REQUIRED and APPROVAL2_REQUIRED are optional and work the same way as explained above.
- 13. Create/modify business processes for the two DegreeWorks models: Enterprise Management > "Add Business Process". We suggest you name the business processes using the same token as the Business Events: DW_PETITION and DW_PLANNER.
 - a. Set Status=Active.
 - b. Click "Add Workflow Association" and choose the appropriate workflow model/version.
 - c. Click "Add Event Association" and choose the Business Event name you defined in steps 11/12.
 - d. Authorized Initiators can be left empty, it is optional.
 - e. Click "Save Process".
- 14. Verify that you have these two models installed by opening them in the Workflow Modeler: DW_PETITION and DW_PLANNER.
- 15. If you want to customize the models immediately, use the modeler to make a "new version" or "create a copy". Refer to Banner Workflow documentation for more information on using the modeler.
 - a. When customizing the models, if you change any "context parameters" then you will have to update the pl/sql where those parameters are extracted from DegreeWorks in app/sql/wf_parameters_pkg.sql (and reload that package into the DegreeWorks database).
 - b. You will also have to modify the business event parameters you defined later in steps 11-12 to match the parameters names given in the model context parameters and in the pl/sql package.

Managing Plan/Petition Approval – two tools

DegreeWorks also provides tools to allow your users to approve/reject plans and petitions. However, if you are using Banner Workflow to manage plans and petition approval it would be best to not use those tools in DegreeWorks. Using both mechanisms to approve/reject plans will most likely lead to confusion on campus about the approval process. Specifically, the Manage tab under the Planner tab can be used to approve/reject submitted plans and Exception Management supports a way to approve/reject petitions. Both of these should be turned off if Banner Workflow is being used to manage plan and petitions approval.

Please see the Exception Management and Student Educational Planner sections in the Web User Guide for more information.

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 one 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 "cpipconnector", 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 casesensitive.
 - 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:
 - 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
 - configman -s es.degreeworks.configURL
 "http://your.luminis.server:8008/cpipconnector/degreeworks/GetConfigVersion2" (Replace "your.luminis.server" with the actual address of your luminis web server. If your cpipconnector 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.
- Ensure that appropriate access permissions are set on those files: chmod 755 degreeworks.properties chmod 755 degreeworks.xml

These files are available from the CSC in the tar files: DegreeWorks_LuminisGCF_3.3.tar.gz -- Version 3.3 DegreeWorks_LuminisGCF_IV.tar.gz -- Version 4 To untar them, use the following command: gzip -dc DegreeWorks_LuminisGCF_IV.tar.gz| tar xf -

- 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 "degreeworks.yoursite.edu" with the actual address of your degreeworks server/url. Where appropriate, replace http with https if SSL is used.
 - a. http://degreeworks.yoursite.edu/pickup.html
 - b. degreeworks.externalSystemURL = http://degreeworks.yoursite.edu
 - 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 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|access_code"
- For instructions on bulk loading of DegreeWorks users into Luminis LDAP, review the section "Load DegreeWorks credentials into Luminis LDAP" later in this document and Luminis GCF Implementation Guide (lsdk0907im.pdf) section titled "SET THE CPIP EXTERNAL ACCOUNT FOR EXTERNAL SYSTEM USERS"

Luminis Bookmark Channel

(As of DW4.0.2)

All of the following instructions take place using the Luminis web interface for administrator.

 In order to set up a Luminis Bookmark 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 degreeworks.yoursite.edu/dw2/ locations with appropriate values. Also, if SSL is used, replace http in http%3A//degreeworks.yoursite.edu/dw2/ with https.

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

Luminis CPIP Inline Channel

(As of DW4.0.2)

All of the following instructions take place using the Luminis web interface as an administrator.

- 1. Select Portal Admin > Publish a new channel. Select "CPIP Inline Frame" and then click "Next".
- 2. Fill in the Channel Type fields (Title, Name, etc) as you like. Click "Next".
- 3. On the General Settings page, enter the External System ID that you used in step 3. of "Luminis System Configuration" above.
- 4. Still on the General Settings page, enter a Destination URL using the following sample format and substitute "mydegreeworks.edu/IRISLink.cgi" for the actual url to your DegreeWorks web server including the path to IRISLink.cgi.

http://mydegreeworks.edu/IRISLink.cgi?SERVICE=SCRIPTER&SCRIPT=SD2GETMYAUDIT&AC TION=REVAUDIT&REPORT=WEB31&ContentType=xml

- 5. On CPIP Inline Frame Parameters, do not check any of the checkboxes.
- 6. Follow through the rest of the configuration pages setting the Categories, Roles, etc as needed and as appropriate for your site configuration. Then click Finish.

User Guide

Two Luminis SSO options

You have two options with regard to how DegreeWorks verifies the Access Id and Access Code it receives when the Luminis SSO occurs.

- 1. Have DegreeWorks validate against the Luminis LDAP credentials
- 2. Load the DegreeWorks SHP credentials into Luminis LDAP for Luminis to pass to DegreeWorks when the SSO occurs.

Using the Luminis LDAP Credentials (option 1)

(As of DW4.0.2)

Under this first option the actual Luminis LDAP credentials will be sent to DegreeWorks. Instead of verifying the login data passed against its SHP records DegreeWorks will use its LDAP call-out feature to validate the ID and password against Luminis LDAP – where they came from in the first place.

Please be sure to review the Security section in the DegreeWorks Technical Guide and also the UCX-CFG020 LDAP, LDAPDN and LDAPSERVER sections in the DegreeWorks Technical Guide UCX.

To configure Luminis to send its LDAP credentials to DegreeWorks when a SSO occurs you need to set two values in degreeworks.properties:

= true

degreeworks.cpipconnector.getconfig.sendlogin

degreeworks.cpipconnector.getconfig.usePDSCredentials = true

This will tell Luminis to send its LDAP credentials instead of those from the degreeworks external system when the Luminis SSO occurs.

In addition you need to setup the UCX-CFG020 LDAP records in DegreeWorks to point to where the Luminis LDAP credentials reside.

UCX-CFGO20 LDAP

The **User RDN** setting should be set to "uid". This is the actual user-id used when the user logs into Luminis. DegreeWorks uses this to locate the user's credentials.

The **Attribute User ID** should be set to pdsExternalSystemID – but only if that contains the user's SPRIDEN-ID – as it is the RAD-ID used in DegreeWorks. If some other attribute houses the user's SPRIDEN-ID then use that setting instead.

The Attribute User Class can be set to pdsRole but this field must contain a valid DegreeWorks userclass as defined in UCX-CFG012 – such as STU, ADV, etc. If pdsRole is not suitable you can use another LDAP attribute to house this user-class. If you can't house the user-class in LDAP you may leave this Attribute User Class field blank so that DegreeWorks uses the user-class that was set during the bridge extract.

UCX-CFG020 LDAPDN

The LDAPDN setting should look something like this:

ou=people,o=somewhere.edu,o=cp

where the somewhere.edu will be replaced with a valid value for where Luminis resides on your network.

Set the Enable LDAP flag to "Y".

Be sure OpenLDAP is installed on the DegreeWorks server and that the DegreeWorks software was built with the dwenv.config DWLDAP value set to 1. If the DWLDAP setting is not in dwenv.config, you can add it to the Build Environment Variables section as follows:

 $\#\ {\rm DWLDAP}$ - Set DWLDAP to 1 to compile support for LDAP authentication export DWLDAP=1

After saving this change, re-login to the unix server to reset this variable, then issue "build all" so that the DegreeWorks software is compiled with this setting.

When testing be sure to review the logdebug/web.log file to make sure the correct ACCESS_ID and ACCESS_CODE are being passed.

The second option to use is where the SHP logon information is copied into the DegreeWorks external system in Luminis LDAP. An explanation follows on the next page.

Load DegreeWorks credentials into Luminis LDAP (option 2)

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="degreeworks|' || 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, 'YYYYYMMDD') dt from dual;
    spool cptool_cmd_input.txt
    @luminis_user_dw_creds.sql;
    spool off
    exit;
```

- 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="degreeworks|911199999|password"
set user adam_blue ExternalAccount="degreeworks|911199994|password"
set user adelina.green ExternalAccount="degreeworks|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. Grant the necessary privileges to your DegreeWorks user in the Banner database, as follows:
 - a. Type dbb to start sqlplus in the Banner database
 - b. To display and verify your DegreeWorks user name type "show user" and enter:
 - SQL> show user USER is "DWMGR"
 - c. Log into sqlplus in your Banner database as a DBA account and issue the following grants to your DegreeWorks user (note that some of these may already be granted):

grant create procedure to dwmgr; grant create public synonym to dwmgr; grant execute on f_class_calc_fnc to dwmgr; grant execute on TWBKBSSF to dwmgr; grant execute on TWBKWBIS to dwmgr; grant select on SFRSTCR to dwmgr; grant select on SPRIDEN to dwmgr; grant select on STVTERM to dwmgr; grant select on TWGBWSES to dwmgr;

And include the following if you will also be using single DW sign-on for faculty: grant execute on BWCKFRMT to dwmgr; grant execute on BWCKLIBS to dwmgr; grant execute on BWLKILIB to dwmgr; grant execute on BWLKOIDS to dwmgr; grant execute on BWLKOSTM to dwmgr; grant execute on TWBKFRMT to dwmgr;

- Find dwssbstudent.sql file in the app/sql directory on the DW host server (cd app/sql). Use the shortcut script dbb to connect to your Banner database in sqlplus, and then run the file by issuing @dwssbstudent.sql. This will create the package called "DW_Student" in the database.
- 3. 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.
- 4. 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.
 - a. Page Name: DW_Student.P_SignOn
 - b. Description: DegreeWorks. Or, enter whatever text you would like to appear as the description.
 - c. Module: Student Self-Service. Or, select the appropriate module for your site.
 - d. Enter page title, header text, and header graphic as you prefer.
 - e. 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.

- f. Choose Associated Roles according to your preference.
- 5. 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.
- 6. 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.
 - a. Go back to WebTailor > Web menus and Procedures.
 - b. Select the menu where you want to add DegreeWorks (for example the main menu <u>bmenu.P_GenMnu</u> or the student main menu <u>bmenu.P_StuMainMnu</u>)
 - c. Click "Customize Menu Items"
 - d. 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).
 - e. Enter the URL: DW_Student.P_SignOn
 - f. Enter the link text, description and sequence number according to your preference.
 - g. Check the box next to Database Procedure.
- 7. Configure WebTailor Parameters
 - a. DWLINKTEXT The text you want to display as the link to DegreeWorks
 - b. DWURL The url to the DegreeWorks cgi. For example https://yourserver.edu/degreeworks/IRISLink.cgi.
 - c. 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 memebers.

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 "Enable SSB Sign-on" = "Y"
- 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)
ł
// 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 need 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.

Special Topics

To help you use DegreeWorks effectively, there are a variety of special topics that can warrant discussion and elaboration. These topics are typically generated from customer feedback when it becomes clear that an extended explanation is needed on some specific issue. The topic often references other documents that contain the specifics of configuration. The special topic can take the form of an abbreviated "how to" document.

Applicants in DegreeWorks

For DW 4.0.2 and beyond there is now an Applicant extract that can be executed to allow students who have an applicant record, but may not yet be considered a current student to be bridged into DGW. For these students if they have transfer courses, test scores, or other appropriate data in Banner their data will be bridged over into DGW.

Advisors and REG users can see applicants within DGW just as they see other "regular" students. Applicants must log into DGW from within SSB or Luminis, so they must have at least that access on the Banner side. Applicants must have at least applied to the school and have a SGBSTDN, SARADAP, or SORLCUR/SORLFOS record to be able to be extracted and thus be able to log into DGW.

There is not a universal "GUEST" applicant extract or login* that recruiters or admissions officers can use to log potential students into DGW.

*Unless the school has created a GUEST type user which they can then use to access DGW. This is a topic for another discussion though.

How to extract applicants:

- Use the REFRESH button (only will work if UCX-CFG020/BANNER->Process_Applicants = "Y")
- 2. Use the command line or cron

\$bannerextract applicant *applicants* Where *applicants* can be:

- SQL file
- ID file
- Individual ID
- 3. Transit

Select Applicant as the extract type Must use an SQL file, cannot use the selection criteria

Configuration Flags:

There are a few configuration flags that will need to be set using SureCode in CFG020/BANNER:

Process Applicants – Y/N – Setting this flag to "Y" will allow the applicant extract process to happen when the REFRESH button is pressed. The applicant data will be looked up in addition to the student data.

Process Both Goals – Y/N – Setting this flag to "Y" loads both the "LEARNER" and the "ADMISSIONS" data from SORLCUR/SORLFOS into DGW. If it is set to "N", then "ADMISSIONS" data is not loaded into DGW if "LEARNER" data is found.

Extract Process

When the REFRESH button is pressed the following extract process is followed:

- 1. The SORLCUR/SORLFOS records are checked for a "LEARNER" record using the SORLCUR query within the bannerextract.config file. If a "LEARNER" record is found, then the variable LEARNER FOUND is set to "Y".
- If UCX-CFG020/BANNER->Process_Applicants = "Y" then if LEARNER_FOUND = "N" or if LEARNER_FOUND = "Y" and UCX-CFG020/BANNER->Process_both_goals = "Y" then the SORLCUR/SORLFOS records are searched for an "ADMISSIONS" record using the SORLCUR2 query in bannerextract.config. If an "ADMISSIONS" record is found, then the variable ADMISSIONS FOUND is set to "Y".
- 3. Regardless of what happens in steps 1 & 2, the SGBSTDN record is looked up. If found, the variable SGBSTDN_FOUND is set to "Y".
- 4. One of the following two paths will be followed:
 - a. If the variable ADMISSIONS_FOUND = "Y", then the associated SARADAP record will be looked up based on the associated values found in SORLCUR
 - b. If ADMISSIONS_FOUND = "N" and UCX-CFG020/BANNER->Process_Applicants = "Y" and UCX-CFG020/BANNER->Load_SARADAP_GOALS = "Y" the SARADAP record is looked up based on the SARADAP query in bannerextract.config.

When the bannerextract applicant is executed from the command line or a cron job or if Transit is used to extract the applicants, the following process is followed:

- 1. Same as above. The LEARNER_FOUND flag is set to "Y" if found
- If LEARNER_FOUND = "N" or if LEARNER_FOUND = "Y" and UCX-CFG020/BANNER->Process_both_goals = "Y" then use the SORLCUR2 in bannerextract.config to search for "ADMISSIONS" records. If found set ADMISSIONS_FOUND = "Y".
- 3. Same as above
- 4. One of the following two paths will be followed:
 - a. If the variable ADMISSIONS_FOUND = "Y", then the associated SARADAP record will be looked up based on the associated values found in SORLCUR
 - b. If ADMISSIONS_FOUND = "N" and UCX-CFG020/BANNER->Load_SARADAP_GOALS
 = "Y" the SARADAP record is looked up based on the SARADAP query in bannerextract.config.

Once extracted at least one of the variables (ADMISSIONS_FOUND, LEARNER_FOUND, SGBSTDN_FOUND, or SARADAP_FOUND) will have to be "Y" or an error will result. Then one and possibly up to three of the following paths will be taken (one path from the two student type paths, one path from the two applicant type paths, and one from the dual degree path):

1. (Student Path)

If LEARNER_FOUND = "Y" The goal (degree) records are created from the SORLCUR/SORLFOS records

2. (Student Path)

If LEARNER_FOUND = "N" and ADMISSIONS_FOUND = "N" and SGBSTDN_FOUND = "Y" Load goal records from SGBSTDN records

3. (Applicant Path)

If ADMISSIONS_FOUND = "Y" and the Level(school)/Degree combination does not match any of the LEARNER Level(School)/Degree combinations

Load goal records from the ADMISSIONS version of the SORLCUR/SORLFOS records 4. (Applicant Path)

If ADMISSIONS_FOUND = "N" and SARADAP_FOUND = "Y" and UCX-CFG020/BANNER->Process_Applicants = "Y" and UCX-CFG020/BANNER->Load_SARADAP_goals = "Y" and the Level(school)/Degree combination does not match any of the LEARNER Level(School)/Degree combinations

Load goal records from SARADAP

5. (Dual Degree Path)

If UCX-CFG020/BANNER->Check_dual_degree = "Y" and SGBSTDN_FOUND = "Y" and the Dual Level(school)/Dual Degree combination does not match any of the LEARNER Level(School)/ Degree combinations

Load goal records from the SGBSTDN dual degree records

Bannerextract.config file

There are three areas in the bannerextract.config file that need to be examined to determine if they are extracting the appropriate applicant data:

- 1. SORLCUR2
- 2. SORLFOS2
- 3. SARADAP

Applicant User Class

An applicant user class (APP) must be created in the SHPCFG file. This will be the lowest class. If the class does not already exist in AUD012, it will need to be added there.

```
#-----
#-- DegreeWorks keys for applicants
                      -----
#-----
if (DGWUSERCLASS = "APP") then
addgroup = SRNAPP
```

By default, the SRNAPP group has the following accesses:

_ _ _ _ _ _

SDAUDREV SDLOKAHD SDSTUME SDWEB31 **SDWHATIF SDWORKS** SDXML31 **SDAUDPDF**

Using Banner Data to create Scribe Custom Data

There may be a time when a rule needs to be scribed against a variable from Banner that is not by default bridged into DegreeWorks. Some examples of this type of variable includes graduation status, academic standing, and campus code. Follow the procedure below to set up and use these types of variables.

- Create the variable in the BAN080. In this table you will indicate the column, table, and where statements to retrieve the variable from Banner. The following shows an example of the code set up for the academic standing code. You pick a name for this variable. We will call it ACSTCODE. Note: if you choose a table that is not a typical table DGW uses, you must make sure your DBA gives the DGW user read access to this table.
 - a. Create a record in BAN080 with the key of ACSTCODE:TABLE. The Value1 should be the table name you are retrieving from in this case, SGBSTDN.

	22 BANDOD	_(#) X	s (Compatibulity Mode) - Advancedt Ward	100
	KEY ACSTCUDE-TABLE			
				Minute (D) - D
West Internet The Same Neuror Same		aBb	CEDE ABBECEDE ABBE ABBE ABBE ABBE	and the second s
	Valet Luder de a	Nor	mai 1 No Spasi Heading 1 Heading 2 Title Change	Share As Webbs
Vexes Vexes Vexes Vexes See to construct the second building of the second	tool particular	1.000	Oniza Styles	Extense Webby
Vale4 State State <td< td=""><td>48962</td><td>127</td><td>d Higher Education</td><td>- IOI × I</td></td<>	48962	127	d Higher Education	- IOI × I
Valad Solation Costant Revealed Website Costant Revealed Website Costant Revealed Website Costant Revealed Costant Revealed Costant Revealed Costant Revealed Costant Revealed Costant Revealed Costant Cost	Value3			
State Description: Descript	Yalue4	D.	8	2
Custo Image: Cost of C	Status 🗍	10	Descriptione Dynamic SQL Definitions	eer ACSTCODE TABLE
Investor Value	Cuttom		Lunnet week and an and an and an and	account account account a stat
Who Note of control will (cold will and cold will and co	Environ .	pone	ADDIDE ADDIDE ADDIDE ADDIDE ADDIDE ADDIDE	scronel accord annuel a 1141
Weiter Tatel, Coll, 2017 All and tatel tatel, Coll, 2017 Prior Boarton B All and tatel tatel, Coll, 2017 Prior Boarton B All and tatel tatel, Coll, 2017 All and tatel tatel, Coll, 2017 All and tatel tatel, Coll, 2017 All and tatel tatel, Coll, 2017 All and tatel tatel, Coll, 2017 All and tatel tatel, Coll, 2017 All and tatel tatel, Coll, 2017 All and tatel, tatel, Coll, 2017 All and tatel tatel, Coll, 2017 All and tatel, tatel, Coll, 2017 All and tatel tatel, Coll, 2017 All and tatel, tatel, Coll, 2017 All and tatel tatel, Coll, 2017 All and tatel, tatel, Coll, 2017 All and tatel tatel, Coll, 2017 All and tatel, tatel, Coll, 2017 All and tatel tatel, Coll, 2017 All and tatel, tatel, Coll, 2017 All and tatel tatel, Coll, 2017 All and tatel, tatel, Coll, 2017 All and tatel tatel, Coll, 2017 All and tatel, tatel, Coll, 2017 All and tatel, tatel, Coll, 2017 All and tatel, tatel, Coll, 2017 All and tatel, tatel, Coll, 2017 All and tatel, tatel, Coll, 2017 All and tatel, tatel, Coll, 2017 All and tatel, tatel, Coll, 2017 All and tatel, tatel, Coll, 2017		-	Valuet	Value2
Velo Pagreeto Tanding Market Status (2013) 2101 121 2013 2014 2015 2015 2015 2015 2015 2015 2015 2015			(RELECT MAX (E. SCHETCH TEEN CODE EFF)	
Wite 9-100-14.000100_000100_00100_00100_00100_00100_00100_00100_00100_00100_00100_00100_00100_000100_000100_000100_000100_000100_000100_0001000_0001000_000000			FROM SCRETON B	
Mob X. 3713375_00084_0101761_0105_008_0050_001 Mob X. 3713375_0008 9000108_01050_00108_00500_00108_00500_00108_000108_00108_000108_000108_000108_00000000			WHERE D. SOBSTON_FIDM = A. SHRTTRM_FIDM)	
Web 1, 970/375_0004 (r) = 500/3768_0475_0004 Meb 3, 4500/3768_0475_0004 000/3700 000/3700 Meb 3, 4500/3768_0475_0004 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 000/3700 00/3700 00/3700 000/3700 00/3700 00/3700 00/3700 00/3700			AND X.STVASTD_CODE+A.SHETTEN_ASTD_CODE_END_OF_TEL	a i i i i i i i i i i i i i i i i i i i
Wite Page wite Tandiang Description			ARD B.STVASTD_CODE (+) + SCDSTDN_ASTD_CODE	AND A. SHRTTPM_PIDM = SG
Weig Begreat/6 Institute Connection School 11:17:55 Connection		in the second	SUBSTDN_CARP_CODE	
Wite Pagetakli Tansimo Pagetakli			SCRATDN	3
Who Supervisit Supervisit <td></td> <td></td> <td>agbatdn_stat_code</td> <td></td>			agbatdn_stat_code	
Who Page wate Transform Barrowske			sightender m	
Vite Bagreat/6 Vite Bagreat/6 Vite Bagreat/6			a.sphotds_ters_code_eff =	
Who Segretable Segretable <td></td> <td></td> <td>(Select Rar(6.sgbstdn_term_code_eff)</td> <td></td>			(Select Rar(6.sgbstdn_term_code_eff)	
who begreat6 Imaging 6000000000000000000000000000000000000			where h.sgbstdn_pida = a.sgbstdn_pida)	
who Degree 406 Tambing Berreavio			subst.dn_camp_code	
Vho Pegreat/6 Corrector. Tech DWSEED Corrector. Tech			sgöst dn	
Vibo Pagravio Corrector: Tech DWSEED Corrector: Tech DWSEED			sgbstdn_term_code_eff 11 sgbstdn_stst_code	
Who DegreeV6 Tambing 64/09/09 13.17.55			andrat do.	
Connector: Tech DW/SED Terr Num Who DegreeV6 TereSterp Connector: Tech DW/SED			SGPSATT_ATTS_COD#	
Connection Tech/DWSEE0 Date Who DegreeV6 DegreeV6 TamSterp B4709/09 131.17.55 DegreeV6				<u>ت</u> ,
Unit (Unit Point) Who DegreeWe Tanéfing 64709/09 13.17.53			in a manter	and provide provide provide and
Who DegreeV6 TensTamp 04/09/09 121.17.53		-	Connection Tech DWSEED	Free low from from a
Who Degree#6 Tambitup 64/09/09 13 1.17:55			+0+	
Who DegreeV0 TansStarp Get 02.01.17:53				
Who DegreeV6 TeerStep 04/09/09 121.17.53				
Who DegreeNo Tamblings 64/709/09 13:17:53				
Who DegreeVo TamStamp 64/09/09 12:17:53				
Who DegreeV6 TeedStep 04/09/09 131.17.53				
Vho DegreeVo TeedSeep 04/09/09.13117:53				
Who DegreeVo Tansfamp 04/09/09 12:17:53				
Who DegreeV6 TandStapp 64/09/09 13:17:53				
Who DegreeWo 0 TendStep 04/05/05.13:17:53 0 0 0				
Who Degree 49 0 TausStarp 04709709 121:17:57 0 0				
TeielSteine 04/09/09 131:17:53	Who DegreeVo			
	Terestamp 04/09/09 13:17:53		101-05-04	210 100N (A) 7 (4)
Start Decree 2 yabo + Profession - Professio	Start Otors C 22 yebs + Speard C tobars 177	Suref ada (\$ 149.24.2	Sate	

 Next create a record for the column with the key code: ACSTCODE:COLUMN. For this example, we will be retrieving the SGBSTDN_STST_CODE, so that should be entered in the Value1 field.



c. Any SQL statements that will be used to find the desired instance of the variable should now be entered into records with keys beginning with WHERE. For multiple where statements, add multiple where records. These records should be named WHERE_1, WHERE_2, etc. In this example, we will find the academic status value for the latest term. The SQL to accomplish this will be:

Where a.sgbstdn_term_code_eff = (Select Max(b.sgbstdn_term_code_eff) from Sgbstdn b where b.sgbstdn_pidm = a.sgbstdn_pidm)

In this example, four records will need to be created. They can be named: ACSTCODE:WHERE_1, ACSTCODE:WHERE_2, ACSTCODE:WHERE_3, and ACSTCODE:WHERE_4.

Screen shots of these are as follows:

D B S X 2 B D	*	-	1		
- + H Table: (SANOSU	Description: Opnamic Stat. De	and one	Keg: ACSTCODE WHERE 1		anna fi san an Fisana a
nges vmbeits vmbeits vmbeit	4 wnooi2 wnooi8 wnoos3 wno	029 AUD034 AUD047 BAND	eu schoos schoos studie studza studza stu	036 \$TU307 \$TU316 \$TU323 \$TU324 \$	TU350 STU352 STU5.4
Key	[Value]		Value2	Value3	
ACADSTSGD:WHERE_R	MHERE B. SGESTON_PIDM =	A. SHRTTPH_PID(H)			
ACADSTSCU: WHERE_9	AND X. STVASTD_CODE=A. SI	RETTER_ASTD_CODE_END_OF	787A		
ACADGTSCD:WHERE_90	AND D.STVASTD_CODE (+)	* SCBSTDB_ASTD_CODE	AND A. SHRITHM_FIDH = SCREEDN_FIDH		
ACADSTTTR: COLUMN	SHETTER_TEER_CODE X	STANDID DESC			
ACADSTITS, ODDERBY					
ACADSTITE: DEPOST	and a state of the second state of the second	An an and the state of the stat			
ACADSTITSTABLE	ERDITION A, SCRETCH, ST	VASTD X, ETVASTD B			
ACADSTTTR: WHERE_1	A GITATTAL SELANDOO		and the second		
ACADSTTTR: WHERE_2	(SELECT)	PER DISCOURTER			
ACADOTTTP:WHEPE_D	FROM SHOT	BET. INCOTOUR MI	····		
ACADSTITE: WHERE_4	WHERE A.2	H 4 F	* × • • *		
ACADSTTTR: WHERE_S	AND SCREI		or an		
ACADSTTTR: WHERE_6	ISBLECT P Votert la sun	the tarm code aff a			
ACADSTITE: WHERE_7	FROM SCRS				
ACADSTTTR: WHERE A	MITER B. (Value2				
ACADSTTTP: WHERE_9	AND X. STL Volue3			1	
ACADOTTTP: WHERE_90	AND P. UTL				
ACAMPUS: COLUMN	SUBSTEM_L Value4				
ACAMPUS: REPORT	Status				
ACAMPUS TABLE	SCBSTON		-2		
ACSTCODE: COLUMN	egbet.dn i				
ACSTCODE: TABLE	signerative a Revenion				
ACSTCODE:WHEDE_1	e, sgb st dr				
ACSTCODN: WHENE_2	(Uelect				
ACSTCODE: WHERE_D	from SGL				
ACSTCODE: WHXEE_4	where b.:				
CAMPOS COLUMN	sgbatdn_c				
CAMPUS. TABLE	s glest da				
DESTYPE: COLUMN	sghut dn_t				
DCSTYPE: REPORT	and the second se				
DESTYPS: TABLE	s gb st.dn				
EDSCREATT: COLUMN	SCRSATT_/ Who	DegraaWo			
EDSGESATT: REPORT	Tendring C	04/03/05 03:47:44			
FE SCREATT: TABLE	australia + TTADEOR	04/05/05 05:47:44			
PE-SCREATT . WHERE_1	SCRAATT_				
EDSCREATT: WHERE 2	('IN', 'UN', 'EB', 'IN	, DAF, DAR, BLNT, S	(V')		
MDSGRSATT: WHERE_D	and A. SGRSATT_TEPH_CO	DE_EFF = (DELECT			
EDSGRSATT: WHIRE_4	BAT (D. SGPSATT_TREM_C	001 114)			
PDSCREATT: WHERE_S	FROM SCREATT D				
KDSGRSATT: WHERE_6	WHERE A. SCESATT_PIL	ON = B.SGRSATT_PIDM)			
EDIESTSC: COLUMN	substdn_coll_code_1 !!	sortest_tesc_code ii	sortest_test_score		
EDTRETEC: OEDREFY	sortest_test_date				
EDTESTEC : REPORT					
EDIRSTEC: TABLE	sgöstdn a, sortest b	- 1000 DUC			
POTROTOC: WHEDR_1	a.sgbstdn_pida = b.sort	cest_pids and			
					3

🔊 Start 🔯 Office Coemuni... 😴 Yahool Hessen... 🕞 Inkox - Hicroso... 😻 Lotto Texas W... 😻 Radio Hargerta... 🔄 To create a Ban... 🔯 Surect ode 🛛 🔯 🖓 🕲 🖉 🖉 🖉 🖉 🖉 🖉 🖉 🖉 🖉 👔 🖉 🖉

• • • • • • • • • • • • • • • • • • •	OBSX200	1	
Audor12 Audor13 Audor14 Audor13 Audor14 Audor27 BANK Valaci Valaci Valaci Name Valaci Valaci Valaci Valaci Valaci Audor1711 Valaci Valaci Valaci Valaci Valaci Audor1712 Valaci Audor1712 Valaci Valaci Valaci Audor1712 Valaci Audor1712 Valaci Valaci Valaci Audor1712 Valaci Audor1712 Valaci	+ + H Table BANOSO	Description: Oynamic SQL Defentions Key: ACSTCODE:WHERE_2	
For Value! Value! Value! Value! Value! Addotrow:exclosionance Addotrow:exclosionance Addotrow:exclosionance Addotrow:exclosionance Addotrow:exclosionance Addotrow:exclosionance Addotrow:exclosionance Addotrow:exclosionance <t< th=""><th>He AUDRIZ AUDRIS AUDRI</th><th>14 AUDO15 AUDO18 AUDO27 AUDO29 AUDO24 AUDO27 BANORO SCROOS SCROOS STUDIE STUD23 STUD24 STUD26 STU307 STU307 STU316</th><th> STU323 STU324 STU350 STU352 STU5_</th></t<>	He AUDRIZ AUDRIS AUDRI	14 AUDO15 AUDO18 AUDO27 AUDO29 AUDO24 AUDO27 BANORO SCROOS SCROOS STUDIE STUD23 STUD24 STUD26 STU307 STU307 STU316	STU323 STU324 STU350 STU352 STU5_
Addorson UNIEL 0 Addorson UNI	Keo	Volum1 Volum2 Volum2	
Addentes NUMBER 200 Addentes	ACADOTSGD WHERE 9	AND N. STWASTD CODE-A SHETTEM ASTD CODE END OF TEEM	
ALAUSTITE:OUNDER ALAUST	ACADSTOCE: WHERE 20	AND D. STWANTD CODE (+) - SCRNTEN ANTO CODE AND A DISTTING FIDM - SCRNTEN FIDM	
Academics	ACADUTTTE: COLUMN	SHETTER THER CODE N. STVASTO DESC	
AAAATTE AAATTE AA DEETEN AA DEETEN, STUATE X, STUATE X ACADITIS WEBS I ACADITIS WEBS I ACADITIS WEBS AC ACADITIS WEBS AC ACTION ACADITIS ACADITIS WEBS AC ACADITIS WEBS AC ACADI	ACADSTTTE ORDERSY		
Add20171: 10021 Add20171: 10022 Add20171: 10022 Add201	ACADSTITE, DEPOST		
ALAMOTTS VERSE 1 ALAMOTTS VERSE 2 ALAMOTTS VEL	ACADSTITS, TABLE	SHRITER A. SCROTER, STWASTE X. STWASTE B	
ALAMOTTA WILLIA ALAMOTTA WILLI	ACADSTTTP: WHERE 1	A SHATTON INDU CODE -	
A ALAGOTTA: UNIDES 4: A ALAGOTTA: UNIDES 4:	ACADOTTTR: WHERE 2	(SILACT) ESTIMATION	
ALALATTS: WILES 4. WILE A.L. ALALATTS: WILES 4. (UTLICT) ALALATTS: WILES 4. (UTLICT) BENERALL, A. SUBSCH ALATTS: WILES 4. (UTLICT) BENERALL, A. SUBSCH ALATTS: WILES 4. (UTLICT) BENERALL, M. SUBSCH ALATTS: WILES 4. (UTLICT) BENERALL, WILE 4. (UTLICT) BENERALL, WILE 5. (UTLICT) BEN	ACADOTTTE WHERE 3	PEOR SHOL	
ALADATTS WORL 1 ADD SOT ALADATTS WORL 2 ALADATTS WORL 2	ACADOTTTE: WIERE 4	MARRY A.1 KEY ACSTCODE MIRRE 2	
ALADATTI: UNITSI UNITSI : CHILLY : CHILY : CHILLY : CHILY	ACADOTITS: WHERE S		
ALADYTTS WURSE "	ACADSTTTR: WHERE 6	(SELECT)	
ArAADTT: 000024_0 ArAADTT: 000	ACADSTITS WHERE 7	FROM SCRE	
ALADATTS WIRE 4 ALADATTS WIRE 5 ALADATTS WIRE 5 ALADATO TAUL ALADATTS WIRE 5 ALADATO TAUL ALADATO TAUL ALADATO ALADATO TAUL ALADATO AL	ACADSTITE: WHERE S	WHERE B : Voke1 (Zelect Har(b.sghrtdm_term_code_eff)	
ALADITY: 200704_00 MD 1. 37 ALADITY: 200705 MD 1. 37 ALADITY: 200704 SUBJECT ALADITY: 200705 SUBJECT BUDITY: 200705 SUBJECT BUDITY: 200705 SUBJECT BUDITY: 200705 SUBJECT BUDITY: 200705	ACADOTTTR: WHERE 9	AND X STV Velice2	
ALAMYON COUNDS ALAMYON COUNDS ALAMYON TRUE ALAMYON TRUE A	ACADOTTTE BURRE 90	AND B STL	
A ALMONG JABOAT AALMONG JABOA	ACAMPUS: COLUMN	scasttal (Value3	
A ALMON TABLE OPETIME START AND A CONTROL AN	ACAMPUS: EXPORT	Values	
An TTOOL 10000000 values values of a value o	ACAMETIS TABLE	TRANTIN AND T	
ACTIONS HARS - sebards - Address - A	ACRTCODS (COLUMN	Senar	
Actional sector and a segmental a segmental (sector and actional sector and actional sector and actional sector actional secto	ACSTCODE : TARLE	entest in a Caston	
AdTCOD:HUBE_2 i Claice AdTCOD:HUBE_2 i Claice AddCOD:HUBE_2 i Claice AdTCOD:HUBE_2 i Claice AdTCOD:HUBE_2 i Claice AddCOD:HUBE_2 i Claice AddCOD	ACSTCODE NHERE 1	a schestor Revenue	
ACT CONSINUUUE_0 terms being and the set of	ACSTCODE HOURDE 2	(felect	
ACTIONISHIES 4 Were 5.4 CASTOS: TABLE spin-th_c CASTOS: TABLE spin-th_c DEGUTART. TABL	ACSTCODE MILESE	from SG	
CAMURE CONTINUE Spin-that, Science Control Science Control	ACSTCODE: MIERE 4	Mars b.	
CAUTORS INALE selected and a selecte	CAMPUS COLUMN	substant (
DC1TUPE: Sediment of the sediment of t	CAMPUS TABLE	schot de	
DOCTIFIE 184001 DOCTIFIE 184001 DOCENTIFIE 184001	DCSTVFE COLUMN	schetde t	
DACTYPE TABLE optimum optimum DACTYPE TABLE optimum optimum DACTAGE TABLE SERVET_L DACTAGE TABLE SERVET_L <td< td=""><td>DESTYPE-DEDORT</td><td></td><td></td></td<>	DESTYPE-DEDORT		
UPD/SUBJECT CONTINUE SCOPALT_J UPD/SUBJECT SERVET SCOPALT_J UPD/SUBJECT SCOPEND SubJectsT_SCOPALT_SCOPALT_J UPD/SUBJECT SCOPEND SubJectsT_SCOPALT	DCOTVER-TABLE	arbert do	
Decomposition Decomposition Decomposition Mode Decomposition Decomposition Decompo	ICDSGRSATT: COLUMN	SCREAT /	
NO ODSUATT TABLE DEDATT DEDOCRAATT WORK 1 DEDATT DEDOCRAATT WORK 2 OF 10 Fr. ' DEDOCRAATT	EDSGREATT: REPORT		
Disclatif Disclatif <thdisclatif< th=""> <thdisclatif< th=""> <thd< td=""><td>EDSGREATT: TABLE</td><td>SURPATT / Who DegreeVo</td><td></td></thd<></thdisclatif<></thdisclatif<>	EDSGREATT: TABLE	SURPATT / Who DegreeVo	
DBOGRAATT WEBER 2 ('1B', 'J DBOGRAATT WEBER 2 ('1B', 'J DBOGRAATT WEBER 2 AAAA A ROFAATT TESER COOK NFF = (SELECT DBOGRAATT WEBER 3 ABAC AL CONSTANT TESER COOK NFF = (SELECT DBOGRAATT WEBER 4 DBOGRAATT WEBER 4 DBOGRA	EDSCREATT. WHERE 1	8038ATT TemStamp 04/09/09 09:48:08	
Documentary 1000000000000000000000000000000000000	HE-SCREATT , WHERE 2	(1887, 11	
DD:0054XT: W0184_4 mar(h:0050XT: TENE, COLE IFF) DD:0054XT: W0184_6 sport DD:0154TC: 001000 sport	EDSCREATT WHERE 3	and A. SCREATT TERM CODE EFF = (RELECT	
Noncodestry 100032_0 UPS00 GoodAstry 100 Noncodestry 100032_0 UDS2 & A.Schordstry PDD8 + D.Schorty PDD8) NOTIFIESTC: CONDEND september _ context_set_set_secore NOTIFIESTC: CONDEND september _ context_set_secore NOTIFIESTC: CONDEND september _ context_set_secore NOTIFIESTC: CONDEND sectext_secore NOTIFIESTC: CONDEND sectext_secore NOTIFIESTC: CONDENDIT sectext_secore	IDSGREATT-WHERE 4	BAY/D GGREATT TEM CODE EFF)	
NDOCRATTINUESE_6 WEEPE A.SCREATT_PIDH + D.SCREATT_PIDH) NDOTESTIC:COLUMN rightsdm.coll_code_1 () sortest_test_code () dottest_test_score NDOTESTIC:COLUMN rightsdm.coll_code_1 () sortest_test_code () NDOTESTIC:COLUMN rightsdm.code_test_code () NDOTESTIC:COLUMN rightsdm.code () N	KDSGESATT NULLE 5	FROM SADDATT D	
DD1111121 septentim_coll_cold1 () servert_truc_cold_1 () servert_truct_truc_truc_truc_truc_cold_1 () DD1111121 servert_truct_truct_truc_tructt_truct	SUPERFATT: WHERE 6	MIRDE A. SCREATT FIDH + B. SCREATT FIDH)	
UPDTETTC:00D0BBT cottest_test_date UPDTETTC:01D0BBT cottest_test_date UPDTETTC:18818 sphotdm.s.cottest_b UPDTETTC:18818 sphotdm.s.cottest_b UPDTETTC:18818 s.sphotdm.test_code_eff + (Select	EDTEXTSC COLUMN	substdn coll code 1 11 sorrest test code 11 sorrest test score	
187 187 187 187 187 187 187 187 187 187	EDTESTIC: ORDERET	Fortrast data	
RDTERTECTERE RD	EDTRETEC BRDORT		
NOTESTIC: NUESS_1 & spheredm_pichs = h_servers_pichs was NOTESTIC: NUESS_2 & s.phredm_term_code_stf + (Select	EDTRATEC: LARLE	solution a, somean b	
NDTESTO: MNEDE_2 a.soburds_ters_code_off = (Select	EDTESTIC: WHELE 1	a schoold wide - h server nick and	
	IDTESTSC WIRDS 2	a schutch term code eff = (fielent	
	land		
	8		
--------------------------	--	--	---
A P Pr Table: BANDRO	Description: Dynamic SQL Definitions	Key: ACSTCODE WHERE_3	
AUDOIA CEDUA SEOUA AUDOI	4 AUD015 AUD018 AUD027 AUD029 AUD034 AUD	047 84N080 SCR000 SCR009 STUDIE STUD23 STUD24 STUD	205 STU207 STU216 STU223 STU224 STU260 STU262 STU
Key	Value1	Sauke V	Value3
ACADSTOCE: WHERE 20	AND D. STVASTD CODE (+) - SGDSTON ASTD	CODE AND A SHETTER FIDE - SUPETER FIDE	
ACADSTTTR: COLORN	SHRTTAN TRAN CODE (1 X. STVASTD DESC		
ACAD STTTE: ODD FIST			
ACADOTTTR- REPORT			Lin Li
ACADOTTTR: TABLE	SHETTER A. SCREEDN, STRASTD X. STRAST	D D	
ACADOTTE: WHERE 1	A. SHETTER TEER CODE -		
ACADSTITE-WHERE J	(ERLECT MAX (B. SHDITON TELR CODE)		
ACADOTTTR: WHERE D	FROM SHEET DESTRUCTION		- inixi
ACADOTTTE WREEK 4	WHERE A.S		ALLO ALL
ACADOTTTE: WHERE &	AND SCREET KEY, ACT	STCODE: WHERE 3	
ACADOTTTP: MREPE_G	(BELECT)		
ACADOTTER: WHERE 7	FROM SCRI		
ACADSTTTP: MHERE E	WHERE B. :		
ACADSTTTR: WHERE D	AND X. STI Veket from SOBSTON D		
ACADSTTTR: MIRRE 90	AND B. STI Volum		
ACAMPUS COLUMN	SCROTON (
ACAMPUS: REPORT	Value3		
ACAMPUS - TARLA	SCRSTDat Voluet		
ACSTCODE: COLUMN	repation a strate		
ACSTCODA TABLE	softer de		
ACSTCODE WHERE I	a and at de Custom		
ACSTCODE: MHERE 2	(Select Revision		
ACSTCODE: WHERE 2	from SGL		
ACSTCODE: WHERE 4	where b.t		
CAMPUS - COLUMN	with et day a		
CAMPUS-TABLE	athat do		
DCSTYPE: COLUMN	subst do 1		
DCSTVPS REPORT			
DOSTVOR - TABLE	subst.do		
ID-SCREATT-COLUMN	SUBRATT J		
KORCONATT - BREGET			
ID SCREATT / TABLE	SCREATT A		
CONGREATT- WIREF 1	ho DegreeVo		
CORPORATT: WHERE 2	11111 11 TimeStamp 04/09/09 10:10:	00	
DECESATT WHERE 3	and A. SC		
EDECOSATT INHERE 4	BAR (B. SCREATT TEDN CODE EEF)		
COSCREATT: WIERE 5	FROM SCREATT D		
DEGREATT: NHERE 6	WHERE A SCREATT FIDM . B SCREATT	FID: N	
EDTRETAC: COLUMN	ambatda coll code 1 () sortest tesc o	ode 11 sortest test score	
COTESTSC: ORDERBY	sortest test date		
IDTESTSC: PEPORT			
EDTRETEC TABLE	autation a. sortest b		
EDTRETEC: WHERE 1	a sobetion pids w b cortest pids and		
ID-TROTOC - WIERE 2	a schutch tera code eff a ffalare		
IDTRITIC: MARRE 3	payle solution term code aff) from an	brtdn c	
	and the second count of the second se	1	
- File			

D # # X 2 # D	8	n a dia		
+ + + Table: BANDOD	Description: Dynamic SQL Definitions	Koy: ACSTCODE WHERE_4		
81 AUD012 AUD013 AUD01	AUD015 AUD018 AUD027 AUD029 AUD034 AUD	047 8AN080 SCR000 SCR009 STUDIE STU023 ST	U024 STU035 STU307 STU316 STU323 STU324	STU260 STU262 STU5
(ny	Value1	Value2	Value3	
CADSTTTR: COLUEN	SHRTTPM_TERM_CODE N. STVASTD_DESC			
CADSTTTR: ORDERSY				
ACAD STTTE: BREGHT				
ACADSTTTR: TABLE	BHRTTPH &, SGRSTON, STVASTD X, STVAST	D B		
ACADOTTTR: WHERE_1	A.SHETTER_TERS_CODE =			
CADSTTTS: WHERE_2	(SELECT MAN (B. SHRTTPH_TERM_CODE)			
CADSTITE-WHERE_3	FROM SHOTTEN B			
ACADOTTTR: MILERE_4	VIETE A .: DANDOO	A REAL PROPERTY AND A REAL		
ACADSTTTE: WHERE_S	AND SOBS1			
ACADSTITE: WHERE E	(SELECT) KEY ACS	STCODE: WHERE 4	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
ACADOTTTP:WHERE_7	FROM SGR	4 4 F H X 🖬 🚳 🖇		
CAPSTTER: WHERE_0	W1888 D.1	A COLORED A		
ACADSTTTP: MHERE_9	AMD X. STL			
CADSTTTR: MHRRE_90	AND B.ST. Vouri there b.substdn_pida	r = #.igbit.ds_pids)		
CAMPUS: COLUMN	SGBSTESI_(Volue2			
CAMPUS: REPORT	Value3			
CAMPUS: TABLE	SCBSTEN			
CETCODE: COLUMN	agkation a Voluet			
CSTCODE: TABLE	substan : Sinhas			
CSTCODE: WHERE_1	a.sqbst.dr			
CSTCODE: WHERE 2	(Zelect Cuttom			
CSTCODE: MHERE_3	from SGI Revision			
CUSTCODE: WHERE 4	where b.t			
AMPUS . COLUMN	spirat-do_(
AMPUS TABLE	a gh an dn			
CSTYPE: COLUMN	sign at do_t			
CSTYPE: REPORT	and the second se			
CETYPE: TABLE	s glest de			
DISGRMATT: COLUMN	SGRSATT_J			
DUGRDATT: REPORT				
DISCONATT: TABLE	SCRSATT :			
DECORATT: WHEDE_1	SCREATT_ Who DegreeVo	12		
DEGREATT: WIERE_2	('III', 'I			
DECREATT: WHERE_D	and A.St TimeStamp 04/09/09 09:40:	47		
DSCRSATT: WHERE_4	nas (B. E.			
DECREATT: WHERE S	FROM SGREATT R			
DEGREATT: WIERE_6	WHERE A. SCROATT_PIDE = D. SCROATT_I	PIDH)		
DTESTSC: COLUMN	sgbstdn_coll_code_1 () sortest_tesc_co	ode 11 sortest_test_score		
DTESTSC: OPDERBY	sortest_test_date			
DITESTSC: REPORT				
DTESTSC: TABLE	sobstibn a, sortest b			
OTRETSC: WHEER_1	a.sgbstdn_pids = b.sortest_pids and			
DITENTSC: WHERE 2	a.sghardn_ters_code_eff = (Select			
OTESTSC: WIERE_0	maxic.sqbstdm_ters_code_eff) from sqt	betdn c		
DTESTSC: WHERE 4	where c.splitdn_pids = a.splitdn_pid	da)		

A summary of the records in BAN080 for this variable should look like the following:

ACSTCODE:COLUMN	Sgbstdn_stst_code	
ACSTCODE:TABLE	Sgbstdn a	
ACSTCODE:WHERE_1	a.sgbstdn_term_code_eff =	
ACSTCODE:WHERE_2	(Select Max(b.sgbstdn_term_code_eff)	
ACSTCODE:WHERE_3	From SGBSTDN b	
ACSTCODE:WHERE_4	Where b.sgbstdn_pidm = a.sgbstdn_pidm)	

2. Next a record needs to be added into the SCR002 table in which to scribe against. This record should have the same name as the variable created in step 1 above. The Data Element value should be the record from UCX-SYS999 that is to be used in the Scribe IF statement. Since you are pulling data from BAN080, this data will go into the rad_custom_dtl record. The rad_custom_code with a value of 'R322' in SYS999 should then be entered in as the Data Element. The UCX table can be left blank since this is coming from BAN080 data. The Edit Element1 should be the value of the data item. In this case, 'R323' in SYS999 points to the rad_custom_value field, this should entered into the Edit Element1 field. We will be retrieving all the values for this data item, so the Type value should be set to EV. Finally the value of the SCR002 record should be the name of the variable from the BAN080 table. In this case ACADSTST. The following is the screen shot of the SCR002 record:

Description Data Element UDC Table Edit Element Type IValue 1 Edit Element 2 Type IValue 2 Edit Element 2 Type IValue 3 Reterve AddetEct Element 2 Type IValue 3 Reterve AddetEct Element 2 Type IValue 1 Edit Element 2 Type IValue 3 Reterve AddetEct Element 2 Type IValue 3 Reterve FEY: Let To Colspan="2">Let To Colspan="2">Let To Colspan="2" FEY: Let To Colspan="2" FEY: Let To Colspan="2" FEY: Let To Colspan= Colspan= Colspan= Colspan= Colspan= Colspan= Colspan= Colspan= Colspan= Colspan="2" AddetEct Element 3 FEY: Let To Colspan Let To Colspan= Colspan C	Description Data Externed (UEC Toble Edd Externer) Type 1 Value 1 Edd Externer 1 Type 1 Value 2 Edd Externer 1 Type 1 Value 3 Restrict 1 Type 2 Value 2 Edd Externer 1 Type 1 Value 3 Restrict 1 Type 1	rbe AUD0131 AUD0331 SI	CR001 SCR002 SCR003 SCR004 SCR0	05] SCR007 SCR044 S	cno45 STUDIG	STU035	STU307 STU31	6 STU323 STU324	STU350 STU	352) STU560] STU563]	
ACCTCODE Accelerate Deandsing P 203 P 202 P ACCTCODE ACCT	CETCODES CETCODES Recodense: CHanneling Bacid Bacid Bacid Bacid Bacid Bacid Bacid PARM Diversal I DFA BCC BACID PARM Diversal I DFA BCC BACID FIF: LETCODES IN CRADUE FIF: LETCODES IN CRAD	Key	Description	Data Element L	ICX Table Edit Eleme	est 1 Type	1 Value 1	Edit Element 2 Type	2 Value 2	Edit Element 3 Type 3 Value 3	Reserved
ATTRIUNT# Bitadenic Attributes B223 B222 BV GRAUV Oversall GPA B223 B222 BV GRAUV OVERDIGE Description FACTORS Image: Control of the state of	TEXINITY Benaver Devidence. Are risk under a 1920 1922 197 ATRINITY 1920 1920 1920 1920 1920 1920 1920 1920	ACSTCODE	Academic Standing	Raza	\$355	87	ACSTCODE	1.10			
DPANY DPARYA B323 B322 BY GPANY OVERSIDA OVERSIDA B322 B322 BY GPANY OVERSIDA OVERSIDA B322 B322 BY OVERSIDA OVERSIDA SECKOOZ Image: SecKooz Image: SecKooz Image: SecKooz 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. Image: SecKooz Description Academatic (Imandiang) Description Academatic (Imandiang) Data Demert JUCX Table State of a detail this element is used as a filter Type Type Type Th Data is on a detail this element is used as a filter Type If Data is on a detail this element is used as a filter Type Type If Data Into a detail this element is used as a filter Type If Data Into a detail this element is used as a filter Type If Data Into a detail this element is used as a filter Type If Data Into a detail this element is used as a filter	DADY Deverall GPA B123 B122 BV GRADY VERGYA Overall GPA B123 B122 BV OVERGYA VERGYA Overall GPA B123 B122 BV OVERGYA VERGYA Overall GPA B123 B122 BV OVERGYA VERGYA Vergetall GPA B123 B122 BV OVERGYA VERGYA Vergetall GPA B123 B122 BV OVERGYA VERGYA Vergetall GPA FEW. CESTCODE Vergetall GPA Immorphics Vergetall GPA VERGETA Vergetall GPA OVERGETA Sciencer Vergetall GPA Vergetall GPA VERGETA Vergetall GPA OVERGETA Vergetall GPA Vergetall GPA Vergetall GPA Vergetall GPA Vergetall GPA Vergetall GPA Vergetall GPA Vergetall GPA Vergetall GPA Vergetall GPA Vergetall GPA Vergetall GPA Vergetall GPA Vergetall GPA Determent Determent Determent Sciencert Sciencert Sciencert Sciencert Vergetall GPA Vergetall GPA Vergetall GPA	ATTRIBUTE	Banner Student Attribut	ses Raza	\$322	80	ATTRIBUTE				
WHERE OX PLD	Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Intervent Interv	GPA0V	Dverall GPA	R323	8322	80	GPA09				_
	Value Use ACTV If Type is AT - "Active Term" Edit Element 3 If Data is on a detail this element is used as a filter 💌		REMIND When us Description A Data Dence (P UCX Table	In Internet Sector Provide Sector Pr	CX-SCR092 char orGPA you do no	nges be t need F staten	sure to restarr an entry here. ent	the web jobs.	*		
			Edit Diennert 1 1 Types T Value A Edit Bionent 2 Volum Bdit Bionent 3	V AT /EV / CSTCODE Ube ACTV If Data is on a detail AT /EV / Use ACTV Use ACTV If Data is on a detail If Data is on a detail	f if Type is AT - "Acti I this element is used f if Type is AT - "Acti I this element is used	vê Têrm" as a filter ve Term" as a filter			 1		
			Edit Diement 1 19 Type: [Velue J. Edit Biement 2] Velue [Velue] Edit Biement 3]	V AT /EV / CSTCODE Use ACTV H Data Is on a detail AT /EV / Use ACTV H Data Is on a detail	r if Type is AT - "Actr I this element is used r if Type is AT - "Actr I this element is used	vê Têrm" as a filter ve Term" as a filter			1		

3. After the tables have been set up in SureCode for the BAN080 records and the SCR002 record. A webrestart and a UCX12job command should be issued. For students to get this data put into their rad_custom_data table, they will need to be re-bridged from Banner into DegreeWorks. If the data item does not exist in Banner for the student, the record will not be loaded in the rad_cust_dtl table. Once a student has been re-extracted, his student data record will now look like this:

DegreeWorks by SunGard I	Higher Educal	tion SunGaro	d DegreeW	orks DWSEED Te	est 2 - I	Mozilla Firefox												
http://149.24.215.2:8802/1	IRISLink.cgi																	☆ 🇳
	KS ION							S	UN	GAF	RD L		IVE	RS] ducation D		on.		
Portal		FAQ			Help		F	Print			Excep	tion Ma	nagemer	nt			Log Out	t
Find Student ID	Name	F FE	Degree	Major		Level Stur	lent C	lass Level	Last Ai	ıdit I.	ast Refre	sh		0				
	ress, Nancy	•	BSN	Nursing		UG Free	shman		Today		Today at 9	9:31 a.m		0				
Worksheets Planne	er Note	s Petit	ions	Exceptions	GP	A Calc Admin												
worksheets > Form	nat: ident Data Re	eport 💌	View	Save a	s PDF	Process New		<u>Class His</u>	tory									
ICHSM	11000 199610	210009604	CHSM	11000	Ancie	nt,Medieval, Ren Thought	N	IN	Y	IN	N	Y	N N		VA	0.000	003.000	1003.000
History NURS	1100 199610	210009604	NURS	1100	Intro.	to Prof. Nursing	N	N	Y	N	N	Y	N		NA.	0.000	001.000	001.000
RELS	1050 199610	210009604	RELS	1050	Christ	ianity:Traditions/Trans	N	N	Y	N	N	Y	N		AV.	0.000	003.000	003.000
What If Trai	nsfer-Dtl																	
Cour	se Term	ld	Discipline	CourseNumbe	r Cour	seTitle	TrEts	TrName	Т	rCrseKey	TrCourse	•	TrCredit	s TrStart	TrStop	Calendar	TrGrade Aud	lit Insufficient
Look Ahead MATH	1230 199610	210009604	MATH	1230	Introd	uctory Statistics	5297	Howard Un	iversity S	TA1500	Statistics I						N	N
POLS	1100 199610	210009604	POLS	1100	Intro.1	to American Government	5297	Howard Un	iversity P	OL1000	1						N	N
PSYC	1000 199610	210009604	PSYC	1000	Intro.1	to Psychology	5297	Howard Un	iversity P	SY1000	Intro to Ps	ychology					N	N
SOCI	1000 [199610	1 210009604	Isou	μοοο	lintro.1	to Sociology	5297	Howard Un	iversity S	001000	jintro to So	ciology					N	N N
Cus	stom-Dtl																<u></u>	
Term	n Id		Custor	nCode	C	ustomValue	Cu	stomTitle		School	De	greeCod	e	Cre	ateDate		CreateWho	Ê e
1	210009	9604	A01		00	125	19	950115						200	90723		RADBRIDGE	
	210009	9604	A02		00	124	19	950115						200	90723		RADBRIDGE	
	210009	3604	A03			125	19	950115		-				200	90723		RADERIDGE	
	210008	1004	A04		00	125	19	950115						200	90723		RADERIDGE	2. 3
	210003	9604	ACSTO	DDE	00	3	13.	500115		-				200	90723		RADBRIDGE	
	210009	1604	CAMPLI	5	M									200	90723		RADBRIDGE	
	210009	9604	OVERG	PA	-					1				200	90723		RADBRIDGE	2
Rep	ort-Dtl																	
Term	Id		Report	ode		ReportValue		Report	Sea	Sch	ool	DegreeCo	ode	0	reateDate		CreateWhy	
	210009	604	ACAMPLI	s	-	M		0001				begi eee		20	090723	·	RADBRIDGE	-
	210009	604	DCSTYPE			199610AS		0001						20	090723		RADBRIDGE	
	210009	604	KDTESTS	ic		NUA0125		0001						20	090723		RADBRIDGE	
	210009	604	KDTESTS)C		NUA0224		0002						20	090723		RADBRIDGE	
	210009	604	KDTESTS)C		NUA0526		0003						20	090723		RADBRIDGE	
	210009	604	KDTESTS	ic .	_	NUA0425	_	0004						20	090723		RADBRIDGE	
	210009	604	KDTESTS	ic .		NUA0325		0005						20	090723		RADBRIDGE	
	210009	604	STSTCOL)E		AS		0001						20	090723		RADBRIDGE	
	210009	604	STYPCO	JE NOE		iranster		0001						20	090723		RADERIDGE	
10 M	210009	604	TESTSCO	ANE .	_	NUA0125	_	0001						20	090723		RADBRIDGE	
	210009	1604	TESTSCO	RE	-	NUA0526		0002		7/25				20	090723		RADBRIDGE	
	210009	604	TESTSCO	RE	_	NUA0425		0004				-		20	090723	_	RADBRIDGE	
	210009	604	TESTSCO	RE		NUA0325	_	0005						20	090723		RADBRIDGE	
						t.		1									l	
				W G	Sterios													
egreeworks is a product of	SunGard Hi	igher Educ	ation All c	ontents are cop	oyright	© 1995 - 2008												
one																		4
Start Office Comm	😛 2 Yahoo! .	+ 💽 Int	oox - Micr	🕘 4 Firefox	• 6	🚹 To create a 🛛 🎆 S	SureCoo	ie 🛃	149.24.21	.5 👮	DegreeWo	ork [3	2008	209	" 🕵 🛛 🌔	0	🥡 🇞 8:39 Al

4. Now you can put in rules into your blocks to use this data variable created above. For our example using the Academic Status code, we can scribe against this value to determine if a particular rule has been met. An example using the ACADSTST code is:



This rule will then look like the following once an audit is run:

and diversity of a state of the	sinico/sessink.op									9
Degree	Vorks					SUNGA		VERSIT	Y	
Portal		FAQ	100	0	Print	2 D	Exception Mar	segment.		Log Out
Student ID	H 4 Name Childress, Nancy	H Degree BSN	Magor Narsing	Level 92 UG P	tudent Class I Veshman	aveil Last Audit L Today	ast Befreih Today at 9:31 a.m.	0		
arksheets [19	lanner Noter	Petitions	Exceptions 0	PA Calc Admir	0					
eksheets >	Student View	• Vary	Save as PC	Process how	2 Che	is History				
loty	Advisor 1					Degree	85 in Nursing	- ⁻		
and in	Advisor 2					Major	Nursing			
	Overall GPA	0.000				Classification	Freshman			
nk Athe-ard										
		Dues deigenerate		100	Degree	Progress				
		Par don munates		Contraction of the local division of the loc						and the second se
		Credits	9%		_					
	Degree of United condition	Credits 12A	equivments: 109	Credits readed				Acadmin Year; (* GPA: 0.0	ана I 00	nsits Required: 12 Orelits Applied: 11
	 Degree of Unimet condition 	Credits	equirments: 109	Credits needed	HEM 1103	General Chensch	vīlub	ALASHIN YAATI (* GPA: 0.0 NA	аа . С	rolits Required. 12 Overlits Applied. 13 Fail 1995
	Degree of Unmet condition	Credits may ans for this set of r	equirments: 109	Credit i needed	04EM 1103 04EM 1131 04EM 1000	General Cherrant General Cherrant Society Heches	v TLah v T	Academia Year: 10 GDA: 0.0 AL AL AL AL AL	a 2776 1 00 00	redits Required. 1.5 Oreflits Applied: 13 Tail 1995 Fail 1995 Ed 1995
	Degree of Unmet condition	Credits IFA res for this set of r	equinments: -109	Credits recoded C	DEM 1103 DEM 1133 DEM 1133 DEM 1000 WES 1100	General Cherrett General Cherrett Ancient Medera Jinto, 12 Port 14	v T Lah y T I, Nen Thought	Academia Vient (2) GPA: 0.0 Alla Alla Alla Alla Alla		resks Required: 12 Orefilts Applied: 13 Tal 1995 Fai 1995 Fai 1995
	Degree of United condition Test rule for	Credits ICA Pris for this set of a most effection	equirments: 109	Criedits recoded	CHEM 1103 CHEM 1131 CHEM 1100 HIRS 1100 HIRS 1100 HIRS 1050	General Cherreth General Oriente Ancient Medieva Inton in Prof. N Chentursty: Trad	v Tulk v 1 Jen Thought Jong/Turks	Antocharan Yanar (20 GaPA: 0.0 Alla Alla Alla Alla Alla Alla	si 200 0 00 00 00 00 00 00 00 00 00	Tal 1995 Fal 1995 Fal 1995 Fal 1995 Fal 1995 Fal 1995 Fal 1995
	Despres of Unmet condition test rule law	Credits IEA res for this set of a asstallation	equirments: 109	Credits reading	DHEM 1103 DHEM 1133 DHEM 1133 DHEM 1000 WIRS 1100 RELS 1050 Stil Needect 3	General Cherrath General Cherrath Ancient,Madewa Inho- to Prof. Ta Chethinthy,Tradi 5 Gasses In ⊕ Φ	v TLB v T , hen Thought John/Thims	Альновичин, Чинит; 0.00 68/14; 0.03 485 485 485 485 485 485	00 00 00 00 00 00 00 00 00 00 00 00 00	Tal 1995 Fal 1995 Fal 1995 Fal 1995 Fal 1995 Fal 1995
	Congress of Unimet condition	Credits crA one for this set of a and allotton Als	equirments: 109	Credits needed C C C C C C C C C C C C C C C C C	CHEM 1103 CHEM 1133 CHEM 1133 CHEM 1000 WRES 1100 RELS 1050 Still Needect	General Chartett General Chartett Anders J Medera Inter to Port, T Chartett, Trial 5 Gazes th @ @	Y TLAB Y 1 I Gene Thought Joing Thirts	Academia Your: 10 GPA: 0.0 Na Na Na Na Na Na	5 2 776 0 00 00 00 00 00 00 00 00 00	Tal 1995 Ad 1995 Ad 1995 Ad 1995 Ad 1995 Ad 1995 Ad 1995
(Degree of United condition Descharge degree Astronomes Mager Region	Credits CCA one for this set of a and allotton Als read	equirments: 109	Credits needed C C C C C C C C C C C C C C C C C	CHEM 1103 CHEM 1131 CHEM 1131 CHEM 1000 WIRS 1100 RELS 1050 Still Needect 3 Still Needect 3	General Charrent General Charrent Anders Medena Inton. to Prof. In Creativety, Trail 5 Gasses In @ @ "ALOR block was not	v TLati v 1 Linus Thought Linus const Turus focund but is rega	Academic Year: (2) GRA: 0.0 Ha Na Na Na Na Na Na Na Na Na		nekts Horganida – 12 Conditis Agodinici – 11 Tali 1995 Fali 1995 Fali 1995 Fali 1995
	Degree of Unnet condition Unnet condition Inst rule for Asstcope is Major Regul	Credits (r.o.) one for this set of a anti-flation Alls read	equérements: 109	Creates meading C C C C C C C C C C C C C C C C C C C	0464 1100 0464 1131 0464 1131 0465 1000 4815 1000 4815 1050 518 Needoct 3 518 Needoct 3	General Charrent General Charrent Andren Medera Jinto, to Pool In Onstituetty-Tool Chases in @ @ "ACKIR Mock was not	Y TLID Y I Liner Thought Ions/Thims found but is rega	Academic Your: (0) GRA: 0.0 Ala Na Na Na Na Na Na Na Na Na Na Na Na Na		matts Hospanida (20 Conditis Agolesid) (11 Tal 1005 Tal 1005 Tal 1005 Tal 1005 Tal 1005 Tal 1005
	Compare of United condition United condition Institute for the form Astronomy Institute Condition Pharmal 228	Credits rick one for this set of a and dilation Ads rend breakatory Stansico	equérments: 109	Credits neodina C C C C C C C C C C C C C C C C C C C	0464 1100 0464 1131 0464 1131 0465 1000 4815 1000 4815 1050 518 Needoct \$	General Chernet General Chernet Anount Motesus Inton. to Prof. In Cristiantly-Tool 5: Gazoes In ⊕ Ø MAJOR Mock was not	v TLatt v T insv Thought insign form/Thens format but is resp s	Accelorum Your: 10 GAC: 0.0 10 10 10 10 10 10 10 10 10 10 10 10 10		Chanans Applied
	Degree of Unnet condition Unnet condition test are for Astronot as Page Reput Denote recent Saturded by	Credits dravit orn for this set of a and all atom ADS introductory Sounders Statistics - Howard West		Credits meaded C R N N S S	046M 1100 046M 1100 046K 1100 046K 1100 046K 1100 046K 1100 050 Needed 1 500 Needed 1	General Cherneth General Cherneth Anorent Medine Inter to Nord: To Chernetinety Thata S Gazese n ⊕ ⊕	y TLuit y I Lines Thought ainto cons/Trims found but is reg	As address: Your: (o) (202: 0.0 103 103 103 103 103 103 103 103 103 10		Classes Applied:
	Degree of Unset coulds Unset coulds test rate for Astronot s Manual Could s Saturbook Saturbook Saturbook POSS 109-	Credits creat crea	equivaments: 109	Credits recoded	3464 1107 3464 1103 3464 1000 4865 1100 4865 1050 588 Needoct 3 588 Needoct 3	General Oversett General Oversett Anderst Norde to Understand for the Con- Constant of the Con- Constant of the Con- Status of	v Tulit I han Thought array array found but is rega	Assolutions Your; 100 GAA: 0.0 As. As. As. As. As. As. As. As. As. As.		Trait 1995 Field 1995 Fiel 1995 Fiel 1995 Fiel 1995 Fiel 1995 Classes Appelled:
	Description of Univer condition test note that Astronomer Require Manar Require Manar Require Manar Require Manar Require Saturded by: Saturded by: Saturded by:	Credits con con con con con con con co	ergulermonentu: 109 ergulermonentu: 109 erendy anmeert anmeert anmeert	Creater needed C C C C C C C C C C C C C C C C C	DIEM 1103 DIEM 1103 DIEM 1100 WIES 1000 WIES 1050 ELS 1050 Still Needoct 1	General Cherreth General Cherreth Anorent Medewa Inter, to Not, fr Orstanstry, Trial 5 Genes In ⊕ Φ MAXIE Mock was not 1 1 1 1 1	y Tule y I Les Thought ang boor/Thes found but is requ t	Accelorus Your: 10 GAA: 00 MA 26 NA 26 NA 26 Constitute 2 Accelorus 2 Accelorus 2 Accelorus 2 Accelorus 3 Accelorus 3 Accelorus 3 Accelorus 4 Acceloru		noks Angelered 13 Oreflit Agolied 13 Tell 1995 Fel 1995 Fel 1995 Fel 1995

Nant Sofer Ca. State M. State M. State M. State M. State Sta

Scribing against Test scores:

Test scores are brought into the DGW rad_custom_dtl table from the SORLCUR table by default based on your bannerextract.config file. By default all test scores are retrieved. You can modify the bannerextract.config file to delimit what test scores are brought in by adding WHERE statements into the bannerextract.config file. Since these scores are already being brought in, it is not needed to create a BAN080 set of records to retrieve them. You will need to create SCR002 records for the test scores to be able to scribe against the tests. As an example, if a student has the following records in SORTEST:

SORTEST_TESC_CODE	SORTEST_TEST_SCOR	RE SORTEST_TEST_DATE	
A01	25	15-JAN-95	
A02	24	15-JAN-95	
A03	25	15-JAN-95	
A04	25	15-JAN-95	
A05	26	15-JAN-95	

To be able to scribe against one of these, the test code will need to be added to the SCR002 table. For example to scribe against test types of A01, the following SCR002 record needs to be created:

REMINDER: Biological Sciences SPA SPA SPA SPA SFA SFA SFA SFA SFA SFA SFA SF	Data Element P227 P229	UCX Table Edit Ener 9222 9222 9222 9222 9222 9222 9222 92	nt 1 Tipe NV NV NV NV NV NV NV	Vake1 A01 ACUTCODS ATTRIBUTS GPA07 OVEROPA OVEROPA	Edit Element 2 1 P323	ype2¥ake2 ,401 □ ×	Edit Element	Type 3 Value 3	Flessred 1 F
ADL 2 Sending Inviden Afersburke GRA GRA GRA CRA CRA CRA CRA CRA CRA CRA CRA CRA C	P222 P222 P222 P223 P22 P2 P	8022 8022 8022 8022 8022 8022 8022 8022	EV EV EV EV	A01 ACSTCODE ATTRIBUTE GPA07 OVEROPA OVEROPA	9323	401 			
E Pendiano Invisione Arershotras GPA OPA STRAY SECOOD2 REMINDER: T When using Description Trans UCK Table [225] UCK Table [225]	ECOPER - A01 Specify the datum	8022 B022 B022 B022 B022 B022 B022 CX SCR007 chan irGPA you do no	EV EV EV EV EV	ACTIVITY GRADY OVERSEA OVERSEA sure to restart an entry here.	9325 the web jobe.	401 [] ×			
Ruden Afersburge GA GA GA CA CA CA CA CA CA CA CA CA CA CA CA CA	Roza Roza	8222 8322 8322 8322 1 • • • • × •	RV RV RV RV RV	ATTRADUTE GPA07 OVERSPA	9323	201 201 2012			
GPA GPA STRV SCR002 REMINDER: T When using Description Term. Data Denett [1225 UCK Table	Roze Bozo Bozo Rozy In Int Int SSGPA or Banne Roores - A01 Specify the dature	BOZE BOZE BOZE BOZE CX SCR002 char rrGPA you do no	tv tv	GPA09 OVERGPA	9323	201 201 201 201 201 201 201 201 201 201			
693. SERV SERV SECO02 REMINDER: 1 When using Descriptor [from Data Denet [223] UCK Table	8023 8023 KEY A01 Id d To enable your U SSGPA or Banne Recret - A01 Specify the dature	B322 B322	tv	aure to restart	p323	401 			
REMINDER: 1 When using Descriptor: Text. Data Bandet: P225 UCK Table	KEY A01 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	I I I X II ICX SCR002 char ICX SCR002 char	1 🔿 🎗 ngos bo s t need a	sure to restart	the web jobs.	×01 ×101×		d: 1.	1
REMINDER: T When using Description Team Date Element 1923 UCK Table	KEY 401 14 4 To enable your U SSGPA or Banne Beores - A01 Specify the datum	I F FI X R ICX-SCR002 char erGPA you do no	1 🥵 🎙 nges be s it need a	sure to restart	the web jobs.	× 01.			
REMINDER: T When using Description Team Date Element 1925 UCX Table	KEY 401 14 To enable your U SSGPA or Banne Reores + A01 Specify the datum	ICX SCR002 char erGPA you do no	1 🥌 🎙 1ges be 1 it need a	sure to restart in entry here.	the web jobs.	ŕ			
REMINDER: 1 When using Description Term Data Demot P323 UCX Table	To enable your U SSGPA or Banne Reores + A01 Specify the datum	CX SCR002 char erGPA you do no	1 🚭 🎗 iges be s it need a	sure to restart in entry here.	the web jobs.				
REMINDER: 1 When using Description Text Data Denert [2323 UCX Table	To enable your U SSGPA or Banne Reores - A01 Specify the datum	ICX-SCR002 char rrGPA you do no	iges be i I need a	sure to restart in entry here.	the web jobs.				
REMINDER: 1 When using Description Team Date Denert [2325 UCX Table	To enable your U SSGPA or Banno Reores - A01 Specify the datum	ICX SCR002 chai rrGPA you do no	iges be s It need a	sure to restart in entry here.	the web jobs.	-			
When using Description Team Data Element 19323 UCX Table	SSGPA or Banne Reares + A01 Specify the datum	erGPA you do no	d need a	in entry here.					
Description Term Data Dement 1923 UCX Table	Scores - A01		4 неео а	in endy nere.					
Description Team Data Element (19323 UCX Table (Scores - A01 Specify the datum								
Description Team Data Deniert 19325 UCX Table	Specify the datum								
Data Element 9323 UCX Table	Specify the datum	a station of the state of the state of the							
UCX Table	and a second second second	to be used in a Schor	Fittere	int .					
UCX Table		10 00 0000 01 0 00000							
Foll Demont 1 11177	W Details on a deta	this element is used	and in Street						
CON COMMENT PARTY	- Data is on a deta	NURS CIERIPIE IS USED	02 0 MOT						
Type EV A	IT /EV /								
Value A01	Use ACT	/ if Type is AT - "Add	ve Term*				1		
Edit Element 2	If Data is on a deta	I this element is used	as a titer						
Turne	TIEVI								
1996 B	17647								
Volue	Lise ACT	/ If Type is AT - "Acti	ve Term"						
Edit Demert 3	M Details on a deta	this element is used	as a filter				1		
1 con pensions 1	a bacana chi a beca	I has even bere is upon	92.0 INSt			1	1		
	Edit Element 2 Type A Volue Edit Dement 3	Edit Element 2 If Distance on a deter Type AT / EV / Value Use ACTV Edit Element 3 If Detenic on a deter	Edit Element 2 If Data is on a detail this element is used TypeAT (EV / Value: Edit Denent 3 If Data is on a detail this element is used Edit Denent 3 If Data is on a detail this element is used	Edit Element 2 If Data is on a detail this element is used as a titler Type AT / EV / Value Use ACTV if Type is AT - "Active Term" Edit Dement 3 If Data is on a detail this element is used as a filter	Edit Element 2 forta is on a detail this element is used as a filter Type AT / EV / Use ACTV if Type is AT - "Active Term" Use ACTV if Type is AT - "Active Term" Edit Element 3 forta is on a detail this element is used as a filter	Exit Element 2 Forta is on a detail this element is used as a titler Type AT / EV / Velue Use ACTV if Type is AT - "Active Term" Exit Element 2 Forta is on a detail this element is used as a filter	Bit Element 2 If Data is on a detail this element is used as a filter Type AT / EV / Value Use ACTV if Type is AT - "Active Term" Dit Element 3 If Data is on a detail this element is used as a filter	Edit Element 2 II Data la on a detail this element is used as a titler Type AT / EV / Velue Use ACTV II Type is AT - "Active Term" Coll Dennet 3 II Data is on a detail lible element is used as a filter	Exit Element 2 IF Data is on a detail this element is used as a filter Type AT / EV / Veder Use ACTV if Type is AT - "Active Term" Exit Element 3 If Deta is on a detail this element is used as a filter

Then in Scribe the following rule can be put into place:

File Edt Search	Parse Options Host Window Help	_ia
) 🛸 🖬 🏟 😭		
tules Block Blockype Classes inde Confain nie Confain nie Confain nie Confain nie Croup Infelie Nested Stoup nie Nested Stoup nie NorCourse Subset	<pre>X #Frank DBM #Filing 1n filing 1n</pre>	
	Lamel "test all for instaliation"; If (ACTOODE is AD" Rule-Complete Label "ACTOODE is AD" Rule-Independent Pule-Independent File - Solid - Solid Ad" Laben Pule-complete Label - Solid - Solid - Solid Ad"	
	else: Fule-Incomplete Label *Test A01 score is 24 or lower"; 1 Storm: Label 2 "Major Reference"; SUD,	
	Infinite The Beerspice Bee	
	1	2

Running the audit with our test student gives the following result:

Portal If 4 Name Chickess, Name Worksheets Planner Parksheets Planner Portal Planner Portal Planner P	AQ AQ A	Level Studient Class (UG Presiman alt. Admin Exosena.tenel Class Despres	Last Audit Last Refer Today at S st History Classification free or Progress	tion Management sh :21 a m ①	VP-ME: 1992-1996	Log Out
Maria Carlos and Carlo	m + H Degree [Major cr = DPA = [Maring training Exceptions EPA = [Maring w > Verw Seen as POF A 0.000 Enceptions Encertaints Encodes 04w Seen as POF Encodes Image: encodes 04w Seen as POF Encodes difference 04w Seen as POF Encodes	Lond proteint Clear US Frankman alt. Admin Process rend Clear Degree 0.3%	Level Last Audit. Last Stefn Today at 9 st History Classification Pre- or Progress	ahr (Carlonne) Academid	Vene: 1995-1996.	Drabit Respond
Vorksheets Planne No forskheets Sharner No forskheets Sharner No forskheets Sharner No forskiewe Sharner Sharner Sharner Sharner Sha	Anny Petitions Exceptions 60A 0	Admini Process tend Open Open Open	sa Jihitony mana rea Classification free in Programs	n ng Ihman Academic	Year: 1975-1996	Drebits Respared 1
International Statement of Constraint Statement Constraint Statement Constraint Statement Constraint Statement Constraint Statement Constraint	View: Saver as PDF Occo Requirements Crotits Occ Ore ditions for this set of requirements: 100 Occ	Fracess tend Clar Degree Of so	s History Tease Classification free of Progress	ning Amari Academic	Year: 1920-1996	Credits Respired: 1
Instanty Decreal CAV has a strain of the company take Abread	0.000 Requérements Crocites Seu ord tos ditions for this set of requérements: 100 Outles	Dogra 0.2%	Classification free	Acadomic	Year: 1995-1996	Credits Required: 1
hus 17 Ish' Ahriad Cheese cond Cheese cond	Requirements Cradits deu et land datases for this set of requirements: 100 Guida	Dogra 6276	e Progress	Academic	Year: 1995-1999	Erreißta Respared: 13
int Abriad int Abriad int Abriad int Coul int Coul	Requirements Croality Only Only Croality Only Croality Only Croality Croality Only Croality	Dogra 0.3%	e Prograss	Academic	Vour: 1995-1999)	Credits Repared: 1.
Inter Advised	Ecolits 9% Ecolits 9% Editions for this set of requirements: 200 Cusits	0.376		Academic	Year: 1995-1996	Credits Required: 12
Disprine Description text outer description description description description description	Erndits des	r resected		Academic	Vour: 1925-1995	Credits Required: 17
Dupping Dupping text code Construct Cond Construct	nd Mines difficute for this set of requirements: 100 Oudri	e rusaction		Academic	Veur: 1995-1996	Credits Respaced: 12
Corporation Construction Const	r of OSN ditions for this set of negatiments: 100 Code	r peoded		Academic	Year: 1995-1995	Credits Required) 1.
timmet cond text rule & AUTCODE S fext A01	ditions for this set of requirements: 100 Ordin	r needed				
C test nde		a a teatratic estis a			GPM1 0.000	Creckts Applied; 11
C) test role		CHEM 1103	General Chemistry ELab		F46 (3)	Fall 1995
C test rule		CHEM 1131	General Charmitry 1	12210	NJA (33)	Fall 2095
	for installation	MRS 1100	Intro. to Prof. Nariang	ASP E	NA (3)	Fall 1995
		RELS 1050	Christianity: Traditions/Trans		NAA (CI)	Full 1995
Test ADT		Stil Needed.	5 Classes in 🖶 👁			
🖾 Test A01	E H AS					
	Excore is greater than 24					
Ci Hajor Re	equired	Still Needled:	MAJOR block was not found bu	it is required		
ALC: NOT THE OWNER OF THE OWNER						
trautflictory	Marcheller Territor			Ene	dits Applied: 12	Classes Applied
Fairful 1230	and an and the second second		1995	1	1.00	
POLS 1100	Intro. to American Government		TO	2	Full 1995	
histinfied by						
PSYC 1000	rys POLIDOD - Howard University				Fail 1995	

Running an audit with a student who does not meet this qualification or has not taken this test will get the following result:

SUNGARD HIGHER EDUCATION			SUNGARD UN	IVERSIT	Y	
Portal	FAQ	Help Prin	it Exception M	lanagement		Log Dut
Marks, Pa	Name + H Degree Major	Level Student Clas	s Level Last Audit Last Refresh Today Today at 9:27 a.	m. ()		
rksheets Planner	Notes Petitions Exception	S GPA Calc Admin				
exmeets Back	Selected Wont-it Itemst	LOOK Abend Courses Used:	110.0			
ory Deera	GPA 0.000		Classification Fredman			
tif >		Degr	ree Progress			
k Ahead	Requirements	38%6				
	Credits 139					
	must of DCM			Academic Year: 1	995-1996	Credits Required:
(End) (A)	No. Avenue.			6PA: 0	201	Credits Applied:
Unmet	conditions for this set of requirements	 105 Godts needed 				
		CH5M 1000	Ancient, Medieval, Ren Thought.	NA	(3)	Fal 1995
		ECON 1101 ENGL 1005	Literature & Composition I	NA	(3)	Fall 1995
🗆 test	rule for installation	MGMT 1006	Microcomputers with Applicatos	NA	(3)	Fall 1995
		RELS 1050	Christianity: Traditions/Trans	NA	(3)	Fal 1995
		Still Noveked:	5 Classes in @ @			
IZA AST	DDE is AS					
D Test	AU1 score is 24 or lower					
	r Required	Stil Needed:	MAJOR block was not found but is re	quired		
O Mat						
- Mat				Credits App	ied: 15	Classes Appl
⊡ Maj	gress					
In-pro UISM 10	gress 10 Ancient, Medieval, Ren Thought		144	3 Fall 1993	<u>y</u>	
In-pro Clism 10 ECON 11	gress 20 Ancent, Medeval, Ren Thought 1 Principles of Microeconomics		144. 144	3 / 68 1995 3 Fail 1995		
In-pro Crism 10 ECON 11 ENGL 10	gress 20 Ancent, Medeval, Ren Thought 1 Principles of Microeconomics 5 Literature 6, Composition 1		144. 164. 144.	3 Fal 199 3 Fal 199 2 Fal 199		
In-pro crism 10 ECON 11 LNGL 100 MGMT 1	grobs Ancent, Medievel, Ken Thought 1 Principles of Microeconomics 5 Literature (), composition 1 06 Microecomputers with Applications		760. 160. 160.	3 Fall 1997 3 Fall 1997 3 Fall 1997 3 Fall 1997		

Using Banner data as Transit selection

There may be a need to use a data item from Banner that is not normally pulled over into DegreeWorks as a selection criteria in running reports in Transit. This can be done by creating and pulling over the variable into DegreeWorks, then updating the selection criteria used in Transit. Here is the procedure: (Our example will pull a graduation code from Banner.)

1. Create the variable to be pulled over from Banner in the BAN080 table. (For information on how to create BAN080 variables, please refer to the documentation on retrieving BAN080 variables.)

For our example we will create the variable GRADCODE from field SHRDGMR_GRST_CODE of SHRDGMR.

2. Make sure the variable's REPORT value is created as a blank record so that the data gets loaded into the rad report dtl table without a validated value.

		2 SCRUTZ [ASSUTZ [ASSUTZ [ASSUTZ [ASSUTZ [ASSUTZ [ASSUTZ [ASSUTZ [ASSUTZ]
	Value1 Value	2 Value3
CSTCODE: COLUMN	sqbstdn stst code	
CSTCODE: TABLE	sgbstdn : 🐼 BAN080	
CSTCODE: WHERE_1	a.sgbstdr	
CSTCODE: WHERE 2	(Select KEY: GRADCODE: REPORT	
CSTCODE: WHERE_3	from SGE 😽 🖌 🛏 🗙	n 🖨 💡
CSTCODE: WHERE_4	where b.s	
AMPUS: COLUMN	sgbstdn_c	
AMPUS: TABLE	sgbstdn	
CHOLD: COLUMN	sprhold_F Value2	
CHOLD : REPORT	Value3	
CHOLD : TABLE	sprhold	
CHOLD: WHERE_1	sprhold_t	
CSTYPE: COLUMN	sgbstdn_t Status	
CSTYPE: REPORT	Custon	
CSTYPE: TABLE	sgbstdn	
RADCODE: REPORT	Revision	
DSGRSATT:COLUMN	SGRSATT_J	
DSGRSATT: REPORT		
DSGRSATT: TABLE	SGRSATT #	
SGRSATT: WHERE_1	SGRSATT_	
DSGRSATT: WHERE_2	('IH', 'L	
SGRSATT: WHERE_3	and A.S.	
SGRSATT: WHERE 4	max (B. S	
SGRSATT: WHERE_5	FROM	
SGRSATT: WHERE_6	WHERE	
DTESTSC: COLUMN	sgbstdn_c Who	
DTESTSC: ORDERBY	sortest_t TimeStamp	
DTESTSC: REPORT		
DTESTSC: TABLE	sgbstdn a, sorcest p	
DTESTSC: WHERE_1	a.sgbstdn_pidm = b.sortest_pidm and	
DTESTSC: WHERE_2	a.sgbstdn_term_code_eff = (Select	
DTESTSC: WHERE_3	max(c.sgbstdn_term_code_eff) from sgbstdn c	

- 3. Re-extract the students so they get this variable loaded into their rad_report_dtl.
- 4. On the DegreeWorks application server, go to the *transit* directory

\$cd transit

5. Edit the selection criteria file DAPIDCRI by inserting a line in the file at the location you want the new selection criteria to appear.

/app/transit\$vi DAPIDCRI

ctopus [DWSEED] /app/transit\$ vi DA	PIDCRI	
001 - Student ID	0001001TEXT	10
03 - GOAL: Degree Code	R503005UCX	12
504 - GOAL: Catalog Year	R504008UCX	10
502 - GOAL: School (UG,GR,etc)	R502002UCX	12
505 - GOAL: Student Class Level	R505004UCX	02
506 - GOAL: Term	R506073UCX	08
513 - GOAL DATA: Degree Code	R513005UCX	12
514 - GOAL DATA: Catalog Year	R514008UCX	10
512 - GOAL DATA: School (UG, GR, etc)	R512002UCX	12
:5MJ - GOAL DATA: Major	R5MJ000UCX	12
5MN - GOAL DATA: Minor	R5MN000UCX	12
5PG - GOAL DATA: Program	R5PG110UCX	12
SCL - GOAL DATA: College	R5CL014UCX	06
SCN - GOAL DATA: Concentrations	R5CN013UCX	12
5LL - GOAL DATA: Lib Learning	R5LL016UCX	12
SSP - GOAL DATA: Specialization	R5SP015UCX	12
SAV - GOAL DATA: Advisor ID	R5AV017TEXT	10
555 - GOAL DATA: Stu Status	R5SS103UCX	02

6. The first line to add is a choice to create a picklist of report variables to test against. This will be the name of the report variable created in BAN080. For our example, we will be adding a variable called GRADCODE. The format of the line to add will be as follows:

Field	Length	Description	Value for the example
Element number	4	Must be a valid UCX-	R402
		SYS999 element	
		number	
Filler	3	Normally "–"	-
Literal	30	Displayed in Transit's	Report Code
		picklist	(Free form text here)
Element number	4	Must match first field	R402
Select criteria number	3	No longer used	000
Edit Type	4	UCX, DCM3, PICK,	PICK
		etc. see Edit Type table	
		below	
Picklist file	8	File located in transit	REPOFILE
		directory; only if Edit	(Select a filename to
		Type is PICK	create)
Data length	2	Length of data user is	10
		allowed to enter	

R402 is the record in SYS999 which points to the report_code field in the rad_report_dtl table.

•	•		u≞ ►	 ►	T	able	6	SY:	999 999		R	De	scri	otion:	R	DDat	a Dic	tiona	зry					Ke	y:	R402				-									
s	ivst	tem	٦Ľ	SYS0	101	SY	609	al :	sys	910	l s	YSS	30 I	SYS9	al si	\$935	SY:	5941	l sy	′S942	el s'	/\$976	SYS97	al sys	987	l syses	8 SY	/\$999	ľ										
Г	Īĸ	ey	i.								1.75		Bas	e Nam		[Data	iset N	lame		Ī	Data N	ame		D	escription				Initial	Value	0)ata Ty	vpe Da	ata Length	UCX Cod	de L	JCX Mas	k UCX Le
٢	R	40	0										RAI	DB;	100		RAD	REI	PORT	-DTL		SRN-I);		0:	midex	ID C	ode				R	¢ .	00	14	-	0	0	00
	R	40	11										RAL	DB;	8	5Y599	9																		- 0 ×	1	0	0	00
•	R	40	12		_		_	_		_	_		RAI	DB;								-	_													CROOZ	0	1	30
	R	40	13										RAL	DB;						K	EY:	R402															0	0	00
	R	40	14										RAI	DB;								14		-	×		3 8									TUO16	0	a.	12
	R	40	5										RAI	DB;	-																					7	0	0	00
	R	40	6										RAI	DB;					n	0.10	OT I	MOD	v												<u>.</u>		0	0	00
	R	40	97										RAI	DB;					10	U NI	UT	WODI	1														0	0	00
	R	40	8										RAI	DB;																							0	0	00
Î	R	40	9										RAI	DB;		Base I	lame	RA	DDB		_															Y S987	0	1	30
Ì,	R	41	0										RAI	DB;			1	le.																			0	0	00
	R	41	1										RAI	DB;		taset	vame	Re	10 - R.	RPOR	.T-D	гь;														78998	0	0	00
	R	41	.2										RAI	DB;		Data I	lame	RA	D-R	BPOR	T-C	ODE;															0	0	00
	R	42	0										RAI	DB;		Descr	ption	RE	POR	r Co	de		-														0	0	00
ĺ,	R	42	1										RAI	DB;			63.6 63.6	-																			0	0	00
	R	42	2									_	RAI	DB;		Initial	alue	1																			0	0	00
	R	42	3									_	RAI	DB;		Data	Туре	x																			0	0	00
	R	42	4										RAI	DB;) oto I i	neth	Ini	2																		0	0	00
Ì	R	43	80										RAI	DB;		/ata Li	angun																				0	0	00
	R	43	81										RAI	DB;		UCX	Code	sc	ROO	2																	0	0	00
	R	43	32										RAI	DB;	-	UCX	Mask	01	1																		0	0	00
																ICV I.	math																						
															3	JCX LI	ngu	130																					
																Search	Item	6																					
															D	ataset	Туре	D																					
1																0.4	mize	Ē	VI	N																			

The DAPIDCRI file will now look like this:

0001		Student ID	0001001TEXT	10
R402		Report Code	R402000FICKREPC	FILE10
5 03		GOAL: Degree Code	R503005UCX	12
R504	-	GOAL: Catalog Year	R504008UCX	10
R502		GOAL: School (UG,GR,etc)	R502002UCX	12
R505		GOAL: Student Class Level	R505004UCX	02
R506		GOAL: Term	R506073UCX	08
R513	T	GOAL DATA: Degree Code	R513005UCX	

- 7. Create a second line in the DAPIDCRI file to pick up the value of the code we will be looking for. In our example we will be looking for different GRADCODE values. Some examples include:
 - GR Graduated
 - AG Applied for graduation
 - NG Not able to graduate

This line will contain the following values:

Field	Length	Description	Value for the example
Element number	4	Must be a valid UCX-	R403
		SYS999 element	
		number	
Filler	3	Normally " – "	-
Literal	30	Displayed in Transit's	Graduation Code
		picklist	(Free form text here)
Element number	4	Must match first field	R403
Select criteria number	3	No longer used	000
Edit Type	4	UCX, DCM3, PICK,	PICK
		etc. see Edit Type table	
		below	
Picklist file	8	File located in transit	GCODEFIL
		directory; only if Edit	(Select a filename to
		Type is PICK	create)
Data length	2	Length of data user is	10
		allowed to enter	

The DAPIDCRI file will now look like this:

0001	Student ID	0001001TEXT	10
R402	Report Code	R402000PICKREPOFILE	10
R403	Graduation Code	R403000PICKGCODEFIL	02
R503	GOAL: Degree Code	R503005UCX	12
R504	GOAL: Catalog Year	R504008UCX	10
R502	GOAL: School (UG,GR,etc)	R502002UCX	12
R505	GOAL: Student Class Level	R505004UCX	02
R506	GOAL: Term	R506073UCX (08
R513	GOAL DATA: Degree Code	R513005UCX	12
R514	GOAL DATA: Catalog Year	R514008UCX	10
R512	GOAL DATA: School (UG, GR, etc)	R512002UCX	12

8. Create the picklist files. In this case we will create REPOFILE and GCODEFIL. In the REPOFILE, list any BAN080 variables that you want to select against. For this example we will only have one value in this file: GRADCODE.



The GCODEFIL will contain all of the values for the GRADCODE which we would like to select against.



- 9. Run the programs to load these changes into Transit: ode20get and daprestart
 - a. App/transit\$ ode20get
 - b. App/transit\$ daprestart
- 10. Log into Transit and choose the DAP22 Program
- 11. Use the selection criteria pull down menu



🗷 Start 🛛 🛐 Mon... + 🥥 Z Mon... + 😥 Indox - ... 🕞 Z Mon... + 🛞 Wendow... 💯 SawCodel 🖉 149-24.... 🕐 Degreen... 🏘 SCAO - ... 🔁 SCAO - ... 🔁 SCAO - ... 🔁 SCAO

		-Y		-					_		
Selection	Guestions		Sort								
Criteria	ID File										
DAP22 - Selection Cri	leria List	By: E	quivalent:		Value:						
P402 - Report Cod	. 100	lame	Equal To	2			<u> </u>	1.00			
Element # R402		Apply A	ND (+) to thi	is Criteria 🦵	GRADCOD	t - Gr	aduation Code				
Add Criteria				1007/	goden.	Reienst	Etint Caleria				
+ User Selection Criteria		Equivalent	Value		Description						
1								-			
4											
								1			
								1			
								†			
								+			
								ŧ			
								1			
								ŧ			
								*			
								*			
								†			
								•			
								•			
								•			
End Number Selecte	<u>p</u>		Colonia Herma L	Jaed			Quar Al Criteria				
End Number Selecter			Colona Illens (Jaed			Jear All Criteria				
End Number Selector			Cotena Herra (0	Jaed			Jean All Criteria				

12. Select Report Code and GRADCODE. Press the Add Criteria button

13. Select the Graduation Code and the type of Graduation Code to select against. Press the Add Criteria button and the "And with" check box.

	DAP22	Report DAP22 Gen	erate Audit Processor	<u></u>		_
Selection	Guestions	Sort				
Criteria	ID File					
DAP22 - Selection Cri	eria List	By Equivalent:	Valu	er:		
8400 - Graduation	Code . C	iroup Equal To	2			
Element # R403		Apply AND (+) to I	his Enteria 🗂 📊	- Applied for Graduation	a	
Add Criteria			Moory No	 Graduated Not able to Graduate 		
+ User Selection Orteria		Equivalent Value	Descri	ton		
R402 - Report Code		* GRADCODE	E - Orac	Austion Code		
					1	
End Number Selected	[u]	Ceteria litera	Used .	Des Al Crimis		
End Number Solected	ju	Ceitenia litereta 1	Used	Deer Al Citeria		
End Number Selected		Catoria liena 1	Used	Des Al Crimia		
End Number Selected	_ p	Citoris liena	Used	Deer Al Citeria		
Erd Number Selected		Criteria Itema 1 Launch	Used Cancel	Dise Al Gibnia		

14. Make sure you select the "Find Number Selected" button to check the number of records you have selected.

- WG / 0	DAP22 Report	t DAP22 Generate	Audit Processor		2			
Selection	Guestions	Sort						
Criteria	ID File							
DAP22 - Selection Crite	ria List By:	Equivalent:		Value:				
2403 - Graduation (ode • Group	Equal To	*					
Element #	Appl	AND (+) to this (Criteria 🖓					
Add Criteria				Define	Deletter	Etint Criteria	1	
+ User Selection Criteria	Equivale	nt Value		Description				
R402 - Report Code		GRADCODE		- Graduation	Code			
+ R403 - Graduation Code	1 () () () () () () () () () (AP		- Applied for	Graduation			
End Number Selected	<u>p</u>	Citoria Itomo Use Z	ed :			ng Al Citeria	1	
End Number Selected	<u> </u>	Citeria Itores Use	ed Cane	cet		nge All Citenia]	