

ellucian.

# Degree Works Transit User Guide

Release 5.0.1  
March 2018



© 1999-2018 Ellucian.

Contains confidential and proprietary information of Ellucian and its subsidiaries. Use of these materials is limited to Ellucian licensees, and is subject to the terms and conditions of one or more written license agreements between Ellucian and the licensee in question.

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

Ellucian  
2003 Edmund Halley Drive  
Reston, VA 20191  
United States of America

# Contents

---

<b>Introduction to Transit.....</b>	<b>5</b>
Activating Transit.....	5
Logon Authentication .....	5
Navigating Transit.....	6
<b>Steps to Launch a Report/Process.....</b>	<b>8</b>
Selection Criteria.....	9
Questions.....	13
Sort .....	15
Launching Reports .....	16
<b>Saving Criteria.....</b>	<b>17</b>
<b>Jobs .....</b>	<b>19</b>
<b>Reports/Processes .....</b>	<b>21</b>
ADMIN – Administrative tasks .....	21
UCX01 – UCX Records Modified.....	23
AUD01 – List unhooked and unenforced exceptions.....	24
AUD02 – Delete audits by freeze type and date.....	26
BAN62 – Satisfactory Academic Progress Processor .....	27
DAP16 - Parse Requirements Processor.....	29
DAP21 - Extract Articulated Transfers.....	30

<b>DAP22 - Batch Audits .....</b>	<b>53</b>
<b>DAP27 – What-if Audits .....</b>	<b>59</b>
<b>DAP28 – Alternate What-if Audits .....</b>	<b>62</b>
<b>DAP53 - Plan Assignment Processor .....</b>	<b>65</b>
<b>DAP54 – Create Plan from Template Processor .....</b>	<b>66</b>
<b>DAP58 – Batch Tracking Processor.....</b>	<b>71</b>
<b>DAP59 – Batch Timetabling Processor.....</b>	<b>73</b>
<b>RAD11 – The Traditional Bridge Batch Processor .....</b>	<b>74</b>
<b>RAD30 – Banner Extract and Bridge.....</b>	<b>79</b>
<b>Scribe Management Reports .....</b>	<b>80</b>

## Introduction to Transit

---

Transit is a PC-based graphical user interface that allows users of Degree Works to launch batch reports and processes. The purpose of this guide is to provide basic operating instructions for Transit and to acquaint users with the Transit features and functionality.

### Activating Transit

Transit must be installed on each individual PC where Transit services are to be invoked. Once Transit is properly installed, the Transit icon will display on the PC Desktop. The desktop icon looks like this:



By double-clicking the desktop icon, the user invokes the Transit Batch Reporting Interface.

### Logon Authentication

To start the Transit application, click the Transit icon on the desktop. This will open up the Logon Authentication screen. Enter your *User ID* and *Password* in the appropriate spaces. You can move from one field to the next by using the tab key, by pressing *Enter* or by using the mouse.

If your installation uses multiple Host environments, a drop-down list box will appear. You can open up the drop-down list box by clicking the arrow button and select the appropriate environment either by using the mouse or using the arrow keys to highlight your server choice. If your software installation uses only a single Host location, no Host drop-down list box will appear.



The screenshot shows a standard Windows-style dialog box titled "Logon". The background is a light gray. The text "Please enter your User ID and your Password:" is centered at the top. Below this, there are two text input fields. The first is labeled "User ID" and contains the text "000001". The second is labeled "Password" and contains six asterisks "\*\*\*\*\*". Below the input fields are two buttons: "OK" and "Cancel". At the bottom of the dialog is a dropdown menu with a small arrow on the right side, and the text "IRISDEV" is visible in the dropdown list.

Once you have entered your *User ID*, *Password* and selected your Logon location, click the **OK** button to logon to your account. Once logged on, you will be able to use the Transit application to produce various reports.

## Navigating Transit

Once you have successfully been logged on to the Host via Transit, the following screen will be displayed. This screen contains a menu bar with four drop-down list options. It also contains a toolbar along the top with a set of icons and a drop-down list box for the desired Report. The menu bar drop-down lists include *File*, *Options*, *Tools* and *Help*. The toolbar icons include ***Open PC Report***, ***Save Report to PC***, ***View Jobs***, and ***Help*** along with the Report drop-down list box. Additionally, there is a ***Diagnostics*** icon located on the far right end of the toolbar. Each of these icons will be described in more detail in the following sections. You can select an item by using the ALT key in conjunction with the underlined letter.

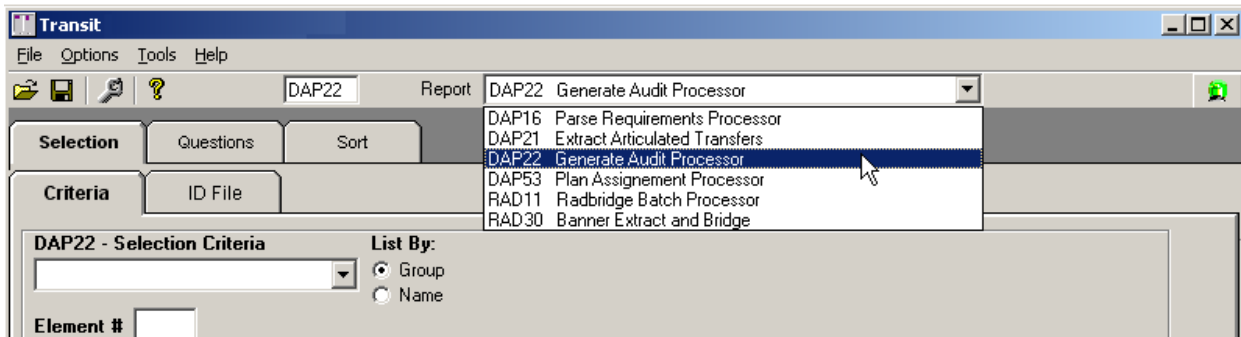


## Menu Bar Map

Menu	Menu Entry	Notes
<u>F</u> ile	Open PC file	
	Save to PC	Saving the selection criteria for a report allows the user to produce the same report at a later date or to use the same selection criteria to produce a different report. Reports that need to be produced at the same time during each academic period (such as a list of students with a certain degree) are good examples of report criteria that should be saved.
	Print Criteria	Allows local PC Printing of Criteria. Only available when Criteria Tab in visible.
	Exit	Exits Transit.
<u>O</u> ptions	Trouble Shooting	<p>The Trouble Shooting item toggles the Diagnostics icon on and off. When Diagnostics is 'on' the icon located on the toolbar turns red and a log file is created on the Host that can be used for debugging purposes. To turn Diagnostics 'off', click the Diagnostics icon again or uncheck the Troubleshooting item from the Options group on the menu bar. The Diagnostics icon will turn green when turned off.</p> <p>Turning on Transit's red light turns on debug in DAP22JOB (dap72 sends param to dapsched and it sets DWDEBUG=1); logdebug/dap22xxx dir is saved and utl39, utl08 and dap32.xml debug files are created</p>
<u>T</u> ools	Job	The Jobs item on the menu bar launches a view listing window which contains a listing of jobs executing in the Host environment. The function of this item is described in greater detail later in this User Guide.
<u>H</u> elp	Help	The Help item on the menu bar invokes the on-line documentation for Transit. You can also view the Help file by pressing the F1 key. The Help file is stored in PDF format and requires an installed copy of Adobe Acrobat reader to view.
	Tech Support	
	About Transit	

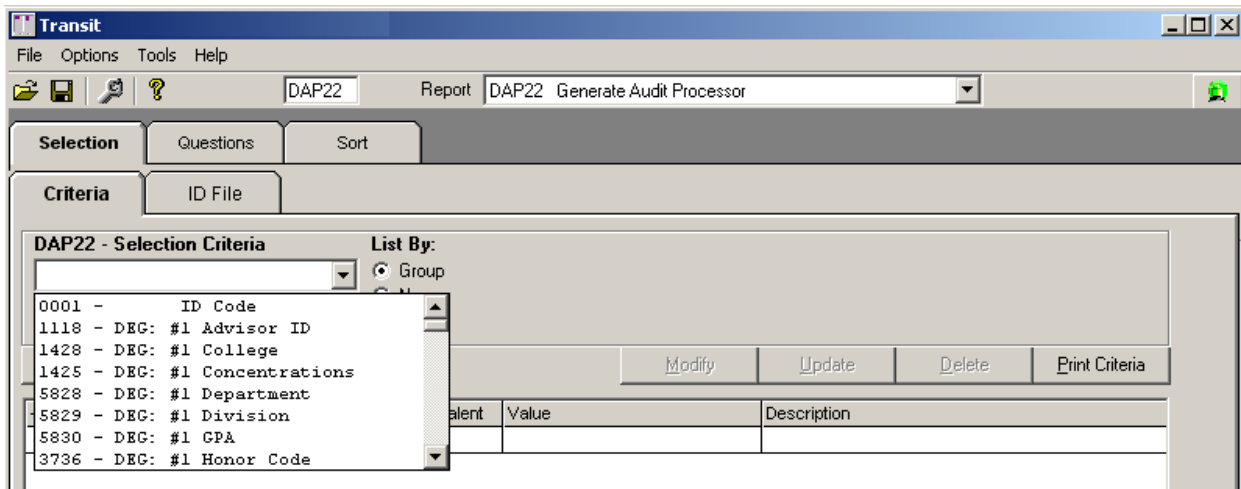
## Steps to Launch a Report/Process

The Transit application can be used to produce batch audits in Degree Works. The first step in producing a batch audit is selecting the correct student population on which to produce degree audits. Select *DAP22 Generate Audit Processor* from the Report drop-down list. Selecting a report can either be done using the scroll bar or by highlighting the Report area and typing the first letter of the report you are looking for. Repeating this will scroll through the various reports that happen to begin with the same first letter.



Selecting DAP22 will bring up the *DAP22 Selection Criteria* screen shown below. This screen is used to define the population of students you wish to use in producing batch audits.

The *DAP22 Selection Criteria* screen is composed of a number of tabs arranged in two rows along the top of the screen. The **Selection** tab brings up two screen tabs. The **Criteria** tab is used to define a set of selection criteria. The **ID File** tab is used for retrieving a set of student ID's defined earlier in Transit or in some other application. The **Criteria** tab is the screen default and will be opened automatically when the **Selection** tab is selected. To define a new set of selection criteria, begin by clicking in the *DAP22 Selection Criteria* box. This will open up a drop-down list box of items that can be used to define the population of students for whom you wish to produce audits.



## Selection Criteria

Items in this drop-down list box are grouped together based upon the type of data they represent.

User Selection Criteria	Equivalent	Value	Description
1112 - DEG: Degree Code	=	BS	BACHELOR OF SCIENCE

For example, items such as *Degree Code*, *Degree Catalog Year*, *Major* and *Minor* all have to do with information about a **Degree** and all have a prefix of **DEG** in the drop-down list box. It is possible to get the information Listed By Group or Listed By Name by selecting the appropriate radio button

DAP22 - Selection Criteria  
List By:  
 Group  
 Name  
Element #

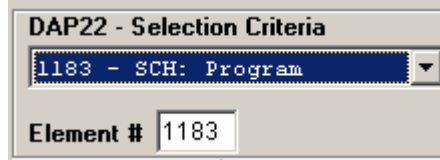
prior to selecting the drop-down list box. Other items such as *Ethnic*, *Handicapped*, *Veteran*, *Religion*, and *Residence* represent Biographical information and have the prefix **BIOG** in this drop-down list box.

DAP22 - Selection Criteria  
List By:  
 Group  
 Name  
Element #

- 1192 - SCH: HS Def-Soc Sci
- 1192 - SCH: HS Def-Soc Science
- 1105 - SCH: Matric Date
- 1182 - SCH: Matric Term
- 1183 - SCH: Program
- 1148 - SCH: School
- 1103 - SCH: Stu Status
- 1184 - SCH: Stu Type
- 1150 - SCH: Term

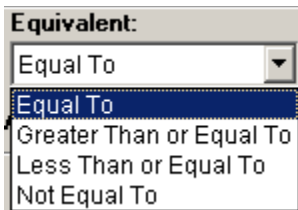
Choosing **Group** as the **List By** option, sorts the drop-down list box by the three or four character prefix while choosing **Name** sorts the list by the field name.

Notice that every item in the drop-down list box has a number associated with it. This number, located to the left of the item in the drop-down list box, represents the *Element Number* of that item in the various databases. If you happen to know the element number of the item you wish to

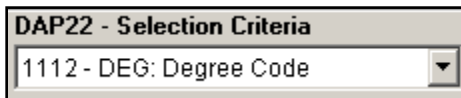


select, you can enter it directly in the **Element #** box located below the *DAP22 Selection Criteria* drop-down list box.

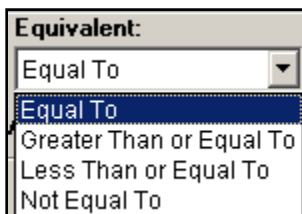
Once you have selected the criteria you wish to add, select the correct logical operator from the **Equivalent** drop-down list box and enter a value in the **Value** field.



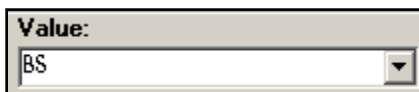
Values can be entered in the *Value* field either by typing a correct value in the text box or by using the drop-down list boxes if one is available. For example, if you wanted to run audits on all students seeking a Bachelor of Science degree, select “*Degree Code*” in the *DAP22 Selection Criteria* box,



select “*Equal To*” in the *Equivalent* field



and type “*BS*” in the *Value* field or select that value from the drop-down list box.



Click the *Add Criteria* button or press *Enter* to add the criterion to the *User Selection Criteria* box. To find the number of students who meet the listed criteria, click the *Find Number Selected* button at the bottom of the screen. This process can be repeated until you have added all the criteria required for your selection.

You can add up to 90 different selection criteria to your criteria list. You can use the *UP* and *DOWN* arrows on the right side of the screen to change the order of the selection criteria. Simply click on the criterion you wish to move and then click the *UP* or *DOWN* arrow to move it up or down in the list of criteria.

When entering values in the *Value* field, it is important to remember that values are being validated as they are entered. For example, if *GPA* is selected as a selection criterion, only numerical data will be accepted in the *Value* field. All data elements that are validated in a UCX table will be formatted correctly once the *Add Criteria* button is clicked. For example, if a “4” is entered in the *Value* field, that number will be converted to 004000 when the *Add Criteria* button is clicked. This is because GPA data values are stored in the database in the form 999v999. No decimal point will appear in the value once it has been added to the list of selection criteria items, but the database will insert the decimal point in the correct place based upon formatting information contained in the associated UCX Tables. Data items are also validated against field length criteria. If a text data field has a maximum length of 12 bytes, the *Value* field will allow a maximum of 12 bytes to be entered.

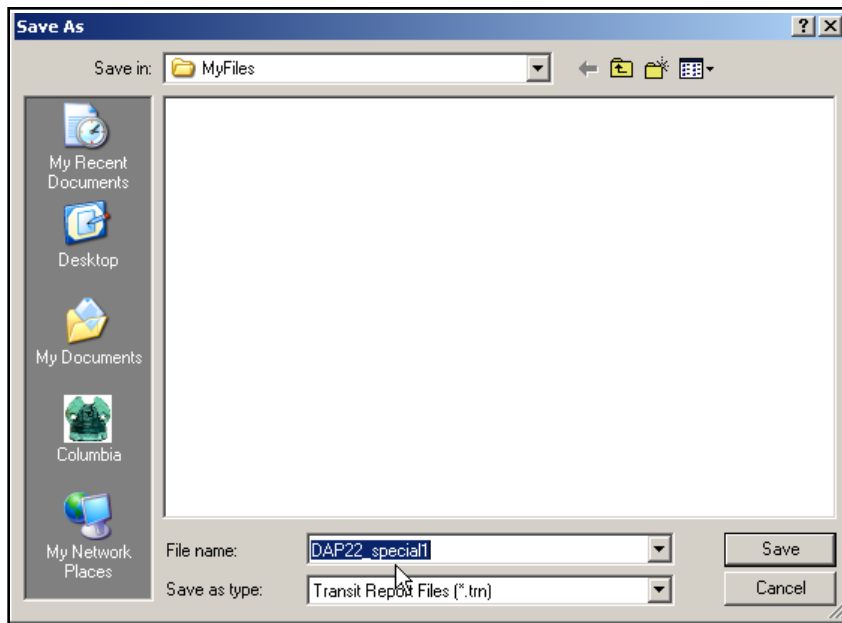
You can also use Transit to select students or blocks that fit within a range of data items. For example, suppose you wanted to select all students who matriculated between September 1, 1999 and June 1, 2001. You would begin by selecting the *Matric Date* item in the *Selection Criteria* field. Select “greater than or equal to” in the *Equivalent* field and enter “September 1, 1999” in the *Value* field. Place a check in the **Apply ADD (+) to this Criteria** checkbox and then click the *Add Criteria* button. Now select *Matric Date* selection item again. Use “Less than or equal to” in the *Equivalent* field and use “June 1, 2001” as the *Value*. Be sure to place a check in the *Apply ADD (+) to this Criteria* checkbox. Notice that a + sign shows up to the left of the items in the list of selection criteria. This means that the flagged items define a range of values. The “+” sign must show up on a least two items sequentially in order to define a range. If entered correctly, all students who fall within this range will be selected as part of the population. A range can be used as a selection criterion on any data items that can be logically sorted. Be sure to use the correct logical operators when defining ranges to be used in selection criteria.

Once all the selection criteria have been added, be sure to click the **Find Number Selected** to verify that your list of selection criteria is valid. To clear all criteria currently in the User Selection Criteria screen, click the **Clear All Criteria** button.

If you would like to leave the DAP22 report screen, click the *Cancel* button located at the bottom of the screen. The report will be discarded and you will go back to the Report Type drop-down list box to select a new report. Any criteria that were previously selected will be brought up again automatically in the next report type selected.

Not all criteria contained within these drop-down list boxes may be functional. For example, if your institution contains only a single school or Division (such as the College of Arts & Sciences), then selecting students using this variable would be meaningless because all students would be in that college. However, if your institution contains multiple Schools or Divisions, then using the School value would allow you to distinguish between students in the College of Arts & Sciences vs. students in the School of Business or Engineering. Similarly, if your institution does not use Location codes, selecting Location as one of your criteria will give erroneous results even though these items may be present in your drop-down list boxes.

Once you have generated a set of selection criteria for a report, it is possible to save that report to a file for future use. To save the report, click on the *SAVE PC* icon on the Toolbar. This will open the *Save As* screen.



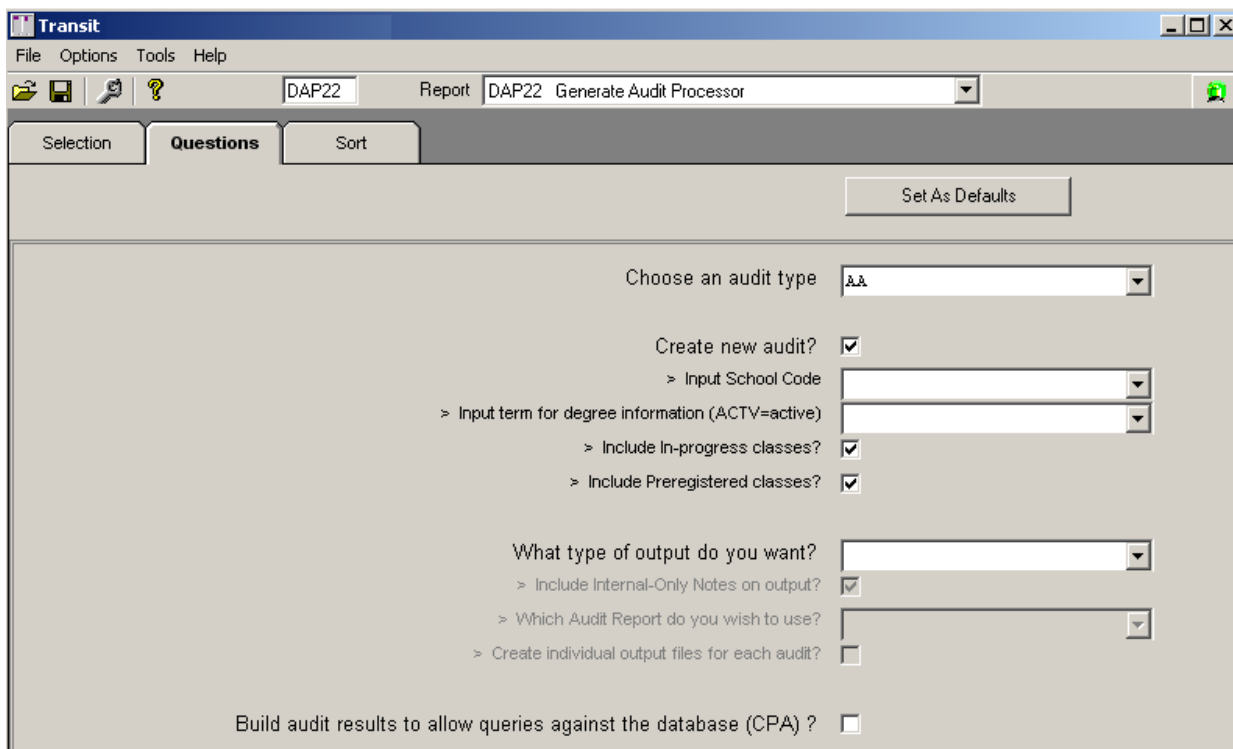
You can then enter in a file name to save the report. Click the *SAVE* button at the bottom of the screen to save this file to your PC. To retrieve this file at a later date, click on the *OPEN PC* icon on the toolbar and select the appropriate file. Transit uses the Windows 'long file name' convention when saving to PC. You are not limited to 8 character file names and may use names that are descriptive of the file you are saving. By default, all search criteria files that are saved to the PC are saved in the Transit folder.

### **Banner search:**

When selecting students the element numbers that start with "B" specify that the searching should be done in Banner. There is no underlying criteria specifying that the students are active students so be careful with your searches. For example, if you select on major of history you will find new students, current students and perhaps graduated students in your selection.

## Questions

Once you have selected the population of students you wish to run batch audits on, you can create and print out those audits in a batch process, defer the printout so that it prints at a specified time in the future or save those selection criteria to a PC file. To create the audits, select the **Questions** tab in the *DAP22 Generate Audit Processor* screen. This will bring up the questions screen shown below. This page contains questions relevant to the type of report you are producing. Notice that some of the questions make reference to a specific UCX table. These questions will only allow values that have been defined in the referenced table.



The questions shown on this screen are dependent on the type of report being produced. The drop-down list boxes associated with these questions are populated with valid values where appropriate. Some drop-down list boxes contain no values. For these drop-down list boxes, there is a reference to the UCX Table that holds the appropriate values for that drop-down list box. Please refer to that UCX Table to find out what the valid entries are for that item.

Throughout Transit, anytime there is a box requiring a check mark, the standard is blank = a *NO* (N) response and a check = a *YES* (Y) response. Many times these boxes can be defaulted to a particular response if one is more frequent than another.

For many of the reports, there is a dependency relationship between the questions. By answering a parent question a certain way may cause child questions to become enabled or disabled. A question that is disabled should be ignored by the user since it is ignored by the report program. Parent questions appear with a bigger font than their child questions and are listed before them on the page. In addition to child questions having a smaller font they are preceded with a ">" character.

Some reports contain questions that must be answered. These required fields appear with an asterisk after the question text. If the user attempts to launch the report with a required field left blank Transit will inform the user that required fields were left blank and will mark the questions in red. If the question is disabled the field will not be considered required and thus the user may launch the report without having filled in the question. No check-box fields are marked as required since they are always either Yes or No.


The **Input term for degree information** item determines which GOAL-DTL record to use in generating degree audits. Select the appropriate term from the list of values in the drop-down list box. If a value is entered and a student has no record with that value, the student will not be included in the batch audit report. A standard list file will be generated detailing any students who were left out of the report along with an explanation of why they were not included. This file can be retrieved from the Host account where the batch job was launched. See the Jobs section of this User Guide for more information on retrieving this file.

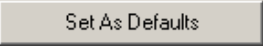
The **Input School Code** item determines which schools to include in batch report. You can specify all undergraduates, all graduates or all students in any school by selecting the appropriate value.

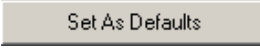
Batch degree audits can be produced using a number of different report types. You can specify which report type to use by selecting the appropriate value in the **What type of output do you want** drop-down list box. You can also specify whether Transit should create a new audit report by placing a check in the appropriate check box. If this box is not checked, a student will be included in the batch report only if that student has an existing audit report in the DAP database.

By default Internal-Only Notes are included on batch audit reports. To prevent these Notes from being included in the audit reports, remove the check mark from the **Include Internal-Only Notes** check box.

Notice the  button at the top of the screen.

If you want to establish one or more default answers for a particular report, enter the data into those fields or check the check boxes for the items you wish to set as defaults and click . Each time you select that report in the future, those fields will appear with the selected defaults already entered. These will only appear on the PC which initiated the defaults and they will exist until such time as they are changed from that PC.

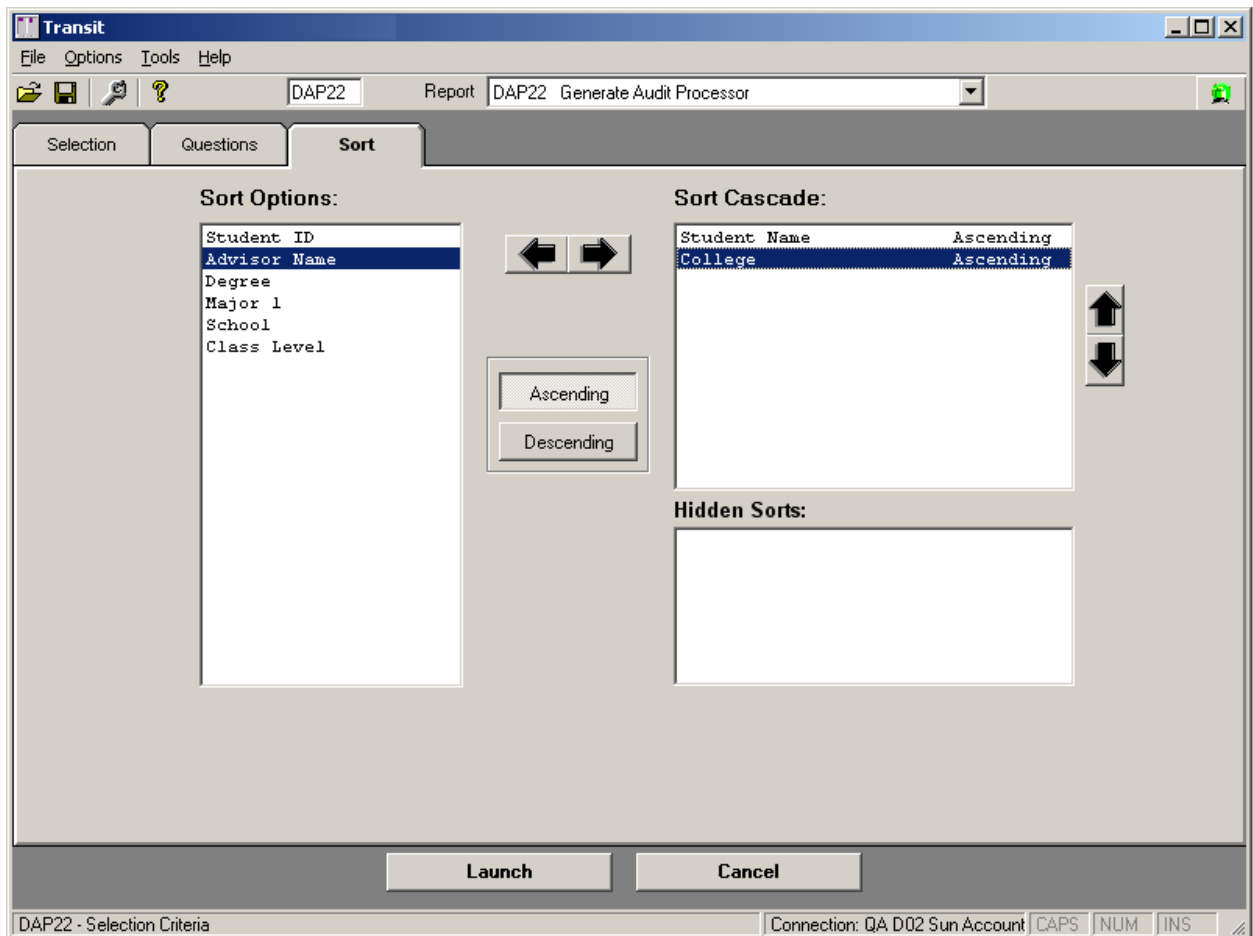
If you decide on a different set of defaults, simply enter the new data or check or uncheck the check boxes and click . The new defaults are now in effect.

If you simply want to clear all defaults, clear all the fields and click .

## Sort

Once you have selected your population of students and entered the information necessary to create the degree audits, you will need to determine how the report should be sorted. Click the Sort tab to bring up the following screen. The sort screen allows you to determine which criteria should be used in sorting the output and the order in which those criteria should be used. Notice that on the left side of the screen, elements used in the selection criteria appear as sort criteria in the Sort Options box.

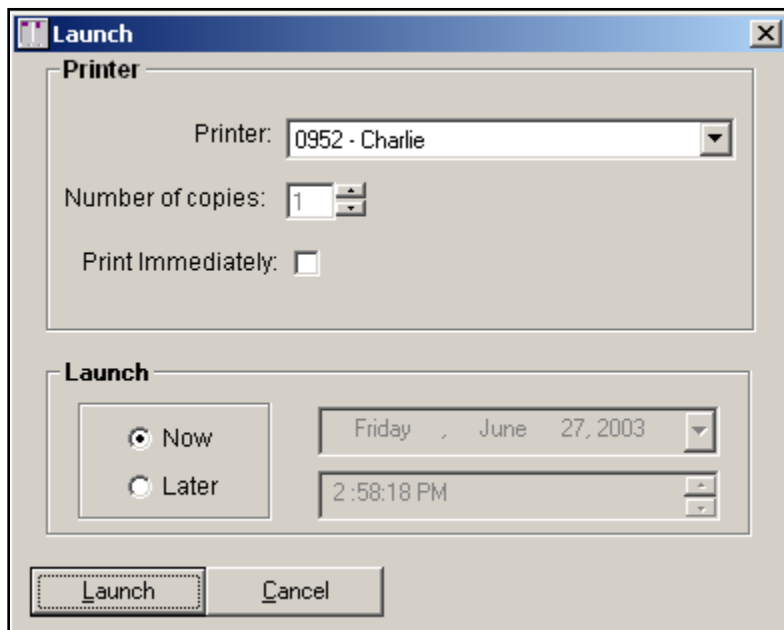
Any element used as a selection criteria item can also be used as a sort criterion. To use one of these elements as a sort criterion, highlight the item by selecting it with the mouse or arrow keys and then click the right arrow to move it from the Sort Options box to the Sort Cascade box. To delete an item in the Sort Cascade window, select it with the mouse or arrow keys in the Sort Cascade window and click the left arrow button to move it back to the Sort Options box. As each item in the Sort Cascade is selected, you have the option to choose the Ascending or Descending option button located between the two windows. The Sort Cascade box will show which direction the sort will take place for that item. Once an item has been selected and moved to the Sort Cascade window, it is not possible to change the Ascending or Descending option associated with it. To change an item that has already been selected from Ascending to Descending, deselect it by moving it back to the Sort Options box. Then choose the Descending option button prior to moving it back to the Sort Cascade box.



## Launching Reports

Once all of the *Selection Criteria* have been chosen and the *Sort* and *Address Criteria* have been determined, click the **Launch Report** button located on the bottom left of any of the tab screens. This will bring up the **Launch on Host** screen shown below. A printed report will **only** be generated if the "Produce Printed Report" checkbox has been selected on the Questions screen.

To produce a printed report, select the printer you wish to use for generating the report using the printer drop-down list box. Only network printers defined in your configuration tables can be used for printing batch reports. To have a printer added to the Printer drop-down list box, contact your system administrator. Select the number of copies to be printed of each report. To produce the reports immediately upon launching, select the **Print Immediately** checkbox and click the **Now** option in the Launch window. Click the **Launch** button and the job will be executed on your Host computer.



The screenshot shows a dialog box titled "Launch". It is divided into two main sections: "Printer" and "Launch".

- Printer Section:**
  - Printer: 0952 - Charlie (dropdown menu)
  - Number of copies: 1 (spinner control)
  - Print Immediately:
- Launch Section:**
  - Now:  (selected)
  - Later:
  - Date: Friday, June 27, 2003 (dropdown menu)
  - Time: 2:58:18 PM (spinner control)

At the bottom of the dialog are two buttons: "Launch" and "Cancel".

If you would like to defer the printing until a later time, select the **Print Immediately** checkbox but select **Later** in the Launch window. Selecting the **Later** option button will activate the scheduling windows to the right. Select the date and time to launch the print job and then click the **Launch** button. The print job will be scheduled to run at the specified time. Technically, this does not defer the printing of the report, it defers the actual running of the report. Once the report has been processed, it prints immediately upon completion but at the time entered in the Launch window (using the Later option).

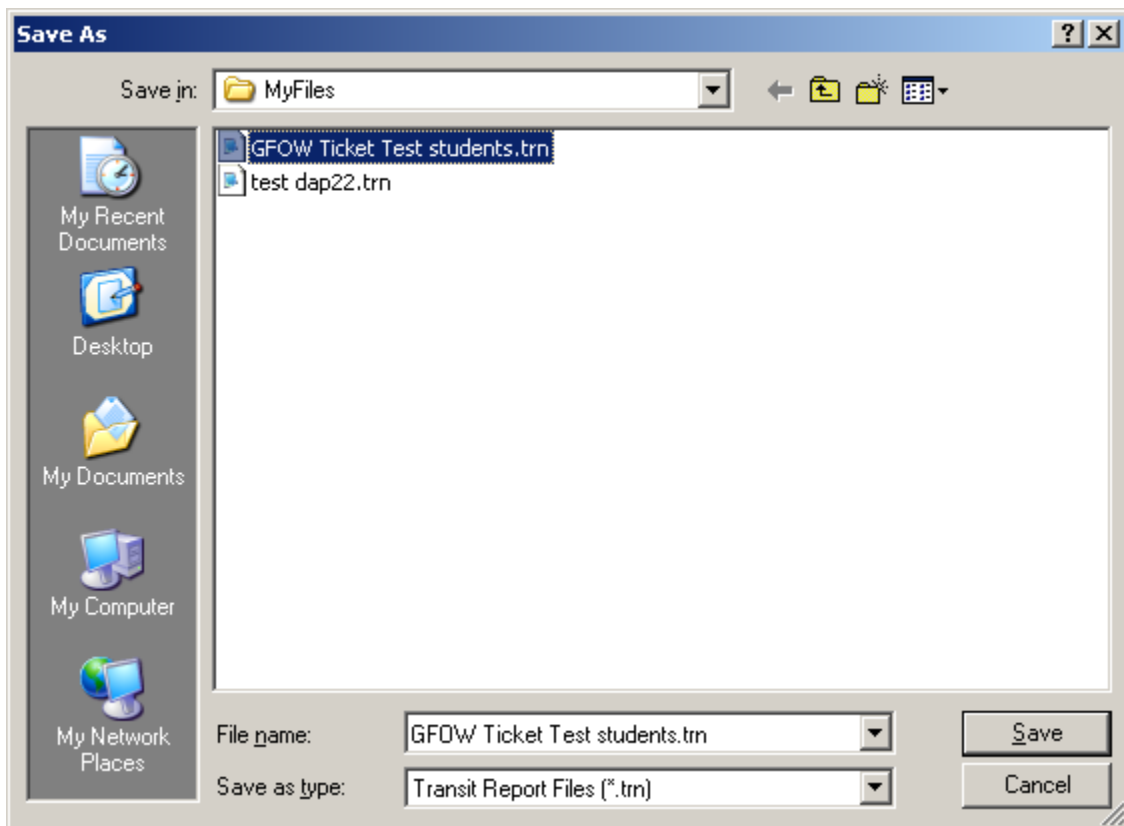
To launch a job without printing a report, uncheck the **Print Immediately** checkbox and choose one of the options in the Launch window. Click the **Launch** button to launch the job.

## Saving Criteria

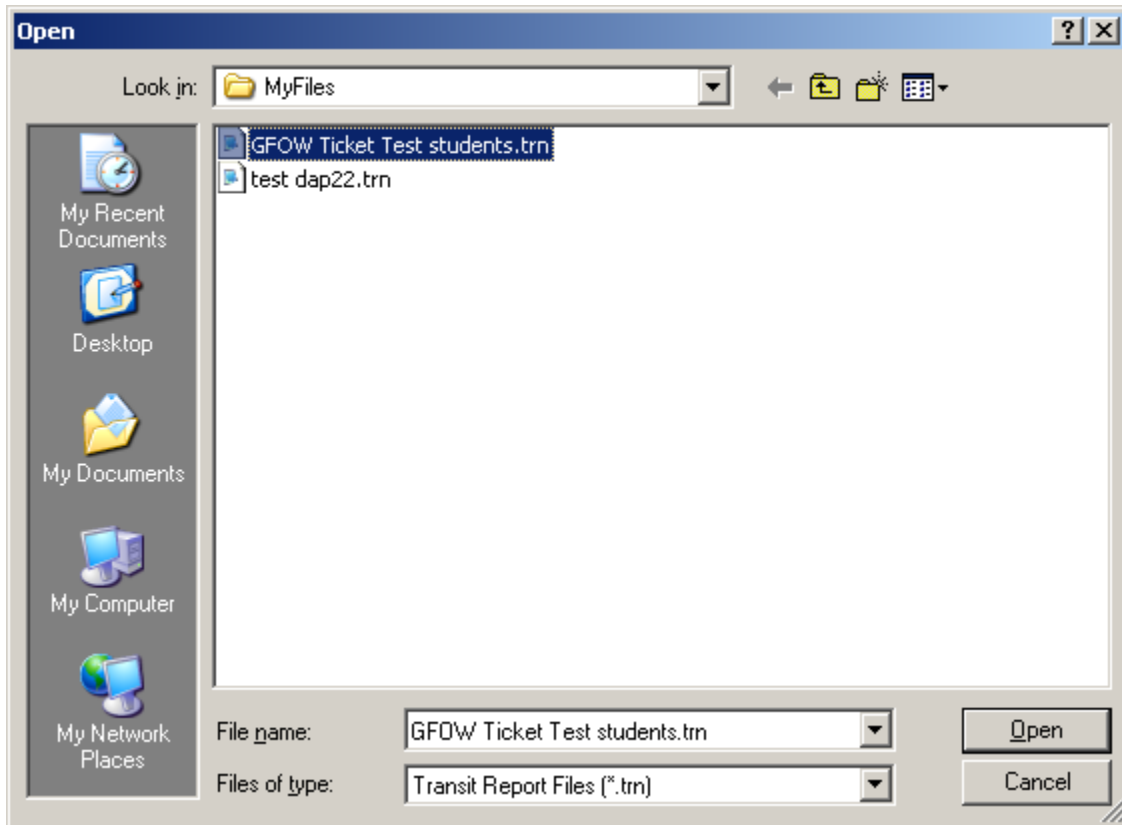
---

This screen allows you to save batch criteria (including Selection items, answers to Questions, Sort order, and Address cascade) to a file located on a PC directory (local or shared). By default, files are saved to the MyFiles directory where the Transit application is installed locally. All files saved using Transit will be given a file extension of **.trn**.

The Save PC item in the drop-down menu activates the same function as the Save PC icon. To save a file of criteria to your PC, click the *Save PC* icon (or select that item from the drop-down menu). This will open up the *Save PC* screen shown below. To save a set of selection criteria (which could even be ID numbers), select the directory to save the file to in the browse window. Enter a filename in the *File Name* field at the bottom of the screen and click the **Save** button or press *Enter*.



The Open PC menu item activates the same function as the *Open PC* icon on the toolbar. Clicking this button will bring up the following screen.



Any files which have been saved to the PC via Transit will be listed in the browse window. To retrieve a set of previously saved criteria, select the file from the browse window and click the **Open** button at the bottom of the screen. If the file is in a directory different than the default directory, you can navigate to that directory using the directory drop-down list box or navigation icons located at the top of the screen. Files with a file extension other than **.trn** cannot be opened using the Transit application.

# Jobs

Once you save successfully completed all of your selection criteria, answered the appropriate questions in the *Questions* Tab and selected the appropriate sort criteria and printer, click the *Launch* button to launch the job. A screen will pop up showing the number of the Job just launched. Once launched, it is important to make sure that the job has actually been executed on the Host computer. To accomplish this, select the **Job** item on the menu bar and click on Jobs in the drop-down list box or click on the **Jobs** icon. This will bring up the Jobs screen shown below. Here, you can identify the job that you just executed and verify that the job is running and is not in a wait state.

As a batch job is executing, a spool file is created which contains details about the job and any errors that occurred during the run. To view the spool file, first identify the job number for the job you just launched. Select that job by double-clicking on the appropriate entry or by highlighting the entry and clicking the **View Listing** button. If the job is still executing, the standard list will be incomplete. You can determine how far the job has proceeded by identifying where the standard list has ended. To see more contents of the spool file, cancel this present view, go back to the Jobs screen and select the spool file again and click on the *View Listing* button. This spool file will contain a little more data than the previous one since the job continues to execute while you are viewing the spool file. The spool files generated by Transit are temporary files and will eventually be deleted from the Host unless you specifically choose to save them for later use. Contact your system administrator for information on saving and retrieving old or deleted spool files.

Output#	Report	Report Title	Date	Time	Job#	KBytes	Printer
0136A	DAP16	Parse Requirements Action List	Nov 26	00:38	0136A	26	
0136L	DAP16	Parse Requirements Listing	Nov 26	00:38	0136L	68	
0045A	DAP53	Plan Assignment Action List	Nov 19	14:00	0045A	1	
0045L	DAP53	Plan Assignment Listing	Nov 19	14:00	0045L	1	
0044A	DAP53	Plan Assignment Action List	Nov 19	13:56	0044A	1	
0044L	DAP53	Plan Assignment Listing	Nov 19	13:56	0044L	1	
0043A	DAP53	Plan Assignment Action List	Nov 19	13:55	0043A	1	
0043L	DAP53	Plan Assignment Listing	Nov 19	13:55	0043L	1	
0042A	DAP53	Plan Assignment Action List	Nov 19	13:50	0042A	1	
0042L	DAP53	Plan Assignment Listing	Nov 19	13:50	0042L	1	
0041A	DAP53	Plan Assignment Action List	Nov 19	13:50	0041A	1	
0041L	DAP53	Plan Assignment Listing	Nov 19	13:50	0041L	1	
rayA	DAP16	Parse Requirements Action List	Nov 19	13:37	rayA	60	
rayL	DAP16	Parse Requirements Listing	Nov 19	13:37	rayL	28	
0036A	DAP53	Plan Assignment Action List	Nov 18	14:22	0036A	1	
0036L	DAP53	Plan Assignment Listing	Nov 18	14:22	0036L	1	

Buttons: View Listing, Print, Abort, Delete, Refresh, Close

For DAP16 reports, you can also view the actual report once the job has finished. To view the report, select the DAP16 report you wish to view and click the *View Listing* button. While DAP16 reports can be viewed in this manner, DAP22 reports cannot be viewed. DAP22 reports contain printer control commands that cannot be viewed on the screen. The standard lists for both DAP16 and DAP22 Jobs can be viewed using Transit.

While a client site may have a few dozen to several hundred blocks of Scribe requirement text, that same site may have tens of thousands of students. Consequently, batch degree audits (DAP22 jobs) may take much longer to run than batch parsing and printing of Scribe blocks (DAP16 jobs). Batch degree audits that involve large numbers of students are best scheduled to run during off-peak computer center hours. These jobs may require a significant amount of time to complete depending upon the number of student records involved in the batch.

If you launch a batch job erroneously, you can abort that job using the **Abort** button on the Jobs screen. The *Abort* button is active only when a job which is scheduled to run or is currently running is selected in the Jobs screen, and your logon ID has been given the appropriate capabilities to abort a batch job, select the job you wish to cancel and then click the *Abort* button. The screen will automatically be refreshed after aborting a job, allowing you to verify the job was aborted. This is HP3000 MPE operating system specific. For clients using systems with operating systems other than MPE they will need to consult the *Degree Works Technical Guide* for information on viewing and aborting scheduled jobs.

### View Listing

DAP16JOB Started Fri Dec 14 01:29:02 EST 2007

Type	Value	Catalog Years	ID	Status
CONC	107	111 :1980	RA000006	Parsed okay
DEGREE	000000	0000 :9999	RA000015	Parsed okay
DEGREE	AA	111 :1981	RA000007	Parsed okay
DEGREE	AS	20012002 :99999999	RA000009	Parsed okay
DEGREE	BA	0000 :9999	RA000008	Parsed okay
DEGREE	BA	19961997 :99999999	RA000011	Errors found
DEGREE	MA	0000 :9999	RA000017	Parsed okay
MAJOR	0000	1948 :1981	RA000003	Parsed okay
MAJOR	0000	1948 :1981	RA000004	Parsed okay
MAJOR	ACCT	2001 :99999999	RA000010	Parsed okay
MINOR	105	111 :1973	RA000005	Parsed okay
OTHER	BARBJANSEN	0000 :9999	RA000016	Parsed okay
OTHER	CIVILSTUDIES	19941995 :99999999	RA000014	Parsed okay
OTHER	GEMAJOR	19941995 :99999999	RA000012	Errors found
OTHER	GEREQ	1948 :99999999	RA000001	Parsed okay
OTHER	NOGENED	19981999 :99999999	RA000013	Parsed okay

Blocks that Parsed Without Errors: 14  
 Blocks that Parsed With Errors: 2  
 Total Requirement Blocks Parsed: 16

DAP16JOB Ended Fri Dec 14 01:29:03 EST 2007

Print from Host    Close Window    Copy Selected Text    out    Zoom    in

## Reports/Processes

---

### ADMIN – Administrative tasks

You may run certain administrative tasks through Transit. Instead of contacting your IT staff to run these commands you may use Transit to run them yourself.

The ADMIN job allows you to run the following scripts on your Degree Works application server:

- **blockunload** Run dapblockunload (archive all blocks)
- **dap22dbg** Run audit and capture debug files
- **daphits** Show the web hits on audits, notes, etc
- **daprestart** Restart the dap/PC daemons
- **rad30dbg** Run Banner extract and capture debug
- **ucx12job** Run the Transit refresh script
- **webanalyze** Analyze the active logdebug/web.log
- **webrestart** Restart the web daemons
- **weblogon** Test a user's logon to view key list

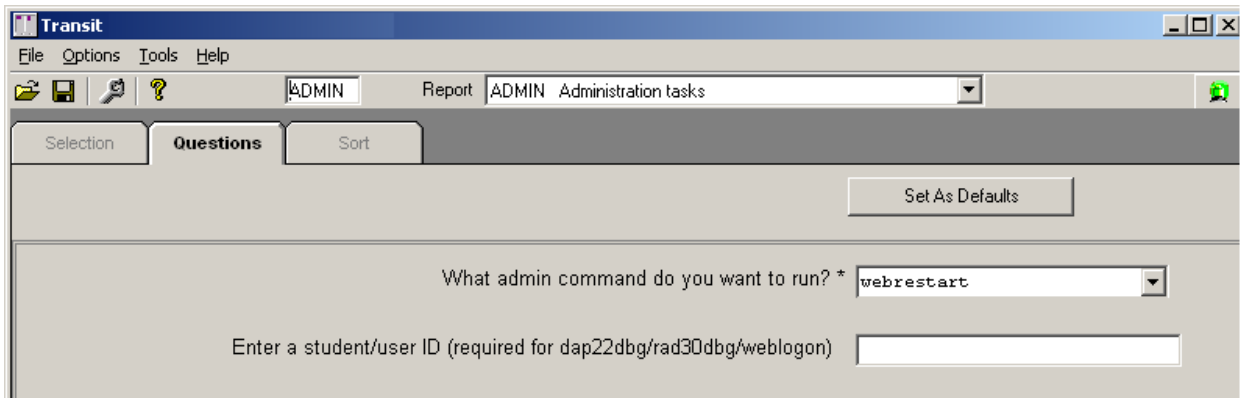
The **restart** scripts should be scheduled to run every night but you may have a need to restart the servers in the middle of the day. This is useful when you do not have command prompt access on the Degree Works server.

The **weblogon** script is useful for examining the security keys that will be given to a particular user when they logon to Degree Works. You need to supply the user's logon ID in the field provided. You do not have to provide the password as this special tool does a pseudo logon that does not require the password.

The **dap22dbg** (run an audit) and **rad30dbg** (run the Banner extract) scripts are useful when working with the Action Line when working on a Service Request. You need to supply the student ID in the field provided when running each. When the job is complete you can view the log file via Transit. The bottom of the log file will list the tar file that was created. You can either contact your school's IT team to have this file download (as binary) to a location you can access or you may download it yourself if your school has provided you with an FTP tool.

When launching **daphits** you can enter "reset" in the text box field to reset each of the hit counts.

To gain access to the ADMIN tool you need to have the PTSADMIN key added to your security record.



## UCX01 – UCX Records Modified

You may also run UCX01 to get a list of all UCX records that were modified between a specific date range. The report generated will list the UCX table number, UCX key and modify date.

Transit

File Options Tools Help

UCX01 Report UCX01 List UCX Records modified in a certain date range

Selection Questions Sort

Set As Defaults

Get the UCX records with changes made on or after this date \* 1 / 1 /2011 Date Today Blank

and on or before this date 4 /18/2011 Date Today Blank

Launch Cancel

A sample of the report produced:

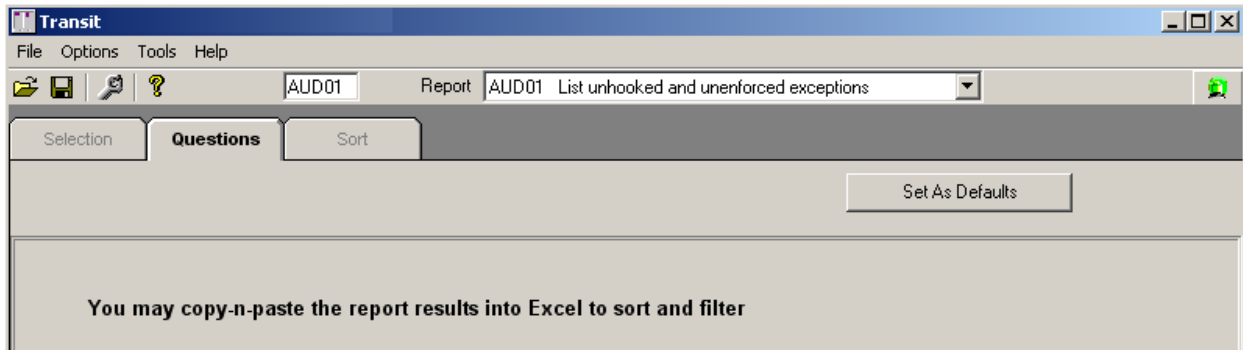
UCX_AUD034	RG	09-MAR-11
UCX_AUD034	ST	09-MAR-11
UCX_AUD034	TM	09-MAR-11
UCX_AUD047	1	09-MAR-11
UCX_AUD047	2	09-MAR-11
UCX_AUD047	3	09-MAR-11
UCX_AUD047	4	09-MAR-11
UCX_AUD047	5	09-MAR-11
UCX_AUD047	6	09-MAR-11
UCX_CFG020	BANNER	15-APR-11
UCX_CFG020	DAP13	06-JAN-11
UCX_CFG020	DAP14	15-APR-11
UCX_CFG020	PLANNER	25-MAR-11

## AUD01 – List unhooked and unenforced exceptions

The AUD01 report will show all unhooked as well as all unenforced exceptions. It is suggested that this report be run on a regular basis in order to identify if exceptions have become unhooked. However, if you use label-tags on all of your labels exceptions should rarely become unhooked. See the *Label Tags* section in the *Scribe User Guide* for more information.

This report is run against all records, as there are no selection options from Transit.

To gain access to the AUD01 report you need to have the PTSAUDIT key added to your security record.



A sample of the report produced:

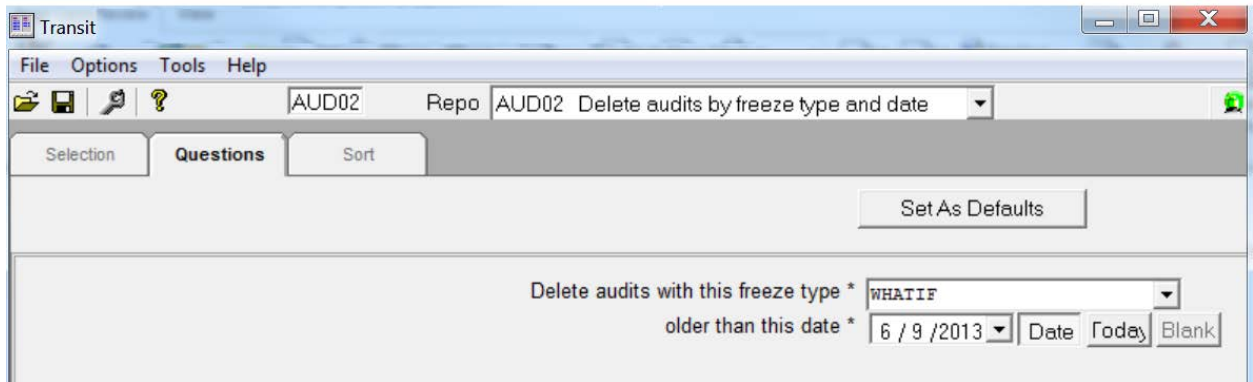
Unhooked exceptions					
6789	RR	BS	RA000112	UN	testing the RR except BH
6789	FC	BS	RA000112	UN	testing fop exceptions
1633092319	FC	BA	RA000112	UN	Force complete this rule
1646953703	FC	BA	RA000235	UN	force complete
1960	RR	BS	RA000263	UN	Replace MATH 1305 with BUS
2397					
1493208187	NN	BS	RA000515	UN	REM CIS3281 AND REQ 2 CLAS
NOT 1					
1493208187	AH	BS	RA000515	UN	apply hist 3500 from elec
1493208187	AA	BS	RA000515	UN	allow psyc 1100 wth 5 cre,
	stu				
					has 4 cred in elec
1493208187	FC	BS	RA000515	UN	testing force complete
1493208187	RR	BS	RA000515	UN	replace cs 3660 w/ engl
3003 from					
					electives
001234	FC	BS	RA000515	UN	Force complete this rule
001234	FC	BS	RA000515	UN	Force complete this rule
Unenforced exceptions					
6789	RR	BS	RA000112	R2	Test--replace COMM 1000
with ART					
					1020

1969 here.	AA	BS	RA000901	R2	Allow hist 3400 to apply
B123456	NN	BA	RA001363	R2	testing 1-3H96G8
B123456 high	FC	BA	RA001363	R2	Macroecon course taken in school
B003 complete -	FC	BA	RA001399	R2	Please force this rule per Stephen's

## AUD02 – Delete audits by freeze type and date

The AUD02 processor allows you to delete frozen audits older than a certain date.

The freeze type is a required field to prevent you from deleting all audits by accident. The primary use of this is to clean out certain frozen audits on a periodic basis. This is especially handy when frozen audits are created using DAP27 to run batch what-if audits.



A sample of the report produced:

```
Deleting audits older than 20130609 with a freeze-type of WHATIF
DAPDELAUDITS deleted 7 audits from the database
```

## BAN62 – Satisfactory Academic Progress Processor

This processor is further described in the *Banner Considerations Technical Guide*.

The screenshot shows the Transit application window for the BAN62 Banner SAP Processor. The window title is "Transit" and the menu bar includes "File", "Options", "Tools", and "Help". The toolbar contains icons for file operations and a help icon. The main area has tabs for "Selection", "Questions", and "Sort", with "Questions" currently selected. A "Set As Defaults" button is located in the top right of the main area. The configuration options are as follows:

- Create new audit?
- > Refresh data from Banner before running audits?
- > Input School/Level Code
- > Include In-progress classes?
- > Include Preregistered classes?
- > Freeze these new audits? - choose a freeze type
- > Describe these audits (optional but ideal when freezing audits)
- Enter the term this processor was run - saved on SHRSAPP

Below the configuration options, there is a text block: "You may choose to run new audits and refresh from Banner. If you choose not to run new audits the most recent audit for each student will be used to populate the SAP tables." At the bottom of the window are "Launch" and "Cancel" buttons. The status bar at the very bottom shows "BAN62 - Selection Criteria".

Run-time questions for BAN62 will be:

### Create new audit?

When this box is not checked, BAN62 will process the most recent audits for the selected pool of students, which is based on the selection criteria entered under the Selection Tab. None of the other check boxes will be available.

When this box is checked, new audits will be created for the selected pool of students. In addition, the following options become available:

**Refresh data from Banner before running audits.** When checked, the student data will be refreshed from Banner before a new degree audit is processed.

**Input School Code.** The selected pool of students will be filtered to the School Code (Banner Level) selected in the picklist.

**Include In-progress classes?** In-Progress classes will be included in the newly generated audit.

**Include Preregistered classes?** Preregistered classes will be included in the newly generated audit.

**Freeze these new audits? Choose a freeze type.** The newly generated audits can be frozen with a new freeze type "SAPFRZ".

**Describe these audits.** It is optional to include a freeze description, but it is recommended that when running the BAN62 processor that the associated audits are kept for reference.

**Enter the term this processor was run.** Enter the current term in which you are running the BAN62 processor. This term value will be written to the SHRSAPP\_TERM\_CODE\_EVAL column in Banner. This is a required field which reflects the term for which academic progress is being evaluated.

## DAP16 - Parse Requirements Processor

DAP16 does not have any questions to be answered in Transit. When you open DAP16 you simply click Launch to initiate it.

DAP16 parses the `dap_req_block` table if the UCX-CFG020 DAP14 UseDapReqBlock flag is Y. DAP16 parses the `dap_req_mst/dap_req_text_dtl` tables if the UCX-CFG020 DAP14 UseDapReqBlock flag is N.

DAP16 processes all blocks found in the database. Each block is reparsed and the `dap-req-crs-dtl`, `dap-req-link-dtl` and `dap-result-dtl` tables are updated. In addition the syntax and remarks trees in the `daptrees` directory are replaced. If any errors were found during the parsing phase, the parse status is updated with "NO"; if no errors were found the status value is set to "OK".

DAP16 can also be run from the host prompt using the "dap16all" script if needed. In addition, a subset of blocks can be run by setting the DAP36\_CRITERIA variable to choose only the blocks you want. You may also override the default order-by clause, for example:

```
$ export DAP36_CRITERIA="dap_block_type = 'MAJOR' "  
$ export DAP36_ORDER_BY="dap_block_title"  
$ dap16all
```

The default order-by used by DAP16 is: `dap_block_type`, `dap_block_value`, `dap_cat_yr_start`, `dap_cat_yr_stop`.

For each block that DAP16 parses a line will appear in the log file showing the block type, value, cat-yrs, req-id and parse status, for example:

Type	Value	Catalog	Years	ID	Status
COLLEGE	SCI	19941995	:99999999	RA001267	Parsed okay
COLLEGE	SCI	20042005	:99999999	RA001269	Parsed okay
COLLEGE	SCI	20102011	:99999999	RA001268	Parsed okay
CONC	ACCT	19941995	:19951996	RA000600	Parsed with errors
CONC	ACCT	19961997	:19992000	RA000203	Parsed okay
CONC	ACCT	20002001	:20022003	RA000506	Parsed okay

When DAP16 is done a summary line will be shown in the log file:

Blocks that Parsed Without Errors:	794
Blocks that Parsed With Errors:	134
Total Requirement Blocks Parsed:	928

When DAP16 is finished the `dap16action` script will run and create a list of all the blocks in the database that have a "NO" parse status. In addition, the same summary line shown in the log file will appear in the action file. Both the action file and log file can be viewed in Transit.

To view and fix the parse errors Scribe should be used. In Scribe you should search on blocks with parse errors and fix each block one by one.

## DAP21 - Extract Articulated Transfers

The DAP21 batch program is the processor used to create an XML file of articulated transfer data from DAPDB. DAP21 is intended for use by sites using Degree Works Transfer Equivalency and wishing to FTP the articulated transfer data back to the student system. DAP21 is launched from Transit after selecting the students to be processed.

**Selection criteria** from the dap\_applicnt\_mst, dap\_appdata\_dtl, dap\_college\_dtl and dap\_transfer\_dtl **must be** used to define the desired group of students. If no selection criteria are used then only the TreqStatus of “AR” and “RO” (If the Re-roll flag is “Y”) will be used to screen students. Run-time questions for DAP21 will be:

**Name of output file?** The XML file name will be no longer than 8 bytes long. The file will be output to the \$ADMIN\_HOME/data directory. If the file name is not supplied then it will default to dap21nnnn.xml, where nnnn is the last 4 digits of the Process Identification Number (PID). For example, if the PID is “27865” for DAP21 the output file name would be dap217865.xml.

**Re-roll students who were already rolled?** Y = Selected students with a dap\_applicnt\_mst treq\_status = “RO” or “AR” will be included. N = Only selected students with a dap\_applicnt\_mst treq\_status = “AR” will be included. If left blank then “N” is used as the default Re-roll Student parameter.

DAP21 will extract the data for the selected students, screened by the Re-roll Student run-time response, to an XML flat file. DAP21 will also change the dap\_applicnt\_mst treq\_status from “AR” to “RO”, set the dap\_roll\_date to the current system date, set the dap\_roll\_time to the current system time and set the dap\_what to “TREQEXPT”. DAP21 will not produce a printed report but will display a tally of the number of records processed in the dap21nnnnA.act (summary) and dap21nnnnL.log (detail) files.

The XML file produced by DAP21 will contain 5 types of records defined in the AcademicRecord\_v1.5.0.xsd PESC schema – “custom” Transfer Equivalency elements have been added for the Applicant Master data, Applicant Detail data, School data and Course data where no corresponding element exists:

<Student> <Person>	- ID code and name from the rad_primary_mst
<Student> <ApplicantMasterData>	- data from the dap_applicnt_mst
<Student> <ApplicantDetailData>	- major/minor/concentration data from the dap_appdata_dtl
<Student> <AcademicRecord> <School>	- transfer school data from the dap_college_dtl
<Student> <AcademicRecord> <Course>	- transfer course data and tests from the dap_transfer_dtl

All records from each table for each student selected from DAPDB/RADDB will be included in the XML output file. It is up to each site to determine which elements out of the generated XML file are to be imported into the local student system.

DAP21 will not generate any errors. No data validation will occur. It is assumed that the data in the database has already been validated. The output file will be named dap21nnnn.xml, where nnnn is the last 4 digits of the job number, unless a file name was supplied in response to run-time question. The XML output file will be written to the \$ADMIN\_HOME/data directory.

## Sample Output XML Data File

DAP21 uses the AcademicRecord\_v.1.5.0.xsd PESC schema for its output XML definitions. The "ArticulationExportEnvelope" has been created to export all selected students and include them in the same output XML file. Many *custom* elements have been added for the Degree Works Transfer Equivalency database tables being exported. Most of the items defined below are *custom* except where they are noted as *standard*.

```
<xs:complexType name="StudentType">
  <xs:annotation>
    <xs:documentation>Student Type</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Person" type="AcRec:PersonType"/>
    <xs:element name="ApplicantMasterData" type="AcRec:ApplicantMasterData"/>
    <xs:element name="ApplicantDetailData" type="AcRec:ApplicantDetailData"
      maxOccurs="unbounded"/>
    <xs:element name="AcademicRecord" type="AcRec:AcademicRecordType"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

These are the first three lines of the XML output file:

```
<!--Sungardhe Treq Articulation Export--><SungardheTreqArticulationExport
xmlns:ns3="urn:org:pesc:core:CoreMain:v1.8.0"
xmlns="urn:com:sungardhe:degreeworks:treq:ArticulationExportEnvelope:v1.0.0">
```

These are the nodes used for each student:

```
<ns3_:CollegeTranscript
xmlns:ns3="urn:org:pesc:message:CollegeTranscript:v1.2.0" xmlns="">
  <Student>
    <Person>
      <ApplicantMasterData> </ApplicantMasterData>
      <ApplicantDetailData> </ApplicantDetailData>
      <AcademicRecord>
        <School> </School>
        <AcademicSummary> </AcademicSummary>
        <Course> </Course>
      </AcademicRecord>
    </Student>
  </ns3_:CollegeTranscript
```

This is the last line of the XML output file:

```
</SungardheTreqArticulationExport>
```

## Person XML

Selected fields from the rad\_primary\_mst will be written to the XML output file, one record per each data element in an XML format as described below.

These standard elements are from the AcademicRecord\_v.1.5.0.xsd "Person" node.

XML Tag	Len	DAPDB	Comments
<SchoolAssignedPersonID>	10	dap_stu_id	This element is the universal ID code assigned to a student at the time of admission. This ID must be valid in the rad_primary_mst.
<FirstName>	180	rad_name (first name after the comma stripped out from the rad_name)	Student's first name.
<MiddleName>	180	rad_name (middle name after the comma and a blank stripped out from the rad_name)	Student's middle name.
<LastName>	180	rad_name (last name before the comma stripped out from the rad_name)	Student's last name.
<CompositeName>	180	rad_name	The student's name in the format: last, first middle.

## Applicant Master XML

Selected fields from the dap\_applicnt\_mst will be written to the XML output file, one record per each data element in an XML format as described below.

These elements are from the AcademicRecord\_v.1.5.0.xsd “Applicant Master Data” node.

```

<!--=====-->
<!--Custom ApplicantMasterData-->
<!--=====-->
<xs:complexType name="ApplicantMasterData">
  <xs:annotation>
    <xs:documentation>Applicant Master Data from the
      dap_applicnt_mst</xs:documentation>
  </xs:annotation>
  <xs:sequence>

      <!-- Custom DegreeWorks Elements from the DAP_APPLICNT_MST
-->

```

All elements from this table are “custom” for Ellucian.

Tag Name	Len	DAPDB	Comments
<ApplicantSchool>	12	dap_school (UCX-STU350)	This element defines the school (a.k.a. campus) to which the student has applied (e.g., UG” for Undergraduate School, “GR” for Graduate School, “LW” for Law School).
<ApplicantDegreeInterest>	2	dap_deg_interest (UCX_STU564)	This element is a code to define the type of degree interest associated with this applicant (e.g., “DS” for Degree Seeking, “ND” for Non-Degree Seeking, etc.).
<ApplicantDegree>	12	dap_degree (UCX-STU307)	This element contains the code that identifies the student's intended degree (e.g., “BS” for Bachelor of Science, “MA” for Master of Arts, “BFA” for Bachelor of Fine Arts).
<ApplicantCatalogYear>	12	dap_catalog_yr (UCX-STU035)	This element defines the catalog year in effect for the student's degree program. The catalog year determines which set of degree requirement definitions should be used when evaluating the student's progress towards completing the degree.
<ApplicantTreqStatus>	2	dap_treq_status (UCX-TRQ062)	The Transfer Equivalency articulation status, should be “AR” for Articulated or “RO” for Rolled Previously.
<ApplicantArticulationDate>	8	dap_artic_date	The Transfer Equivalency articulation date, formatted CCYYMMDD
<ApplicantArticulationTime>	6	dap_artic_time	The Transfer Equivalency articulation time, formatted HHMMSS
<Condition1>	12	dap_condition1	This element stores additional articulation condition as defined by your site.
<Condition2>	12	dap_condition2	This element stores additional articulation condition as defined by your site.
<Condition3>	12	dap_condition3	This element stores additional articulation condition as defined by your site.
<Condition4>	12	dap_condition4	This element stores additional articulation

Tag Name	Len	DAPDB	Comments
			condition as defined by your site.
<Condition5>	12	dap_condition5	This element stores additional articulation condition as defined by your site.
<Condition6>	12	dap_condition6	This element stores additional articulation condition as defined by your site.
<Condition7>	12	dap_condition7	This element stores additional articulation condition as defined by your site.
<Condition8>	12	dap_condition8	This element stores additional articulation condition as defined by your site.
<Condition9>	12	dap_condition9	This element stores additional articulation condition as defined by your site.
<Condition10>	12	dap_condition10	This element stores additional articulation condition as defined by your site.

## Applicant Detail XML

Selected fields from the `dap_appdata_dtl` will be written to the XML output file, one record per each data element in an XML format as described below. This is a repeating table. A student many have several `dap_appdata_dtl` records, one for each Goal Code defined for a student. These elements are from the `AcademicRecord_v.1.5.0.xsd` "Applicant Detail" node.

```

<!--=====-->
<!--Custom ApplicantDetailData-->
<!--=====-->
<xs:complexType name="ApplicantDetailData">
  <xs:annotation>
    <xs:documentation>Applicant Detail Data from the
      dap_appdata_dtl</xs:documentation>
  </xs:annotation>
  <xs:sequence>

      <!-- Custom DegreeWorks Elements from the DAP_APPDATA_DTL -
->

```

All elements from this table are “custom” for Ellucian.

<b>XML Tag</b>	<b>Len</b>	<b>DAPDB</b>	<b>Comments</b>
<GoalCode>	12	dap_goal_code	Must be one of the valid codes: COLLEGE CONC MAJOR MINOR PROGRAM
<GoalCodeDescription>	30		Description of the above valid code used.
<GoalValue>	12	dap_goal_value	If the Goal Code = COLLEGE, the goal value will contain a college code from UCX_STU560. If the Goal Code = CONC, the goal value will contain a concentration from UCX_STU563. If the Goal Code = MAJOR, the goal value will contain a major from UCX_STU023. If the Goal Code = MINOR, the goal value will contain a minor from UCX_STU024. If the Goal Code = PROGRAM, the goal value will contain a program code from UCX_STU316.
<GoalValueDescription>	30	UCX_STU???	The UCX_VALUE description for each Goal Value:  UCX_STU560: COLLEGE UCX_STU563: CONC UCX_STU023: MAJOR UCX_STU024: MINOR UCX_STU316: PROGRAM

## Academic Record - School Type XML

Selected fields from the dap\_college\_dtl will be written to the XML output file, one record per each data element in an XML format as described below. This is a repeating table. A student may have several dap\_college\_dtl records, one for each transfer school attended.

These elements are from the AcademicRecord\_v.1.5.0.xsd "SchoolType" node.

```
<!--=====-->
<!--AcademicSession Types-->
<!--=====-->
<xs:complexType name="SchoolType">
  <xs:sequence>
    <xs:element name="OrganizationName" />
    <xs:group ref="core:OrganizationIDGroup" />
    <xs:element name="LocalOrganizationID" />
    <xs:element name="SchoolOverrideCode" />
    <xs:element name="SchoolLevel" />
    <xs:element name="Contacts" />
    <xs:element name="NoteMessage" />

    <!-- Custom DegreeWorks Elements from the DAP_COLLEGE_DTL -
->
```

All of the elements from this table are "custom" for Ellucian except for the LocalOrganizationIDCode which is part of the standard schema.

Field Name	Len	DAPDB	Comments
<LocalOrganizationIDCode>	10	dap_school_id	This element identifies the transfer institution where this transfer class was taken. It is validated against the rad_ets_mst.
<TransferDegree>	12	dap_tr_degree (UCX-STU379)	This element contains the code that identifies the student's previous degree (e.g., "BA" for Bachelor of Arts, "AA" for Associate of Arts Degree, "MS" for Master of Science, etc).
<TransferConferFlag>	2	dap_confer_flag	This element is a Y/N flag indicating if this institution conferred a degree.
<TransferMajor>	12	dap_tr_major (UCX-STU382)	This element is used to store the major associated with this degree (e.g., "ART" for Art Major, "BUS" for Business Major, "ENGL" for English Major).
<TransferStartDate>	8	dap_tr_start	This element is the date the student began attending this transfer institution. Formatted CCYY-MM-DD.
<TransferStopDate>	8	dap_tr_stop	This element is the date the student stopped attending this transfer institution. Formatted CCYY-MM-DD.
<TransferTranscriptDate>	8	dap_trnscpt_date	This element is the date of the transfer transcript. Formatted CCYY-MM-DD.
<Condition1>	12	dap_condition1	This element stores additional articulation condition as defined by your site.
<Condition2>	12	dap_condition2	This element stores additional articulation condition as defined by your site.
<Condition3>	12	dap_condition3	This element stores additional articulation condition as defined by your site.

<b>Field Name</b>	<b>Len</b>	<b>DAPDB</b>	<b>Comments</b>
<Condition4>	12	dap_condition4	This element stores additional articulation condition as defined by your site.
<Condition5>	12	dap_condition5	This element stores additional articulation condition as defined by your site.
<Condition6>	12	dap_condition6	This element stores additional articulation condition as defined by your site.
<Condition7>	12	dap_condition7	This element stores additional articulation condition as defined by your site.
<Condition8>	12	dap_condition8	This element stores additional articulation condition as defined by your site.
<Condition9>	12	dap_condition9	This element stores additional articulation condition as defined by your site.
<Condition10>	12	dap_condition10	This element stores additional articulation condition as defined by your site.

## Academic Record - Course Type XML

Selected fields from the `dap_transfer_dtl` will be written to the XML output file, one record per each data element in an XML format as described below. This is a repeating table. A student may have several `dap_transfer_dtl` records, one for each transfer class taken.

These elements are from the `AcademicRecord_v.1.5.0.xsd` "Course Type" node. The "standard" Transfer Course items that are extracted by DAP21 are as follows:

```
<xs:complexType name="CourseType">
  <xs:sequence>
    <xs:element name="CourseCreditUnits" />
    <xs:element name="CourseCreditEarned" />
    <xs:element name="CourseAcademicGrade" />
    <xs:element name="CourseRepeatCode" />
    <xs:element name="CourseQualityPointsEarned" />
    <xs:element name="CourseLevel" />
    <xs:element name="CourseSubjectAbbreviation" />
    <xs:element name="CourseNumber" />
    <xs:element name="CourseSectionNumber" />
    <xs:element name="CourseTitle" />
    <xs:element name="CourseBeginDate" />
    <xs:element name="CourseEndDate" />

    <!-- Custom DegreeWorks Elements from the DAP_TRANSFER_DTL -->
  </xs:sequence>
</xs:complexType>
```

Field Name	Len	DAPDB	Comments
<CourseSchool>	12	dap_school (UCX-STU350)	This element is the school code within which the transfer class is to be associated (e.g., "UG" for Undergraduate School, "LW" for Law School, "GR" for Graduate School).
<CourseDegreeInterest>	2	dap_deg_interest (UCX_STU564)	This element is a code to define the type of degree interest associated with this transfer class (e.g., "DS" for Degree Seeking, "ND" for Non-Degree Seeking, etc.)..
<LocalOrganizationIDCode> (standard)	10	dap_school_id	This element identifies the transfer institution where this transfer class was taken. It is validated against the <code>rad_ets_mst</code> .
<TransferSubjectAbbreviation>	12	dap_tr_disc	This element is the discipline code as it was identified at the transfer institution.
<TransferCourseNumber>	12	dap_tr_crse_num	This element is the course number as it was identified at the transfer institution.
<TransferCourseTitle>	30	dap_tr_title	This element stores the title of the course as it was identified at the transfer institution.

Field Name	Len	DAPDB	Comments
<TransferCourseCredits>	7	dap_tr_credits	This element is the number of credits taken at the transfer institution. It is a decimal field in the database, but will only include a decimal if necessary. For example, a >3< will be written for 3.0 credit class while a >2.5< will be written for a 2.5 credit class.
<CourseBeginDate> (standard)	8	dap_tr_start	This element is the start date for a particular transfer class. Format CCYY-MM-DD.
<CourseEndDate> (standard)	8	dap_tr_stop	This element is the stop date for a particular transfer class. Format CCYY-MM-DD.
<CourseCreditUnits> (standard)	2	dap_tr_cr_method (UCX-STU346)	This element defines the calendar structure for the transfer institution's academic year (e.g., "S" = Semester, "Q" = Quarter).
<TransferAcademicGrade>	6	dap_tr_grade (UCX-STU398)	Grade from the transfer institution.
<TransferQualityPointsEarned>	7	dap_tr_grade_pts	This element is the number of grade points awarded at the transfer institution. It is a decimal field in the database, but will only include a decimal if necessary. For example, a >12< will be written for a class with 12.0 quality points while a >10.5< will be written for a class with 10.5 quality points.
<TransferArticulation>	4	dap_articulation (UCX-TRQ065)	The articulation code indicating one-to-one, one-to-many, many-to-one, and many-to-many. Assigned by Transfer Equivalency (DAP12).
<CourseSubjectAbbreviation> (standard)	12	dap_discipline	The discipline code of the resident course articulated by Transfer Equivalency.
<CourseNumber> (standard)	12	dap_course_num	The course number of the resident course articulated by Transfer Equivalency.
<CourseSection> (standard)	2	dap_section	The course section of the resident course articulated by Transfer Equivalency.
<CourseTitle> (standard)	30	dap_course_title	The course title of the resident course articulated by Transfer Equivalency.
<CourseDepartment>	12	dap_dept (UCX-STU362)	The academic department code of the resident course articulated by Transfer Equivalency.
<CourseDivision>	12	dap_division (UCX-STU351)	The academic division code of the resident course articulated by Transfer Equivalency.

Field Name	Len	DAPDB	Comments
<CourseCreditEarned> (standard)	7	dap_cr_earn	The credits earned at your institution as articulated by Transfer Equivalency. It is a decimal field in the database, but will only include a decimal if necessary. For example, a >3< will be written for a class worth 3.0 credits while a >2.5< will be written for a class worth 2.5 credits.
<CourseCreditType>	2	dap_credit_type (UCX-STU355)	The type of credit, e.g. academic, granted for the course articulated by Transfer Equivalency.
<CourseGradeType>	2	dap_grade_type (UCX-STU356)	The type of grade, e.g. A-F or Pass-Fail, granted for the course articulated by Transfer Equivalency.
<CourseAcademicGrade> (standard)	6	dap_final_grade (UCX-STU385)	The grade granted for the course articulated by Transfer Equivalency.
<CourseQualityPointsEarned> (standard)	7	dap_grade_points	The grade points granted for the course articulated by Transfer Equivalency. It must be numeric. Calculated by Transfer Equivalency based on FINAL-GRADE and CR-EARN.
<CourseClassStatus>	2	dap_class_status (UCX-STU380)	The class status code for the course articulated by Transfer Equivalency, "A " = Add, "RP" = first instance of repeated course, "RT" = subsequent instances of repeated course.
<CourseRepeatSubjectAbbreviation>	12	dap_repeat_disc	The course discipline of the resident course repeated by the resident course articulated by Transfer Equivalency. For example, if MATH 115 is articulated but it is a repeat of PHIL 111 then repeat_disc would be PHIL. This field is not calculated by Transfer Equivalency and will only be filled in if it was passed from the student system by some external process.
<CourseRepeatNumber>	12	dap_repeat_num	The course number of the resident course repeated by the resident course articulated by Transfer Equivalency. For example, if MATH 115 is articulated but it is a repeat of PHIL 111 then repeat_num would be 111. This field is not calculated by Transfer Equivalency and will only be filled in if it was passed from the student system by some external process.

Field Name	Len	DAPDB	Comments
<CourseRepeatPolicy>	2	dap_repeat_plcy (UCX-AUD047)	The repeat policy governing how Degree Works will apply the repeated class. Valid repeat policies are documented in UCX–AUD047. This field is not calculated by Transfer Equivalency and will only be filled in if it was passed from the student system by some external process.
<CourseAcadVotech>	6	dap_acad_votech (UCX-XXX376)	The academic-vocational code of the resident course articulated by Transfer Equivalency.
<CourseClassType>	2	dap_class_type (UCX-XXX377)	The class type code of the resident course articulated by Transfer Equivalency.
<TransferArticulationDate>	8	dap_artic_date	The Transfer Equivalency articulation date, formatted CCYYMMDD
<TransferArticulationStatus>	2	dap_artic_status	The Transfer Equivalency articulation status.
<Authorizer>	30	dap_authorizer	The name of the person who authorized the articulation of this transfer course.
<UserDefined1>	12	dap_user_def1	This element stores additional data as defined by your site.
<UserDefined2>	12	dap_user_def2	This element stores additional data as defined by your site.
<UserDefined3>	12	dap_user_def3	This element stores additional data as defined by your site.
<UserDefined4>	12	dap_user_def4	This element stores additional data as defined by your site.
<UserDefined5>	12	dap_user_def5	This element stores additional data as defined by your site.
<UserDefined6>	12	dap_user_def6	This element stores additional data as defined by your site.
<UserDefined7>	12	dap_user_def7	This element stores additional data as defined by your site.
<UserDefined8>	12	dap_user_def8	This element stores additional data as defined by your site.
<UserDefined9>	12	dap_user_def9	This element stores additional data as defined by your site.
<UserDefined10>	12	dap_user_def10	This element stores additional data as defined by your site.
<Condition1>	12	dap_condition1	This element stores additional data as defined by your site that can be used by Transfer Equivalency as a condition to be evaluated during transfer articulation.
<Condition2>	12	dap_condition2	This element stores additional data as defined by your site that can be used by Transfer Equivalency as a condition to be evaluated during transfer articulation.

Field Name	Len	DAPDB	Comments
<Condition3>	12	dap_condition3	This element stores additional data as defined by your site that can be used by Transfer Equivalency as a condition to be evaluated during transfer articulation.
<Condition4>	12	dap_condition4	This element stores additional data as defined by your site that can be used by Transfer Equivalency as a condition to be evaluated during transfer articulation.
<CourseMapId>	8	dap_map_id	The identifier of the mapping rule used during Transfer Equivalency articulation. Filled in by Transfer Equivalency.
<CourseStudentSequence>	4	dap_tr_stu_seq	Sequence number to uniquely identify this transfer course for this student. It is an integer field in the database. For example, if the sequence number is 4 a >4< will be written to the XML file. There will be no leading zeroes.
<CourseTerm>	12	trm_sort (UCX-STU016)	This element stores the term associated with the transfer class (e.g., "201120" for Spring 2011) .

## Example Output

The data file output by DAP21 is an XML file. The data file contains records described above and resides in the \$ADMIN\_HOME/data sub-directory of the test or production directory. The output XML data file contains the Transfer Equivalency Articulation data for all students selected.

DAP21 sorts data according to ID and School-ID.

Sample records in the DAP21 output XML file for one student:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sungardhe Treq Articulation Export--><SungardheTreqArticulationExport
xmlns:ns3="urn:org:pesc:core:CoreMain:v1.8.0"
xmlns="urn:com:sungardhe:degreeworks:treq:ArticulationExportEnvelope:v1.0.0">
  <ns3:CollegeTranscript
xmlns:ns3_="urn:org:pesc:message:CollegeTranscript:v1.2.0" xmlns="">
    <Student>
      <Person>
        <SchoolAssignedPersonID>N00010961</SchoolAssignedPersonID>
        <Name>
          <FirstName>Kathleen</FirstName>
          <LastName>Weber</LastName>
          <CompositeName>Weber, Kathleen</LastName>
        </Name>
      </Person>
      <ApplicantMasterData>
        <ApplicantSchool>U</ApplicantSchool>
        <ApplicantDegree>BA</ApplicantDegree>
        <ApplicantCatalogYear>1978</ApplicantCatalogYear>
        <ApplicantTreqStatus>R0</ApplicantTreqStatus>
      </ApplicantMasterData>
    </Student>
  </ns3:CollegeTranscript>
</SungardheTreqArticulationExport>
```

```

<ApplicantArticulationDate>2011-01-19</ApplicantArticulationDate>
<ApplicantArticulationTime>131020</ApplicantArticulationTime>
<Condition1>testcond1</Condition1>
<Condition2>testcond2</Condition2>
<Condition3>testcond3</Condition3>
<Condition4>testcond4</Condition4>
<Condition5>testcond5</Condition5>
<Condition6>testcond6</Condition6>
<Condition7>testcond7</Condition7>
<Condition8>testcond8</Condition8>
<Condition9>testcond9</Condition9>
<Condition10>testcond10</Condition10>
</ApplicantMasterData>
<ApplicantDetailData>
  <GoalCode>MAJOR      </GoalCode>
  <GoalCodeDescription>Major</GoalCodeDescription>
  <GoalValue>ANSC      </GoalValue>
  <GoalValueDescription>Animal Science</GoalValueDescription>
</ApplicantDetailData>
<ApplicantDetailData>
  <GoalCode>MINOR</GoalCode>
  <GoalCodeDescription>Minor</GoalCodeDescription>
  <GoalValue>ART-MINOR  </GoalValue>
  <GoalValueDescription>Art Minor</GoalValueDescription>
</ApplicantDetailData>
<ApplicantDetailData>
  <GoalCode>COLLEGE</GoalCode>
  <GoalCodeDescription>College</GoalCodeDescription>
  <GoalValue>00</GoalValue>
  <GoalValueDescription>No College Designated</GoalValueDescription>
</ApplicantDetailData>
<ApplicantDetailData>
  <GoalCode>MINOR</GoalCode>
  <GoalCodeDescription>Minor</GoalCodeDescription>
  <GoalValue>ARTHIST-MN</GoalValue>
  <GoalValueDescription>Art History Minor</GoalValueDescription>
</ApplicantDetailData>
<AcademicRecord>
  <School>
    <OrganizationName>Generic Community College</OrganizationName>
    <LocalOrganizationID>
      <LocalOrganizationIDCode>1111</LocalOrganizationIDCode>
    </LocalOrganizationID>
    <TransferDegree>AA</TransferDegree>
    <TransferConferFlag>N </TransferConferFlag>
    <TransferMajor>MATH</TransferMajor>
    <TransferStartDate>2008-09-01</TransferStartDate>
    <TransferStopDate>2008-12-15</TransferStopDate>
    <TransferTranscriptDate>2008-12-20</TransferTranscriptDate>
    <Condition1>testcond1</Condition1>
    <Condition2>testcond2</Condition2>
    <Condition3>testcond3</Condition3>
    <Condition4>testcond4</Condition4>
    <Condition5>testcond5</Condition5>
    <Condition6>testcond6</Condition6>
    <Condition7>testcond7</Condition7>
    <Condition8>testcond8</Condition8>
    <Condition9>testcond9</Condition9>
  </School>
</AcademicRecord>

```

```

    <Condition10>testcond10</Condition10>
  </School>
  <AcademicSummary>
    <AcademicSummaryType>TransferOnly</AcademicSummaryType>
  </AcademicSummary>
  <Course>
    <CourseCreditValue>3</CourseCreditValue>
    <CourseCreditEarned>3</CourseCreditEarned>
    <CourseAcademicGrade>A</CourseAcademicGrade>
    <CourseQualityPointsEarned>12</CourseQualityPointsEarned>
    <CourseSubjectAbbreviation>ENGELECT</CourseSubjectAbbreviation>
    <CourseNumber>150</CourseNumber>
    <CourseSectionNumber>X</CourseSectionNumber>
    <CourseTitle>19th Century American Lit</CourseTitle>
    <CourseSchool>U</CourseSchool>
    <CourseDepartment>ENGL</CourseDepartment>
    <CourseDivision>AS</CourseDivision>
    <CourseCreditType>TR</CourseCreditType>
    <CourseGradeType>AF</CourseGradeType>

  <CourseRepeatSubjectAbbreviation>ENGL</CourseRepeatSubjectAbbreviation>
  <CourseRepeatCourseNumber>144</CourseRepeatCourseNumber>
  <CourseRepeatPolicy>1 </CourseRepeatPolicy>
  <CourseAcadVotech>AC</CourseAcadVotech>
  <CourseClassType>CL</CourseClassType>
  <TransferSubjectAbbreviation>ENGL</TransferSubjectAbbreviation>
  <TransferCourseNumber>130</TransferCourseNumber>
  <TransferCourseTitle>American Literature of the
19t</TransferCourseTitle>
  <TransferAcademicGrade>A-</TransferAcademicGrade>
  <TransferQualityPointsEarned>10.5</TransferQualityPointsEarned>
  <TransferArticulation>1T01</TransferArticulation>
  <TransferArticulationDate>2011-05-15</TransferArticulationDate>
  <Authorizer>DRP</Authorizer>
  <UserDefined1>testundef1</UserDefined1>
  <UserDefined2>testundef2</UserDefined2>
  <UserDefined3>testundef3</UserDefined3>
  <UserDefined4>testundef4</UserDefined4>
  <UserDefined5>testundef5</UserDefined5>
  <UserDefined6>testundef6</UserDefined6>
  <UserDefined7>testundef7</UserDefined7>
  <UserDefined8>testundef8</UserDefined8>
  <UserDefined9>testundef9</UserDefined9>
  <UserDefined10>testundef10</UserDefined10>
  <Condition1>testcond1</Condition1>
  <Condition2>testcond2</Condition2>
  <Condition3>testcond3</Condition3>
  <Condition4>testcond4</Condition4>
  <CourseMapId>MA147444</CourseMapId>
  <CourseTerm>200820</CourseTerm>
</Course>
<Course>
  <CourseCreditUnits>S</CourseCreditUnits>
  <CourseCreditValue>3</CourseCreditValue>
  <CourseCreditEarned>3</CourseCreditEarned>
  <CourseAcademicGrade>B</CourseAcademicGrade>
  <CourseQualityPointsEarned>9</CourseQualityPointsEarned>
  <CourseSubjectAbbreviation>MATH</CourseSubjectAbbreviation>

```

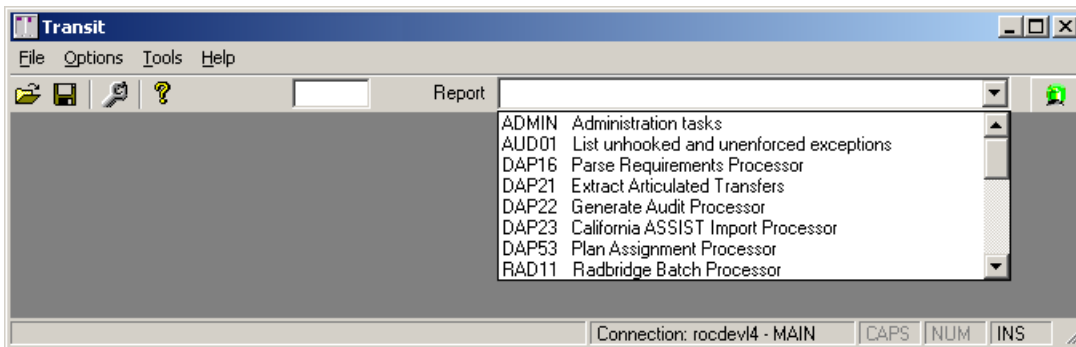
```

<CourseNumber>121</CourseNumber>
<CourseTitle>Algebra I</CourseTitle>
<CourseSchool>UG</CourseSchool>
<CourseCreditType>TR</CourseCreditType>
<CourseGradeType>AF</CourseGradeType>
<TransferSubjectAbbreviation>MTH</TransferSubjectAbbreviation>
<TransferCourseNumber>110</TransferCourseNumber>
<TransferCourseTitle>College Algebra</TransferCourseTitle>
<TransferAcademicGrade>B</TransferAcademicGrade>
<TransferQualityPointsEarned>9</TransferQualityPointsEarned>
<TransferArticulation>1T01</TransferArticulation>
<TransferArticulationDate>2011-01-19</TransferArticulationDate>
<CourseMapId>MA147382</CourseMapId>
<CourseTerm>200820</CourseTerm>
</Course>
</AcademicRecord>
</Student>
</ns3_:CollegeTranscript>
</SungardheTreqArticulationExport>

```

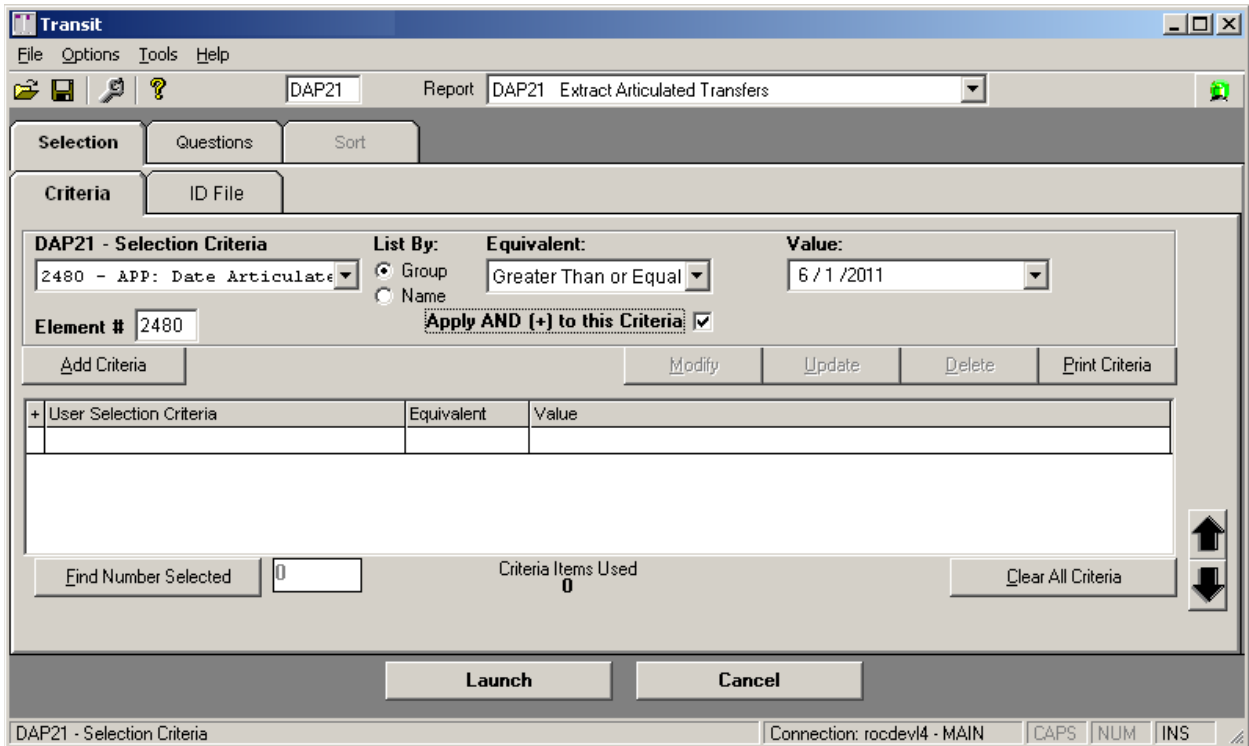
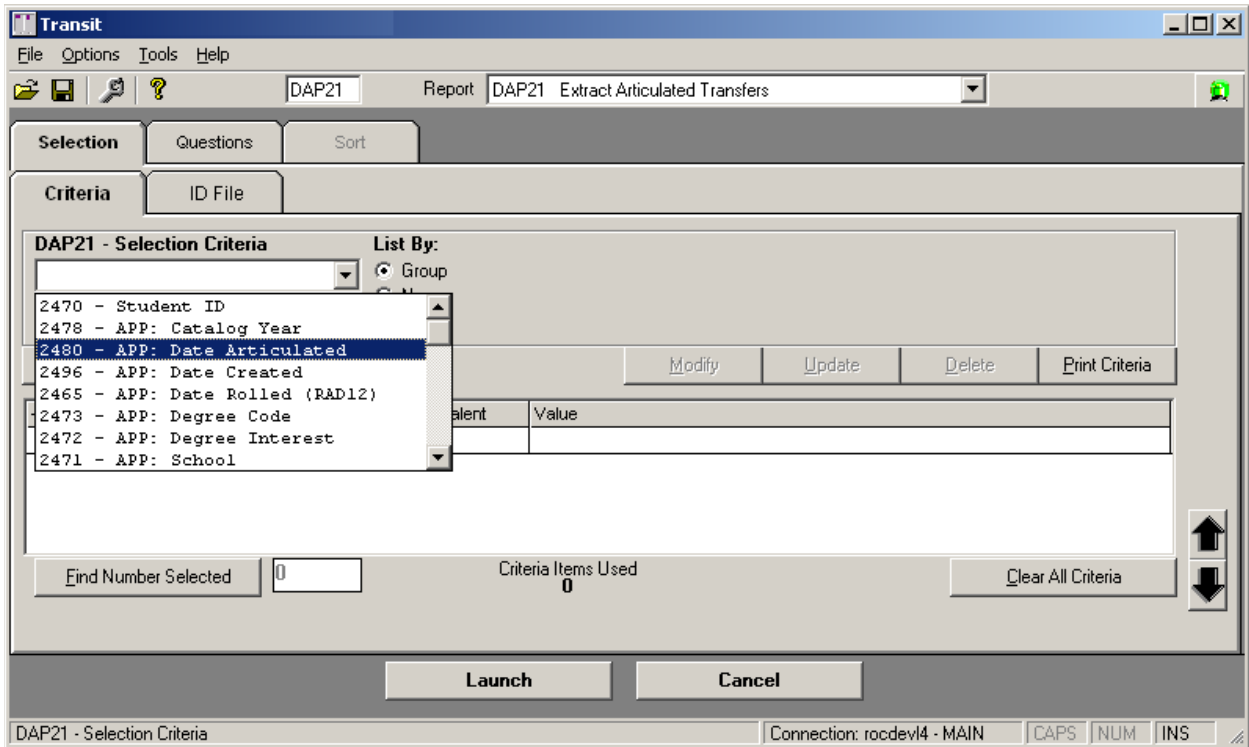
## Running the DAP21 Processor

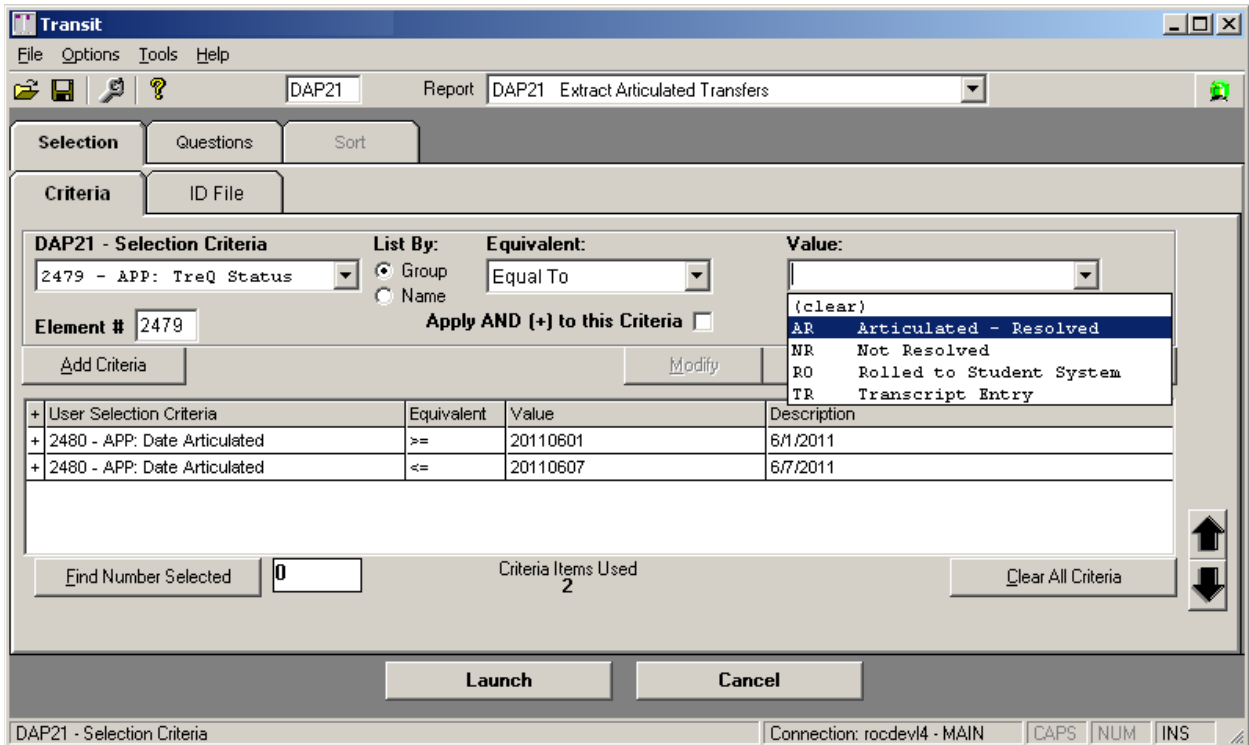
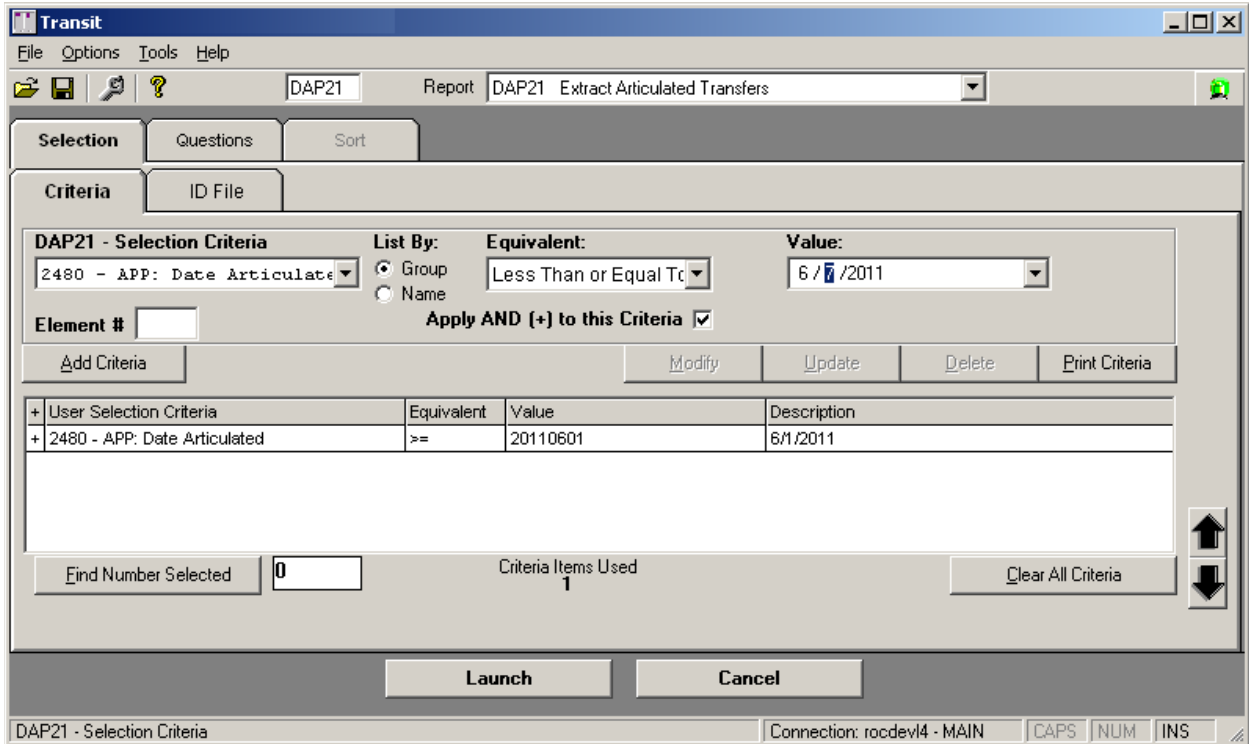
DAP21, “Extract Articulated Transfers”, can be executed from Transit:



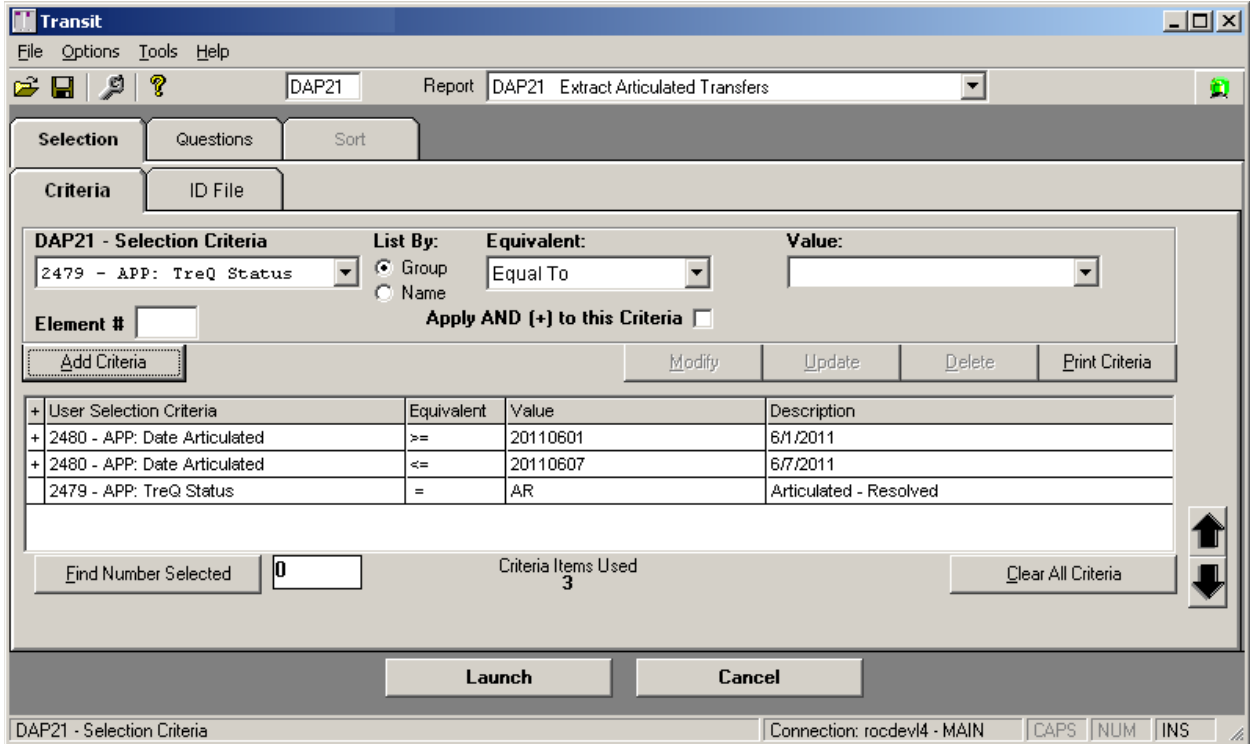
Selection criteria must be used to select the appropriate pool of students to be extracted from Transfer Equivalency for input into your student system. Click on the drop-down box below the DAP21 – Selection Criteria label to see a list of available database columns to choose from.

The example that follows selects a range of Articulation Dates (E#2480) and the Treq Status (E#2479) of “AR” (Articulated) to define the population to be extracted. Note the plus “+” signs to the left of the User Selection Criteria – make sure to check that box “Apply AND (+) to this criteria” when adding date ranges:

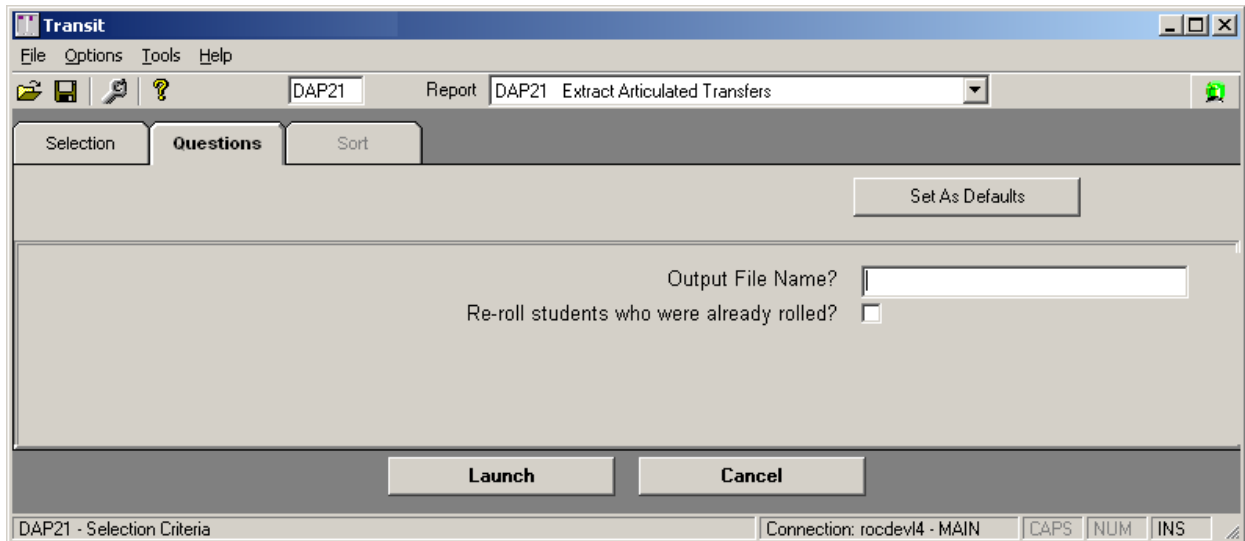




The selection criteria detailed above translates to “select all students with a dap\_applicant\_mst.dap\_artic\_date >= ‘20110601’ AND dap\_applicant\_mst.dap\_artic\_date <= ‘20110607’”. Without the use of the plus “+” sign ALL students with a dap\_applicant\_mst would be selected because the translation would be “select all students with a dap\_applicant\_mst.dap\_artic\_date >= ‘20110601’ OR dap\_applicant\_mst.dap\_artic\_date <= ‘20110607’”. The use of the ‘OR’ would cause ALL records to be selected. Make sure to use check the “Apply AND (+) to this Criteria” box when selecting ranges to make sure the correct pool of students is extracted.



Normally only students whose transfer data has been Articulated (dap\_applicnt\_mst.dap\_treq\_status = 'AR') should be selected for extract by DAP21. Thus, it is best to include this selection criteria (E#2479) when inputting your selection criterion. However, if students whose data has been previously rolled (dap\_applicnt\_mst.dap\_treq\_status = 'RO') are desired then check the 'Re-roll students who were already rolled?' box displayed below the Output File Name question on the next screen:



If no "Output File Name" is specified above then a default Output File Name of dap21nnn.xml (where nnnn is the last 4-digits of the system process ID number or PID) will be created in the \$ADMIN\_HOME/data directory. If desired, specify your own Output File Name (maximum 8-bytes only, no special characters and no embedded blanks). For example, the Output File Name might be 'MyFile', 'myfile2', 'TreqOut' or 'TestFile'. The real output file will be created with an '.xml' extension (e.g., MyFile.xml).

### Email Message Capability

The DAP21 job may also send email notification that includes the contents of the Action File if the "DAP21\_EMAIL\_ADDRESS" environment variable has been set in dwenv.config file for the appropriate staff member(s). For example:

```
export DAP21_EMAIL_ADDRESS=some.staff@sungardhe.com
```

When DAP21 finishes an email message is sent to the appropriate individuals if this configuration has been set appropriately.

### Errors, Warnings and Success

DAP21 does not validate data so it does not give errors about specific records. However, it does filter records based on the Treq Roll Status from the dap\_applicnt\_mst. Edits:

- If the Parameter Re-roll Student flag = 'N' and the dap\_treq\_status = 'RO' (already rolled) the student will be *Skipped* and NO records will be written to the output file.
- If the Parameter Re-roll Student flag = 'Y' and the dap\_treq\_status is NOT = 'RO' (already rolled) and NOT = 'AR' (Articulation Resolved) the student will be *Skipped* and NO records will be written to the output file.

## Warnings

DAP21 does not emit very many warnings. The ID code of the students skipped along with one of the following two messages will be displayed on the DAP21 execution log for the student ID (dap21nnnnL.log file in the \$ADMIN\_HOME/dgwsPOOL directory):

```
Skipping Student ID [N00010961] because TreQ data has already been rolled!
Skipping Student ID [N00010961] because the TreqStatus shows the data has
not been articulated!
```

## Fatal Errors

DAP21 will fail if the output file cannot be created, if the Parameter record cannot be found or read successfully, if the DAP database cannot be opened or if one of the Transfer Equivalency tables being used (rad\_primary\_mst, dap\_applicant\_mst, dap\_appdata\_dtl, dap\_college\_dtl or dap\_transfer\_dtl) cannot be read.

## Success

**DAP21 writes the number of records considered, output and skipped to the dap21nnnnA.act Action File:**

```
===== DAP21 ACTION FILE =====
                                STUDENTS CONSIDERED:  3
                                STUDENTS OUTPUT:      2
                                STUDENTS SKIPPED:      1
                                NUMBER OF LINES WRITTEN TO XML FILE:  485

Output XML File = /home/dpayne/WorkSpaces/main/admin/data/DRP21OUT.xml

DAP21JOB Started Wed Jun 15 16:40:52 EDT 2011
DAP21JOB Ended   Wed Jun 15 16:41:00 EDT 2011

Time elapsed to process students:          0 hours, 0 minutes, 12 seconds.
```

Note that the XML output file will be created in the \$ADMIN\_HOME/data directory.

**DAP21 writes detailed information from the “ArticulationExport” script to the dap21nnnnL.log Log File:**

```
DAP21JOB Started Wed Jun 15 16:32:44 EDT 2011
Working directory = /home/dpayne/WorkSpaces/main/admin/logdebug/dap21drp
DegreeWorks Release = DW4.0.8
-rw-rw---- 1 dpayne users 405 Jun 15 16:32 DAP21M01
ArticulationExport === Input Parameters ===
DAP21A    User, Re12345678                280950021 DAP

DAP21P0001DRP21OUT                N
DAP21S00012470 =8080                Student ID                001
01
DAP21S00022470 =SRI                Student ID                001
```

```

01
DAP21S00032470 =N00010961 Student ID                                001
01

ArticulationExport === End Parameters ===
ArticulationExport === Running utl39 to select ID Codes ===
Running UTL39 to select IDs to an ID file
The UTL39_IDFILE variable is not set - must run utl39
ArticulationExport === Begin UTL39 === Wed Jun 15 16:32:44 EDT 2011

UTL39: 3 students were selected
ArticulationExport === End UTL39 === Wed Jun 15 16:32:44 EDT 2011
-rwxrwx--- 1 dpayne users 33 Jun 15 16:32 utl68.ids
Our dap21.students file of IDs:
-rwxrwx--- 1 dpayne users 33 Jun 15 16:32 dap21.students
ArticulationExport === Setting java_classpath ===
ArticulationExport === Begin Articulation Export === Wed Jun 15 16:32:44 EDT
2011

DAP21 Parameters =
/home/dpayne/WorkSpaces/main/admin/logdebug/dap21drp/DAP21M01

ArticulationExport === ExportExecutor Parameters ===
-----
Input ID File      =
/home/dpayne/WorkSpaces/main/admin/logdebug/dap21drp/dap21.students
Output XML File   = /home/dpayne/WorkSpaces/main/admin/data/DRP21OUT.xml
Reroll Student    = N
-----

ArticulationExport === Running ExportExecutor to create XML Output File ===
JAVA:06/15/11 16:32:45 Version and build number of this jar package is
4.0.MAIN-SNAPSHOT.1206
JAVA:06/15/11 16:32:45 Initializing jndi contex
JAVA:06/15/11 16:32:45 Found datasources list
JAVA:06/15/11 16:32:45 Creating a datasource for dw, jdbc.properties should
cont
ain values for dw.jdbc.driverClassName, dw.jdbc.url, dw.jdbc.username,
dw.jdbc.password and dw.jdbc.jndiname
JAVA:06/15/11 16:32:45 Creating a datasource for ops, jdbc.properties should
contain values for ops.jdbc.driverClassName, ops.jdbc.url, ops.jdbc.username,
ops.jdbc.password and ops.jdbc.jndiname
JAVA:06/15/11 16:32:45 Initializing jndi contex
JAVA:06/15/11 16:32:45 Found datasources list
JAVA:06/15/11 16:32:45 Creating a datasource for dw, jdbc.properties should
contain values for dw.jdbc.driverClassName, dw.jdbc.url, dw.jdbc.username,
dw.jdbc.password and dw.jdbc.jndiname
JAVA:06/15/11 16:32:45 Creating a datasource for ops, jdbc.properties should
contain values for ops.jdbc.driverClassName, ops.jdbc.url, ops.jdbc.username,
ops.jdbc.password and ops.jdbc.jndiname
JAVA:06/15/11 16:32:45 Initializing Spring
JAVA:06/15/11 16:32:54 Finished initializing Spring
JAVA:06/15/11 16:32:54 STUDENTS CONSIDERED: 3
JAVA:06/16/11 14:42:55 ***** <MESSAGES> *****
JAVA:06/16/11 14:43:07 Exporting Student ID [8080]
JAVA:06/15/11 16:32:55 Exporting Student ID [N00010961]
JAVA:06/15/11 16:32:56 Skipping Student ID [SRI] because TreQ data has already
been rolled!

```

```
JAVA:06/16/11 14:42:55 ***** <END MESSAGES> *****
JAVA:06/15/11 16:32:56 0 hours 0 minutes 11 seconds
ArticulationExport === Completed ExportExecutor for the selected students ===
ArticulationExport === XML Output File: ===
-rw-rw---- 1 dpayne users 270 Jun 15 16:32
/home/dpayne/WorkSpaces/main/admin/data/DRP21OUT.xml
ArticulationExport === Total Counts: ===
                        STUDENTS OUTPUT: 2
                        NUMBER OF LINES WRITTEN TO XML FILE: 485
ArticulationExport === End ExportExecutor === Wed Jun 15 16:32:56 EDT 2011
ArticulationExport === Cleanup ===
Time elapsed to process students:          0 hours, 0 minutes, 12 seconds.
DAP21JOB Ended Wed Jun 15 16:32:56 EDT 2011
```

## DAP22 - Batch Audits

DAP22, the Degree Works batch audit processor, can be launched through PC Transit. Students accessing Degree Works from the web may not have the capability to run new audits. Instead, they view the most recent audit stored for them in the Degree Works database. For this reason the Degree Works database must contain an audit for every active student. Registrar and Advisors typically can run new audits, but these audits are performed for one student at a time. The DAP22 processor will run audits in batch, which means it will produce an audit for every student, for every active student or for a group of students selected by some criteria. However, it is rare you will need to do this from Transit since batch audits are run after RAD11 completes – meaning you should have updated audits for your students as their data changes.

If requested, Transit will also produce a printed audit report for each student audited. The reports are generated using FOP – which allows us to convert the XML audit trees to PDF or PCL (printing).

The PC Transit documentation covers the selection criteria and the steps to use when doing student searches and for also creating ID files so that you may audit a small, selected group of students.

You can run simultaneous Batch Audit processes to take advantage of the multiple processors you have on your machine. You may want to run a batch on seniors, another on juniors, another on sophomores and yet another on first-year students. Running batch audit processes like this in parallel should result in better throughput if you have more than one CPU on your system.

You can set the DGWCPUCOUNT variable in dwenv.config to tell DAP22 how many processors you have on your system. The DAP22 script that is launched from Transit and after RAD11 completes will run multiple DAP22 processes in parallel against the list of student IDs selected. Running a DAP22 for each CPU available should result in better throughput but doing this will consume more resources than running a single DAP22. If you select “Produce printed audit report” in Transit the DAP22 job will only run one dap22 process regardless of the DGWCPUCOUNT value.

When running new audits, creating output or building CPA results records be sure you chose the correct audit type on the questions tab. Normally you will want this set to AA (Academic Audit) but if you wish to generate Financial Aid Audits you can set the audit type to FA. You must specify an Aid Term when running a Financial Aid audit however.

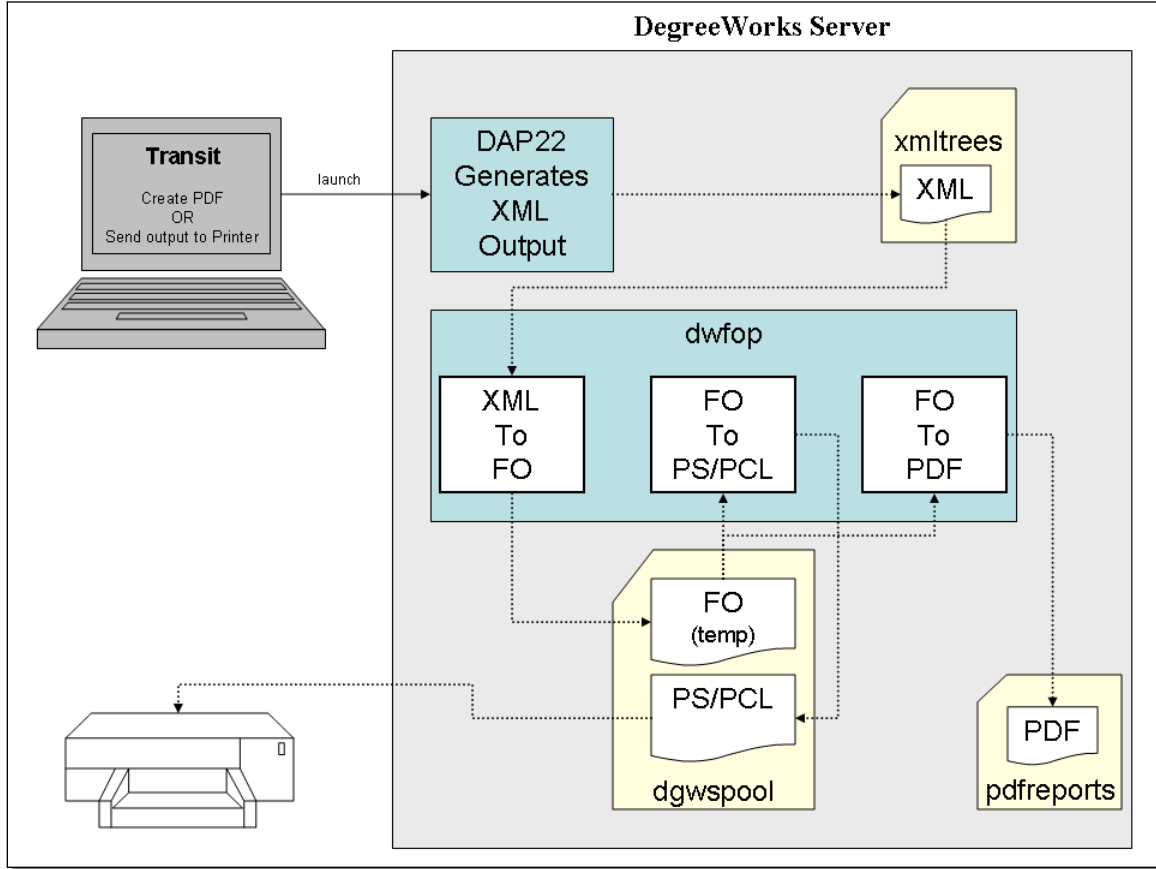
## Report Formats

Only UCX-RPT036 RPTxx reports will appear in the drop-down list for DAP22. Ellucian supplies a suite of RPT reports and you are welcome to create your own. Ellucian suggests that you use report names RPT90-RPT99 for any reports you create in UCX-RPT036. Once you have added additional entries in Surecode you can run ucx12job (or simply wait until it runs the next morning) in order for your reports to appear in Transit.

Select the type of output you want by choosing from the drop-down list for the “*What type of output do you want?*” question. Depending on the report you chose, you can either created a PDF file of audits or you may print the audits directly – or simply create a raw XML file that you may process with your own tools later. See the section below on FOP to see how printed and PDF reports are generated. The Sort Tab allows you to control the order of the audits in the output. Sorting by student name or major are just two of the options.

## FOP - Formatting Objects Processor

FOP is used in Degree Works to convert the XML audit trees into either PDF or a PCL file for printing. It is an open-source tool available at no cost from the Apache Software Foundation. The following diagram shows the process flow when running the DAP22 processor and generating output utilizing FOP. This output can be in the form of a printed report or, if you are processing multiple student audit requests at one time, a single large PDF file with every student's audit in it.



You can learn technical and operational details about FOP from Apache:

<http://xmlgraphics.apache.org/fop/>

Ellucian has developed an XSL stylesheet to transform the XML to fop commands. Each institution will want to localize the XSL or create one or more of their own. The Ellucian-delivered FOP stylesheets are located in **\$ADMIN\_HOME/xsl**. You should copy these files to **\$LOCAL\_HOME/xsl** and make changes to them there. Any newly created stylesheets should also be placed in the **\$LOCAL\_HOME/xsl** directory.

Each of the UCX-RPT036 reports points to the XSL that is to be used. Ellucian delivers a suite of RPT reports and each institution is welcome to create their own – but new reports should be created using names RPT90-RPT99. For any of the Ellucian delivered reports or for new ones created you must specify the name of the stylesheet to be used in Surecode’s UCX-RPT036 screen, as shown in the following example:

**RPT036** → Show All Tables

### RPT036: Audit Report Formats

Key	Title	XSL Stylesheet	Show Block Remarks	Block Qualifiers	Block
RPT30	Registrar Report	fopaudits.xsl	Y	Y	N
RPT31	Student View	fopaudits.xsl	N	N	N
RPT33	Graduation Checklist	fopaudits.xsl	N	N	N
RPT50	Financial Aid Report	fopaudits-aid.xsl	N	N	N
RPT55	Athletic Eligibility Report	fopaudits-athl.xsl	N	N	N
SEP30	Planner Worksheet	AuditSEP.xsl	Y	N	N
SEP31	New Planner Worksheet	DGW_Report.xsl	Y	N	N
TRF31	Transfer Audit	TransferAudit.xsl	Y	N	Y

**Edit mode: RPT31**

**Instructions**

The XSL Stylesheet is required for web reports and those that are to be run through Transit to create a PDF or be printed.

For Transit reports the XSL and any graphics must exist in the admin/common directory.

**Record Details**

Revision

Who

Timestamp

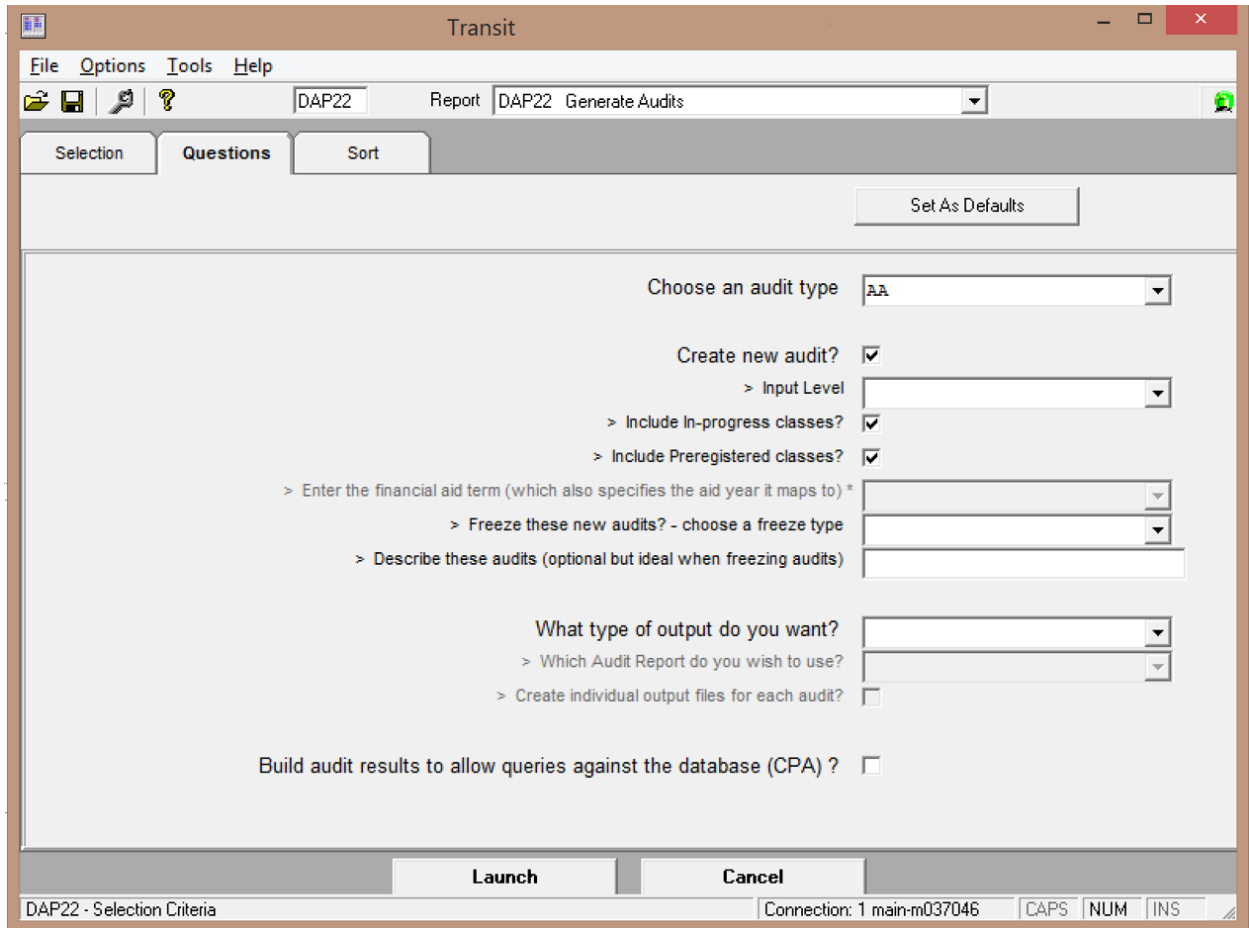
Other flags in UCX-RPT036 are used by the stylesheet to control the output – hiding or showing certain items in the report. Also note that the newer “FOP XSL Stylesheet” is ignored for Transit reports; this newer field is only used by the Web reports when creating PDF worksheets.

## Running the report

In Transit you can specify the type of output desired:

<b>PDF</b>	Create PDF file	<p>When <i>PDF</i> is specified a .pdf file will be created using the stylesheet specified in UCX-RPT036. The .pdf file will be placed in the admin/<b>pdfreports</b> directory. This PDF file can then be downloaded, viewed, and printed as needed. The dap22done script (found in local/scripts) can be modified to move the files from the pdfreports directory to the final destination.</p> <p>When the “Create individual output” option is checked, the files will be placed in the admin/pdfreports/indivNNNN directory, where NNN is the job number returned in Transit. Each PDF file is named student_&lt;id&gt;_&lt;school&gt;_&lt;degree&gt;.pdf</p> <p>example: student_123456_UG_BS.pdf</p>
<b>Printed</b>	Print audit reports	<p>When <i>Printed</i> is specified a .pcl file will be created using the stylesheet specified in UCX-RPT036 and the system’s print commands will be used to send the file to the printer device specified when Launch is clicked.</p> <p>Printers need to be either a PostScript 3.0 printer or a PCL printer (with ghostscript installed)</p>
<b>XML</b>	Create raw xml	<p>As the Surecode screen reminds you, the XSL specified is ignored if you want only the raw XML. The XML file created will be placed in the admin/<b>xmltrees</b> directory. You may use this raw XML as input to one of your own tools as needed.</p> <p>When the “Create individual output” option is checked, the files will be placed in the admin/xmltrees/indivNNNN directory, where NNN is the job number returned in Transit. Each XML file is named student_&lt;id&gt;_&lt;school&gt;_&lt;degree&gt;.xml</p> <p>example: student_123456_UG_BS.xml</p>

You must also select an audit report if PDF or Printed is selected – this tells Degree Works what stylesheet is needed in addition to other UCX-RPT036 configuration options.



## dwfop

The dwfop script is used by Degree Works to take in an XML file and create a PDF file or printed output.

### Creating PDF:

When creating a PDF file from XML you need to supply the XML filename, "PDF" as the 2nd parameter, the XSL filename and the PDF filename.

Format: dwfop xml-filename pdf xsl-filename pdf-filename  
Example: dwfop myfile.xml pdf some.xsl myfile.pdf

### Printing:

When printing a file from XML you need to supply the XML filename, "PRINT" as the 2nd parameter, the XSL filename, the printer and the number of copies.

Format: dwfop xml-filename print xsl-filename device num-copies  
Example: dwfop myfile.xml print someother.xsl 0952 1

**Special notes:**

The xml file is looked for in xmltrees if the filename given is not found in current directory

The xsl file is looked for in local/xsl if the filename given is not found in current directory

Images referenced in the xsl should be housed in local/images

The pdf file name must be absolute (basename or w/ directory specified)

The fo file is created in the local/tmp directory and saved if debugging is on (DWDEBUG=1); otherwise it is deleted.

The printer device must be specified if printing – the number of copies is set to 1 if not specified.

The printer device number must be setup in UCX-SYS910. UCX12JOB will output this information to the common/UCX12M06 file.

**Important Note** – there is a 2GB limit to any file created on the system. Because of this limit you need to be careful of the number of students you run in any one particular batch. Each audit's xml can be 50K, 100K or more thus limiting the number of students that can be commingled into a single XML file. Whether you desire PDF output, printed output or the raw XML file, an XML document is created along the way containing an audit for each of the students in your pool. You may have to split up your pool of students into multiple runs to get the output you want.

## DAP27 – What-if Audits

You can run batch what-if audits from Transit on a set of selected students. You may want to do this to determine if some students have completed a certificate or have completed another program.

The screenshot shows the 'Questions' tab in the Transit software. The window title is 'DAP27 What-if Audits'. The interface includes a 'Set As Defaults' button and several sections of configuration options:

- Please select each of these required items:**
  - Select a Academic Year \*: 2013
  - Select a Level \*: UG
  - Select a Degree \*: BUS\_CERT
- You may choose one of the following of each also:**
  - Select a Program
  - Select a College
  - Select a Major: ACCT
  - Select a Minor
  - Select a Concentration
- You can freeze the audits and/or print/PDF the audits.**
  - If you do neither you may not be able to review them later.**
  - Freeze these what-if audits? - choose a freeze type: WHATIF
  - Describe these what-if audits: Business Certificate
  - What type of output do you want?
  - > Which Audit Report do you wish to use?
  - > Create individual output files for each audit?
  - > Include In-progress classes?
  - > Include Preregistered classes?

As shown on the **Questions** tab in Transit, you must select the Catalog Year, School and Degree. You can then select at most one of each of the following: program, college, major, minor, concentration, specialization and liberal learning. Because of this, you cannot run audits for double-majors, for example.

You can then choose to freeze the audits and you can choose to print or create a PDF file out of them. If you choose to freeze the audits you will be guaranteed that they will remain in the database until you choose to delete them. (See the notes below on deleting frozen audits.) This allows you to view the audits on the web in the What-if History tab or to run SQL against the `dap_audit_dtl` to examine the high-level results. (See the notes below on running SQL against the audit results.) If you do not freeze the audits it is possible the audits will get deleted before you

get a chance to review them. It is recommended that you setup a least one freeze type in UCX-**AUD032** to be used for these batch what-if audits. You may choose to freeze all of these audits using a freeze type of WHATIF or you may setup different codes for the different what-if batches you will be running. After changing UCX-AUD032 be sure to run **ode20get** or **ucx12job** on the server and then restart Transit to see the changes. Another reason to freeze the audits is that you can then use the freeze-type later to delete just one set of audits and no others. Usually these batch generated what-if audits are temporary; you can use the freeze type as a way to tag these temporary audits and then delete them when you want. These freeze type also then serves as a tag type for the deletion/cleanup process.

Instead of, or in addition to, freezing the audits, you can print or create a PDF file. Review to the notes under DAP22 about printing and creating a PDF.

The DAP27 log file shows a line for each student processed, much like DAP22. However, this log file also shows you how complete the audit is for each student. Here you can see that one student is 100% complete with the requirements for the BUS\_CERT degree while the other two students still have some work to do. You may want to copy and paste this information into Excel and pull out just those students who are close to completion.

```
Processing student 19719874: UG/BUS_CERT... - 71% complete
Processing student 19843213: UG/BUS_CERT... - 100% complete
Processing student 82347211: UG/BUS_CERT... - 93% complete
```

Note that the **majors** are pulled from UCX-**AUD027** instead of UCX-STU023 and that the **minors** are pulled from UCX-**AUD029** instead of UCX-STU024.

Some of the labels on the **Questions** tab will be replaced with the labels you have configured in UCX-**SCR001**. For example, instead of “Select a Catalog Year” the text will say “Select a Academic Year” if you have “Academic Year” configured in UCX-SCR001 for “CATYR”.

Also note that a picklist will disappear from the **Questions** tab if you have no entries in the corresponding UCX table. For example, if you have no specializations listed in UCX-STU323 the “Select a Specialization” line will disappear.

## Deleting frozen audits

If you chose to freeze the audits you can delete them using the freeze type you specified. The easiest way to do this is to use **AUD02** in Transit. This tool makes use of the **dapdelaudits** script on the server to delete all audits created prior to a specified date with the freeze type you used. For this reason, you should consider if using a single freeze type of WHATIF is sufficient or if you need to create several different freeze types for the different batches you will be running.

## Running SQL against audit results

In order to have the audits saved to the database you must have the UCX-CFG020 DAP14 What-if History Count flag set to a value of 01 or greater – even if you are freezing them. Freezing them ensures that they will be there until you delete them while relying on the History Count only guarantees that they will be saved for some period of time before they will eventually be deleted when new what-if audits are generated – in batch or on the web. When the audit is saved to the database, a DAP\_AUDIT\_DTL record with high level information is recorded. You can find out how complete the degree is for each student by running a simple query such as the following:

```
Select dap_stu_id, dap_audit_pct from dap_audit_dtl
where dap_freeze_type='WHATIF'
Order by dap_stu_id;
```

This assumes you only have one set of batch what-if audits housed in the database at a time and have frozen the audits using the WHATIF freeze type. You can also use the dap\_degree field as part of your WHERE clause or use more than one freeze type when running your audits to help differentiate the different sets of audits you are running.

Note that the dap\_audit\_pct (audit percent complete) field is a CHAR field and will contain values such as 0037, 0098, 0100. Please keep this in mind when trying to select on specific values and using greater-than and less-than. It might be best to export this information to Excel and use the tools in Excel to format/filter/etc as needed.

The CPA dap\_result\_dtl records are not generated as part of DAP27 so you cannot query the details of the audit results. The details of the audit are stored in the dap\_audtree\_dtl but this table cannot be query since the data within in all binary.

## DAP28 – Alternate What-if Audits

You can run batch what-if audits from Transit on a set of selected students based on the **alternate** curriculum you have bridged for each student. You may have an alternate set of curriculum for some set of students for which you want to generate audits.

The screenshot shows the Transit software interface with the 'Questions' tab selected. The window title is 'DAP28' and the repository is 'DAP28 Alternate What-if Audits'. The interface includes a 'Set As Defaults' button and a section titled 'The curriculum use for these Alternate What-if Audits will come from the rad\_custom\_dtl based on what you bridged for each student.' Below this, it lists custom records: 'A-SCHOOL, A-DEGREE, A-CATYEAR' and 'A-PROGRAM, A-COLLEGE, A-MAJOR, A-MINOR, A-CONC, A-SPEC, A-LIBL'. The 'Freeze these alternate what-if audits? - choose a freeze type' dropdown is set to 'WIFALT'. Other options include 'Describe these alternate what-if audits' (Outcome curricula audits), 'What type of output do you want?' (empty), '> Which Audit Report do you wish to use?' (empty), and '> Create individual output files for each audit?' (empty). At the bottom, there are two checked checkboxes: '> Include In-progress classes?' and '> Include Preregistered classes?'.

As shown on the **Questions** tab in Transit, you are reminded that the curriculum for each student is found by retrieving from the rad\_custom\_dtl the alternate school, degree and catalog year. Each of these items must have been bridged with the names specified. For each of the other goal values, you may bridge multiple majors, minors, etc by specifying a number after each. For example, A-MAJOR1 could be bridge for CHEM and A-MAJOR2 could be bridged for BIOL. See the chart below for more information and how to bridge catalog year values for each goal value.

You can choose to freeze the audits and you can choose to print or create a PDF file out of them. If you choose to freeze the audits you will be guaranteed that they will remain in the database until you choose to delete them. (See the notes below on deleting frozen audits.) This allows you to view the audits on the web in the What-if History tab or to run SQL against the dap\_audit\_dtl to examine the high-level results. (See the notes below on running SQL against the audit results.) If you do not freeze the audits it is possible the audits will get deleted before you get a chance to review them. It is recommended that you setup a least one freeze type in UCX-AUD032 to be used for these batch what-if audits. You may choose to freeze all of these audits using a freeze type of WIFALT or you may setup different codes for the different what-if batches you will be running. After changing UCX-AUD032 be sure to run **ode20get** or **ucx12job** on the server and then restart Transit to see the changes. Another reason to freeze the audits is that you can then use the freeze-type later to delete just one set of audits and no others. Usually these batch generated what-if audits are temporary; you can use the freeze type as a way to tag these temporary audits and then delete them when you want. These freeze type also then serves as a tag type for the deletion/cleanup process.

Instead of, or in addition to, freezing the audits, you may print or create a PDF file. Please review

to the notes under DAP22 about printing and creating a PDF.

The DAP27 log file shows a line for each student processed, much like DAP22. However, this log file also shows you how complete the audit is for each student. Here you can see that one student is 100% complete with the requirements for the UG/BA degree while the other two students still have some work to do in their degrees. You may want to copy-n-paste this information into Excel and pull out just those students who are close to completion.

```
Processing student 19719874: UG/BS    ... - 71% complete
Processing student 19843213: UG/BA    ... - 100% complete
Processing student 82347211: GR/MBA   ... - 93% complete
```

One use of this is for Banner schools to bridge the outcome curricula as custom records. You will then essentially be running what-if audits on the outcome records. You may want to create a UCX-AUD032 freeze type of WIFOUT or OUTCOM to help you better identify these audits.

You will need to bridge the alternate curricula to the rad\_custom\_dtl by using the method appropriate for your student extract:

- Banner schools will setup **UCX-BAN080** records.
- Colleague schools will setup **integration.colleague.custom** records in custom.client.properties.
- Other schools will create BIF records as needed using the **R121CUST** record layouts.

Custom code	Meaning	Example code and value
A-SCHOOL	School (aka Level)	A-SCHOOL=UG
A-DEGREE	Degree	A-DEGREE=BS
A-CATYEAR	Overall Catalog Year	A-CATYEAR=2014
A-PROGRAMx	xth Program – x is 1, 2, 3, etc	A-PROGRAM1=BS_CHEM
A-PROGRAMxCY	xth Program Catalog Year	A-PROGRAM1CY=2015
A-MAJORx	xth Major – x is 1, 2, 3, etc	A-MAJOR1=CHEM
A-MAJORxCY	xth Major Catalog Year	A-MAJOR1CY=2013

The same codes are supported for COLLEGE, MINOR, CONC, SPEC, and LIBL following the pattern you see above for PROGRAM and MAJOR.

The CY catalog year code and value are optional for any given goal. If it is not found the overall catalog year value is used when determining the correct Scribe block to use for the given goal code.

For Banner schools, if your school is using the Program-as-degree feature, you will instead bridge the program code (example: BA\_HIST) as the A-DEGREE and optionally bridge the degree code (example: BA) as A-PROGRAM1.

## Deleting frozen audits

If you chose to freeze the audits you can delete them using the freeze type you specified. The easiest way to do this is to use **AUD02** in Transit. This tool makes use of the **dapdelaudits** script on the server to delete all audits created prior to a specified date with the freeze type you used. For this reason, you should consider if using a single freeze type of WIFALT is sufficient or if you need to create several different freeze types for the different batches you will be running.

## Running SQL against audit results

In order to have the audits saved to the database you must have the UCX-CFG020 DAP14 What-if History Count flag set to a value of 01 or greater – even if you are freezing them. Freezing them ensures that they will be there until you delete them while relying on the History Count only guarantees that they will be saved for some period of time before they will eventually be deleted when new what-if audits are generated – in batch or on the web. When the audit is saved to the database, a DAP\_AUDIT\_DTL record with high level information is recorded. You can find out how complete the degree is for each student by running a simple query such as the following:

```
Select dap_stu_id, dap_audit_pct from dap_audit_dtl
where dap_freeze_type='WIFALT'
Order by dap_stu_id;
```

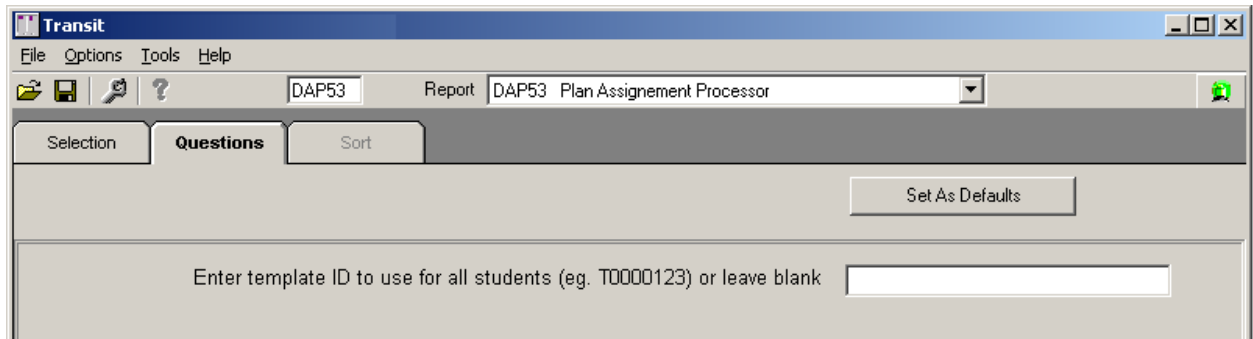
This assumes you only have one set of batch what-if audits housed in the database at a time and have frozen the audits using the WIFALT freeze type. You can also use the dap\_degree field as part of your WHERE clause or use more than one freeze type when running your audits to help differentiate the different sets of audits you are running.

Note that the dap\_audit\_pct (audit percent complete) field is a CHAR field and will contain values such as 0037, 0098, 0100. Please keep this in mind when trying to select on specific values and using greater-than and less-than. It might be best to export this information to Excel and use the tools in Excel to format/filter/etc as needed.

The CPA dap\_result\_dtl records are not generated as part of DAP28 so you cannot query the details of the audit results. The details of the audit are stored in the dap\_audtree\_dtl but this table cannot be query since the data within in all binary.

## DAP53 - Plan Assignment Processor

You can assign plans in batch to students using DAP53 in Transit. You can specify the template that is to be used for all students in your pool or you can let the processor determine the best template for the student based on their catalog year, school, degree and major.



Here are the steps used to find an appropriate template for each student:

- Try to find a template based on the student's **catalog year, school, degree and major**
- If none found try to find a template based on the student's **catalog year, school, and degree** (blank major)
- If none found try to find a template based on the student's **catalog year and school** (blank degree and major)
- If none found try to find a template based on the student's **catalog year** (blank school, degree, and major)
- If none found no plan will be created for the student

The DAP53 action files viewable in Transit gives a summary of how many plans were and were not created for your pool of students.

```
Plans created = 2
No plans created for students with pre-existing plans = 0
No plans created since no appropriate templates were found = 21
No plans created because of errors loading plans = 0
Total Students Processed = 23
```

You can view the DAP53 log file for more details.

The batch assignment processor does not attempt to assign a plan to a student if the student already has a plan recorded.

When building templates you should be sure to create one for every type of student – those that do not have a major on their degree record and perhaps those that don't even have a degree code recorded.

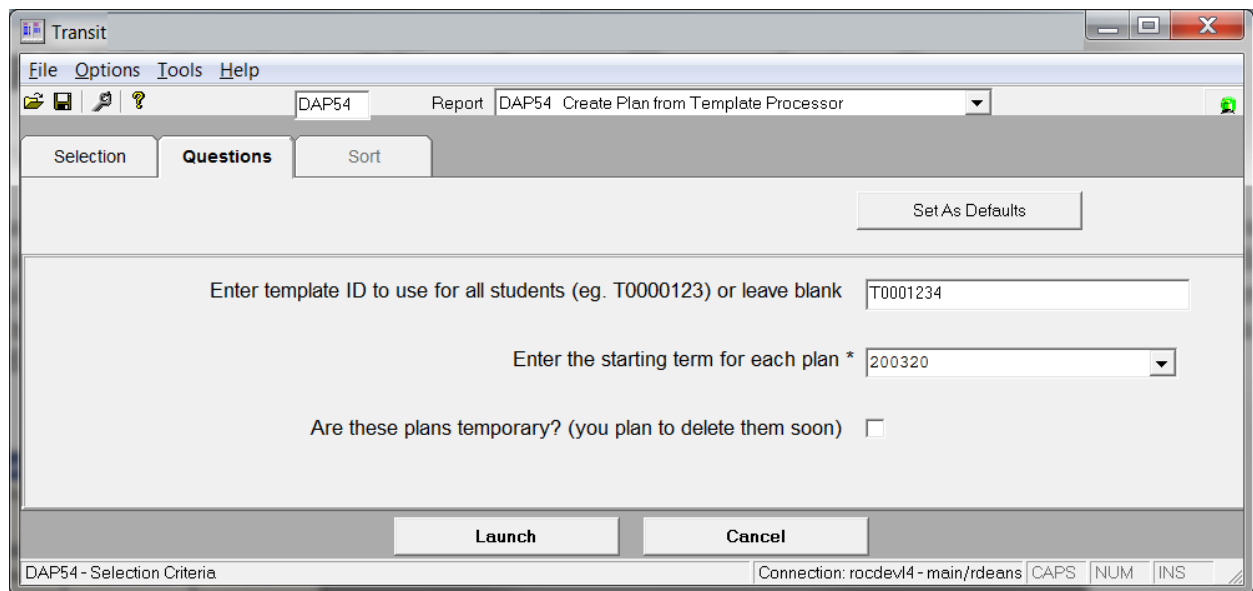
## DAP54 – Create Plan from Template Processor

**Note:** The DAP54 processor is for the new SEP while DAP53 is for the old SEP.

You can assign plans in a batch to students using DAP54 in Transit. You can specify the template that is to be used for all students in your pool or you may let the processor determine the best template for the student based on their curriculum. **Note:** If you use the template ID option, the template you specify **must** have a DEGREE tag (either optional or required). If there isn't a DEGREE tag, plans will not be created for any student.

You must also select the starting term for the plans created since the template only specifies the term type – not the actual terms. If you do specify a template be sure the starting term you specify has the same term type as the first term type on the template.

For the students selected, you can create the plans as temporary ones. When the **Are these plans temporary** checkbox is selected, the plans that are created will have the sep\_plan.create\_what set to TEMPORARY.



The screenshot shows the 'DAP54 - Selection Criteria' dialog box in the Transit application. The window title is 'Transit'. The menu bar includes 'File', 'Options', 'Tools', and 'Help'. The toolbar contains icons for file operations and a help icon. The 'Report' dropdown is set to 'DAP54 Create Plan from Template Processor'. Below the toolbar are three tabs: 'Selection', 'Questions', and 'Sort'. A 'Set As Defaults' button is located on the right. The main area contains the following fields and controls:

- A text input field for 'Enter template ID to use for all students (eg. T000123) or leave blank' with the value 'T0001234'.
- A dropdown menu for 'Enter the starting term for each plan \*' with the value '200320'.
- A checkbox for 'Are these plans temporary? (you plan to delete them soon)' which is currently unchecked.

At the bottom, there are 'Launch' and 'Cancel' buttons. The status bar at the bottom shows 'DAP54 - Selection Criteria' and 'Connection: rocdev14 - main/rdeans | CAPS | NUM | INS'.

The steps used to find an appropriate template for each student are as follows:

If the template is specified:

- The template specified will be retrieved from the database
- It is assumed that the template has a degree tag
- For each student, the student's degree will be compared to the template degree to be sure they match
- If the degree on the template does not match the student's degree, no plan is created
- If a student has multiple degrees, the degree matching the template will be chosen and saved on the new plan

If the template is not specified:

- Try to find a template based on all of the student's data (school, degree, major, minor, conc, etc.) This is done for each of the student's degrees.
- If more than one matching template is found, the template with the most tags is chosen. If multiple templates have the most matching tags, then the majors on the matching templates are examined. The template for the primary major will be chosen. If multiple templates match as a result of multiple concentrations for a single major, then no template is selected.
  - For example, a template with the student's minor may be found and another with the student's conc may be found. Since both are valid for the student, no decision can be made as to which to use.
  - Another example, templates are found for CHEM and MATH majors. Since CHEM is the student's primary major, that template is selected.
  - Another example, templates are found for the VIOLIN and PIANO concentrations for the music major. In this case a decision is not made; no template is selected.
- If any template found has a template scheme whose first term type does not match the term type of the starting term specified, the template will be discarded.
- See the *Finding a template for degree* section below for more information.

The DAP54 log files viewable in Transit gives a summary of how many plans were, and were not, created for your pool of students.

```
--- Plan Creator has completed
91 students were selected for plan creation
83 students received a new plan
```

The batch assignment processor does not attempt to assign a plan to a student if the student already has a plan recorded for the degree.

When building templates you should be sure to create one for every type of student – those that do not have a major on their degree record and perhaps those that don't even have a degree code recorded.

Temporary plans are only created for students who do not have an active, approved plan for the given degree. When the checkbox is not selected, no plan will be created for the student if they already have a plan – even if it is inactive/unapproved. However, multiple temporary plans for the same degree will be created if DAP54 is run more than once without deleting existing temporary plans from the database. To identify temporary plans, "TEMPORARY" is recorded on the `sep_plan.create_what` field in the database. You can delete temporary plans later by doing the following in the Degree Works database:

```
execute sep_helper.delete_plans_create_what ('TEMPORARY');
```

## Finding a template for a degree

The assumption for all examples is that **UCX-SEP001** has the following setup:

SCHOOL	Required=Y	MatchStudentGoalData=Y
DEGREE	Required=Y	MatchStudentGoalData=Y
MAJOR	Required=Y	MatchStudentGoalData=Y
MINOR	Required=N	MatchStudentGoalData=Y
CONC	Required=N	MatchStudentGoalData=Y
COOP	Required=N	MatchStudentGoalData=N

### Example 1

#### Templates for CHEM major

1	SCHOOL=UG	DEGREE=BS	MAJOR=CHEM	MINOR=ART	CONC=BIOCHEM
2	SCHOOL=UG	DEGREE=BS	MAJOR=CHEM	MINOR=ART	
3	SCHOOL=UG	DEGREE=BS	MAJOR=CHEM		CONC=BIOCHEM
4	SCHOOL=UG	DEGREE=BS	MAJOR=CHEM		

**Student's degree:** SCHOOL=UG, DEGREE=BS, MAJOR=CHEM

Template 1 is not chosen because the student does not have the minor or concentration.

Template 2 is not chosen because the student does not have the minor.

Template 3 is not chosen because the student does not have the concentration.

**Template 4 is chosen** because the student has the matching school, degree and major.

### Example 2

#### Templates for CHEM major – with COOP tag

1	SCHOOL=UG	DEGREE=BS	MAJOR=CHEM	MINOR=ART	COOP=SPRING3
2	SCHOOL=UG	DEGREE=BS	MAJOR=CHEM		COOP=SPRING3

**Student's degree:** SCHOOL=UG, DEGREE=BS, MAJOR=CHEM

Template 1 is not chosen because the student does not have the minor.

**Template 2 is chosen** because the student has the matching school, degree and major. Since the co-op has the MatchStudentGoalData flat set to N in UCX-SEP001 this is not compared against the student's degree.

### Example 3

#### Templates for CHEM major; student has minor and conc also

- 1 SCHOOL=UG DEGREE=BS MAJOR=CHEM MINOR=ART CONC=BIOCHEM
- 2 SCHOOL=UG DEGREE=BS MAJOR=CHEM

**Student's degree:** SCHOOL=UG, DEGREE=BS, MAJOR=CHEM; MINOR=ART, CONC=BIOCHEM

The first template is selected because we search on all of the student's goal information (school, degree, major, minor and conc in this situation).

Template 1 is valid because both the minor and concentration match.

**Template 1 is chosen.**

### Example 4

#### Templates for CHEM major; student has minor and concentration – but templates have no concentration

- 1 SCHOOL=UG DEGREE=BS MAJOR=CHEM MINOR=ART
- 2 SCHOOL=UG DEGREE=BS MAJOR=CHEM

**Student's degree:** SCHOOL=UG, DEGREE=BS, MAJOR=CHEM; MINOR=ART, CONC=BIOCHEM

In this example, no template was found matching all of the student's data but two templates were found matching some of the student's data. Since the first template matches on more data than the second one, **template 1 is chosen.**

### Example 5

#### Templates for CHEM major; student has minor and conc also

- 1 SCHOOL=UG DEGREE=BS MAJOR=CHEM MINOR=ART CONC=BIOCHEM
- 2 SCHOOL=UG DEGREE=BS MAJOR=CHEM MINOR=ART
- 3 SCHOOL=UG DEGREE=BS MAJOR=CHEM CONC=BIOCHEM
- 4 SCHOOL=UG DEGREE=BS MAJOR=CHEM

**Student's degree:** SCHOOL=UG, DEGREE=BS, MAJOR=CHEM; MINOR=ART, CONC=BIOCHEM

These templates are selected because we search on all of the student's goal information (school, degree, major, minor, and conc in this situation).

Template 1 is valid because both the minor and concentration match.

Template 2 is valid because the minor matches.

Template 3 is valid because the concentration matches.

Template 4 is also valid because all of its tags match the student's data.

However, **template 1 is chosen** because it has more tags that match the student's data.

## Example 6

### Templates for CHEM major; student has minor and conc also

- |   |           |           |            |           |              |
|---|-----------|-----------|------------|-----------|--------------|
| 1 | SCHOOL=UG | DEGREE=BS | MAJOR=CHEM | MINOR=ART |              |
| 2 | SCHOOL=UG | DEGREE=BS | MAJOR=CHEM |           | CONC=BIOCHEM |
| 3 | SCHOOL=UG | DEGREE=BS | MAJOR=CHEM |           |              |

**Student's degree:** SCHOOL=UG, DEGREE=BS, MAJOR=CHEM; MINOR=ART, CONC=BIOCHEM

These templates are selected because we search on all of the student's goal information (school, degree, major, minor and conc in this situation).

Template 1 is valid because the minor matches.

Template 2 is valid because the concentration matches.

Template 3 is also valid because all of its tags match the student's data.

Both templates 1 and 2 have the same number of matching tags but it cannot be determined which one should take precedence so **template 3 is chosen** because it is the best match of required template tags to the student's goal information.

## Example 7

### Templates for CHEM and MATH major

- |   |           |           |            |  |
|---|-----------|-----------|------------|--|
| 1 | SCHOOL=UG | DEGREE=BS | MAJOR=CHEM |  |
| 2 | SCHOOL=UG | DEGREE=BS | MAJOR=MATH |  |

**Student's degree:** SCHOOL=UG, DEGREE=BS, MAJOR=CHEM; MAJOR=MATH

These templates are selected because we search on all of the student's goal information (school, degree, major, minor and conc in this situation).

Template 1 is valid because the major matches.

Template 2 is valid because the major matches.

**Template 1 is chosen** because CHEM is the student's primary major.

## Example 8

### Templates for MUSIC major:

- |   |           |           |             |             |
|---|-----------|-----------|-------------|-------------|
| 1 | SCHOOL=UG | DEGREE=BA | MAJOR=MUSIC | CONC=PIANO  |
| 2 | SCHOOL=UG | DEGREE=BA | MAJOR=MUSIC | CONC=VIOLIN |

**Student's degree:** SCHOOL=UG, DEGREE=BS, MAJOR=MUSIC; CONC=PIANO; CONC=VIOLIN

These templates are selected because we search on all of the student's goal information (school, degree, major, minor, and conc in this situation).

Template 1 is valid because the major and conc matches.

Template 2 is valid because the major and conc matches.

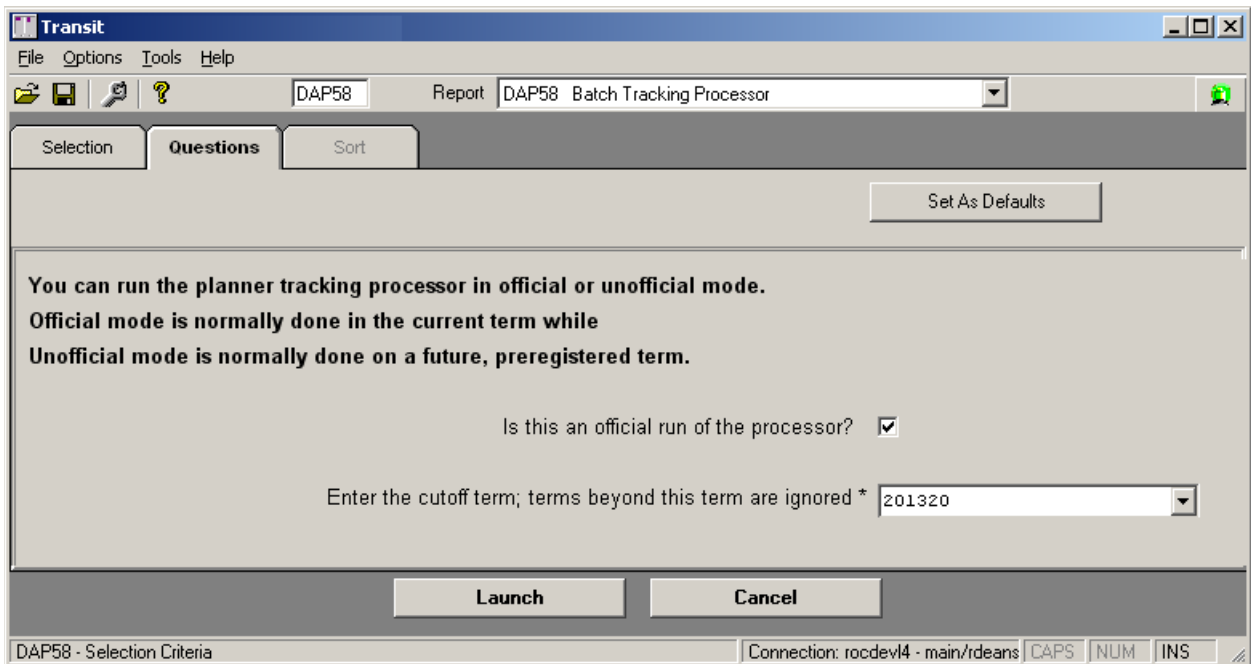
Since PIANO is the student's primary concentration, **template 1 will be chosen.**

## DAP58 – Batch Tracking Processor

The DAP58 processor allows you to track whether or not your students are following their educational plans. The tracking processor checks each plan to see if the specified courses were taken on or before the planned term, checks GPA requirements, checks if minimum test scores were achieved, and if non course requirements were taken. Each requirement, term, and plan has its tracking status updated allowing you to run reports against the data.

When you run DAP58, you specify whether you want to run it in the official or unofficial mode. On the plan (sep\_plan) and term (sep\_plan\_term) tables this determines whether the OFFICIAL\_TRACKING\_STATUS or the UNOFFICIAL\_TRACKING\_STATUS is updated by the processor. The requirements' TRACKING\_STATUS field is updated regardless of the mode being used. The tracking status values are ON\_TRACK, OFF\_TRACK, NOT\_EVALUATED, and NOT\_FOUND. Newly added requirements and those in the future have a status of NOT\_EVALUATED while certain GPA requirements may have a NOT\_FOUND status if the GPA for the past term cannot be determined.

You also must specify a cutoff term when running the processor. Planned terms following this cutoff term are ignored by the processor. Only those terms from UCX-STU016 with Show in SEP Picklist set to Y will appear in this picklist.



The DAP58 log files viewable in Transit give a summary of how many plans were processed for your pool of students.

*241 students were selected for tracking status update  
241 students were successfully updated*

When you run the processor in the unofficial mode, the updated tracking status is **not** reflected on the Web interface for plans. The unofficial mode is meant for reporting purposes. To see the updated tracking status on the Web interface, run the processor in the official mode. See the

planner documentation for more information.

## DAP59 – Batch Timetabling Processor

The DAP59 processor allows you to run audits against students' educational plans. The timetabling processor runs an audit for each of the student's active and locked/approved plans for the given school/level specified. If an active and locked/approved plan cannot be found for a student, the processor will look for a TEMPORARY plan created from a template by DAP54. All classes after the student's active term up to and including the cutoff term specified will be included in the audit. If the student has preregistered for the same classes that appear on the plan, the duplicate classes appearing on the plan will not be included, redundantly in the audit. This special planner audit is then saved to the CPA tables allowing you to find out how each of the planned classes applied in the audit.

Also see the *Timetabling* section in the *Advanced Reporting Technical Guide*.

You must specify the school/level of the students and plans you want to process.

You also must specify a cutoff term when running the processor. Planned terms following this cutoff term are ignored by the processor.

The screenshot shows the 'DAP59 Batch Timetabling Processor' dialog box within the 'Transit' application. The window title is 'Transit'. The menu bar includes 'File', 'Options', 'Tools', and 'Help'. The toolbar contains icons for file operations and a help icon. The main area has three tabs: 'Selection', 'Questions', and 'Sort', with 'Questions' currently selected. A 'Set As Defaults' button is located in the top right of the main area. The main text area contains the following instructions:

**A new audit will be run and Timetabling CPA data will be generated but the audit will not be saved.**  
**Planned classes up to and including the cutoff term will be included.**  
**All in-progress and preregistered classes up to and including the cutoff term that are on the student's record will also be included**  
**- with duplicate classes removed.**

Below the text are two dropdown menus:

Input School/Level Code \* LW  
Enter the cutoff term; planned terms beyond this term are ignored \* 201410

At the bottom of the dialog are 'Launch' and 'Cancel' buttons. The status bar at the bottom shows 'DAP59 - Selection Criteria' and 'Connection: rocdev14 - main/rdeans CAPS NUM INS'.

## RAD11 – The Traditional Bridge Batch Processor

The “traditional” approach to updating the Degree Works data repository (RAD) from the Student Information System (SIS) involves “bridging” information from the SIS database to the RAD database using the “RADBRIDGE” batch program as the processor. It takes as input a file containing a list of data files which have been prepared by a client-written program operating on the SIS side that prepares new information for transmission to Degree Works. This list file is called RAD11M01 and resides in the data sub-directory of the test or production environment. The RAD11M01 file is a record oriented file (there is a newline character on each line). Each record in RAD11M01 contains the name of a data file containing data to be processed.

As the “classic” approach, this methodology may always be used to move data between the SIS and RAD irrespective of other methods that may be developed by Ellucian for specific Student Information systems, such as an integrated data extract, which uses RAD30 (see next section).

### Data Files

The data files used by RAD11 to process data from the SIS (Student Information System) are fixed length files 1000 bytes in length. Each data file contains records described in the Bridge Interface Guide and they also reside in the data sub-directory of the test or production environment. A data file can contain the data for multiple students, ETS schools, courses, UCX tables and Degree Works course equivalency information. It is very important that data for a particular ID (student or staff member) not be split across multiple files.

RAD11 identifies a record with a data file according to its indicator. RAD11 sorts data according to ID, indicator and term if appropriate. For this reason it is essential that indicators be exact. Refer to the Bridge Specifications - RAD documentation for a list of indicators and the data set in the Degree Works data repository to which the record is applied.

### FTP Process

The data files used by RAD11 to process data from the SIS (Student Information System) should be delivered to the admin/data sub-directory of the test or production environment via FTP or a similar process. The details of this process should be worked out with Ellucian. The basic syntax for delivering a data file via FTP is:

```
> ascii
> put datafile
> put RAD11M01
```

An example RAD11M01 file might be:

```
students1
students2
```

Where students1 is a set of BIF records for one set of students and students2 is another file for a separate set of students. Normally the RAD11M01 file has a single file with all of the students to be bridged within it – but you have the option to list multiple files in RAD11M01. The RAD11M01 file also optional. You can simply bridge the datafiles themselves and refer to them directly when you run RAD11JOB.

## Running the RAD11 Processor

RAD11 can be scheduled to run nightly, weekly or on some other regular basis. It is recommended that RAD11 be run at least once a week to refresh student data for the current term or semester.

You may run simultaneous RAD11 processes to take advantage of the multiple processors you have on your machine by splitting your data up into multiple files and listing these files in the RAD11M01 file. For example, if your site has 4 CPUs, split your bridge data files into four separate files and add these files to the RAD11M01. When RAD11JOB runs, it will spawn one process for each of the data files listed in RAD11M01. You may want to run a batch on seniors, another on juniors, another on sophomores and yet another on first-year students. Running batch bridge processes like this in parallel should result in better throughput if you have more than one CPU on your system.

Instead of creating multiple files you may also deliver a single file to Degree Works. When one file of students is listed in RAD11M01 the DGWCPUCOUNT environment variable (set in dwenv.config) comes into play. The student data file will be split up into multiple files – based on this setting – allowing each file to be run by RAD11 simultaneously – having the same effect as having supplied RAD11 with multiple files to begin with.

Each time RAD11 processes data for an ID all records for that ID should be included in the data file. RAD11 will delete old records completely before adding new ones from information provided in the data file. A total refresh of student data is done each time RAD11 processes an ID. If RAD11M01 does not exist or is empty then RAD11 will do no processing and reschedule its next run, if configured to do so. See the section below for more information.

The RAD Bridge program can be executed either on demand from Transit or scheduled on a recurring basis using the UNIX "at" utility. Therefore, there are 2 scripts for executing RAD11: RAD11JOB and rad11sch. Both reside in the batch directory. RAD11JOB is used by Transit, while rad11sch is used with "at" to schedule a recurring bridge.

The rad11sch script is used to schedule RAD11JOB on a periodic basis using "at". Specifically, rad11sch passes the time interval for "at" to the RAD11JOB script. The time interval is configured by you upon installation of Degree Works or whenever you are ready to schedule a recurring bridge of student data. Modify the setting in rad11sch to run RAD11JOB every x minutes, hours or days beginning at y time. You may want to run RAD11JOB every 30 minutes to see if your system has sent it new files to process. If RAD11JOB finds no files, it will remain idle.

```
$ cat rad11sch
```

```
#!/usr/bin/sh
# rad11sch is a script that executes RAD11JOB using the "at" utility.
# It is used to schedule the RAD Bridge program (RAD11) to run at
# recurring intervals.
#
#####
# Run rad11job telling it to reschedule rad11sch tomorrow at 3 a.m.
RAD11JOB $Pid -t "03:00am tomorrow"
#####

# Other examples
# RAD11JOB $Pid -t "now + 30 minutes" - 30 minutes from now
# RAD11JOB $Pid -t "now + 2 hours" - 2 hours from now
# RAD11JOB $Pid -t "now + 1 day" - one day from now
# RAD11JOB $Pid -t "0300 tomorrow" - at 3am tomorrow
```

```

# RAD11JOB $Pid -t "03:00am + 1 week" - at 3am 1 week from now
# RAD11JOB $Pid -t "0300" - 3am tomorrow as long as 3am has
already passed
#
# If no date is given, today is assumed if the given
# time is greater than the current time, and tomorrow is
# assumed if it is less. If the given month is less than
# the current month (and no year is given), next year is
# assumed.

```

RAD11JOB takes in this optional time parameter and runs rad11sch using "at".

For more information on the "at" utility read the man pages on your system.

RAD11JOB is executed from Transit without the optional time parameter. It executes on demand, as opposed to rad11sch which executes as scheduled.

### Errors, Warnings and Success

RAD11 will write fatal errors, warnings and success to the rad\_log\_dtl of the Degree Works data repository. RAD11 will also extract these messages to its log at the end of its run. The log file can be viewed using the View Jobs button in Transit.

#### Fatal Errors

RAD11 requires two records to successfully process an ID: PRIM and DEGR. If either of these records does not exist a fatal error is written to the rad\_log\_dtl and processing continues with the next ID. In addition, essential data items in these records are validated against UCX tables.

Invalid data will also generate a fatal error. The TERM field in the PRIM record is validated against UCX-ST016, the school field in the rad\_goal\_dtl file is validated against UCX-STU350 and catalog\_yr from the rad\_goal\_dtl file is validated against UCX-STU035. ID numbers in each of these data files are validated against the ID number in each header record.

If an ID is a staff member then only a PRIM record is required.

#### Warnings

RAD11 will write warnings to the rad\_log\_dtl if configured to do so. Warnings are produced if ID numbers in the remaining non-required data files do not match the ID in the header record. When a warning is produced for one of these data files the record in question is skipped and processing continues with the next record for that ID. The first 200 bytes of the record are written to the rad\_log\_dtl for easy identification.

#### Success

RAD11 will write success to the rad\_log\_dtl for each successful processing of an ID, if configured to do so. "OKAY" is the message written along with the ID number.

#### Additional Modes

RAD11 can be run to update ETS schools, courses, UCX tables, Degree Works equivalency information, delete all for an ID or to exchange ID numbers. Data for these additional modes can be included in the same data file for an ID or placed in other files as long as the file is listed in RAD11M01.

#### Delete ID

Delete ID deletes all data for and ID from the Degree Works data repository. See UCX table UCX-CFG020 RADBRIDGE to set a configuration flag if DAPDB data such as notes should be deleted as well. The Bridge Interface Guide provides the data layout for this mode.

#### Change ID

If an ID number for a student changes for any reason then RAD11 can be used to change the ID number in all tables that an ID has data for. The Bridge Interface Guide provides the data layout for this mode.

### **UCX Update**

RAD11 can be used to perform a batch update of UCX tables. The Bridge Interface Guide provides the data layout for this mode.

### **ETS Update**

RAD11 can be used to perform a batch update of ETS data. The Bridge Interface Guide provides the data layout for this mode. RAD11 will not delete all ETS data in the Degree Works data repository. It is necessary to include only those records you wish to add or update.

### **Course Update**

RAD11 can be used to perform a batch update of Course data. The Bridge Interface Guide provides the data layout for this mode. RAD11 will not delete all Course data in the Degree Works data repository. It is necessary to include only those records you wish to add or update.

### **dap\_eqv\_crs\_mst Update**

RAD11 can be used to perform a batch update of Course equivalency data. The Bridge Interface Guide provides the data layout for this mode. RAD11 will delete all Course equivalency data in the Degree Works data repository. It is therefore necessary to include all equivalency records when updating the `dap_eqv_crs_mst` table using RAD11. To update/add a single `dap_eqv_crs_mst` record, use Surecode to edit/add the record to the UCX-CFG070 table and then run the `ucx12job` (or `dapucx2eqv` script) to load the records to the `dap_eqv_crs_mst` table.

### **Changed Data**

RAD11 stores a value in the `rad_hash_mst` representing the last set of student data bridged. When RAD11 receives a new set of data for a student the hash value derived from the new set of data is compared to the value stored in the `rad_hash_mst`. If the values are the same RAD11 recognizes that the new student data is exactly what is already in the database and does not save this new data. If the old hash value differs from the new value, RAD11 writes a date along with the new hash value to the `rad_hash_mst`; the new set of data is also saved to the database. In performing this hash check, RAD11 will not spend time processing student data that does not need to be bridged.

### **Over-riding the rad\_hash\_mst**

While the `rad_hash_mst` allows the RAD11 processor to skip student records for which the data has not changed, it is sometimes necessary to ensure that new records are added to the database regardless of the `rad_hash_mst` value. This is particularly true when conducting performance benchmark testing. You may tell RAD11 to ignore the `rad_hash_mst` by using the "FORCE" option:

```
$ RAD11JOB FORCE
```

RAD11 simply knows to ignore the value in the current `rad_hash_mst` and bridge the student regardless of not having changed data.

### **DAP22 after RAD11 completes**

After RAD11 completes, the `radhashid` script extracts the student IDs from the `rad_hash_mst` that were just bridged by RAD11. Only those students who actually had changed data will be selected from the `rad_hash_mst`. Once the list of student IDs has been created, the `batch/dap22r11` script launches DAP22 to run a new audit for each of the students. In doing this, we are ensuring that an up-to-date audit exists based on the latest bridged data for each student. A batch-id and date on the `rad_hash_mst` ensures that DAP22 is run on those students just bridged by RAD11; students bridged earlier in the day by another RAD11 job will not be considered.

## Notifications

The RAD11JOB executes a *rad11started* when it starts up and a *rad11ended* script when it finishes. These scripts may be modified to send an email to your IT department or perform some other duty as needed. The default scripts don't perform any activity.

Do not intermingle different types of records into the same data file. For example, course data should be in one file, UCX data should be another, equivalence records in another and student data in a separate file. In addition, do not mix different file types in one RAD11M01 like this:

```
COURSE
UCX
EQUIV
STUDENT
```

Instead run each type of data in separate runs of RAD11 like this:

```
$ RAD11JOB COURSE
$ RAD11JOB UCX
$ RAD11JOB EQUIV
$ RAD11JOB STUDENT
```

In these examples we are specifying the actual name of the files in the admin/data directory instead of the RAD11M01 file. You can actually refer to the bridge file directly instead of specifying the RAD11M01. Above the COURSE file has all of the course BIF records, the UCX file has all of the BIF UCX records, etc.

## RAD30 – Banner Extract and Bridge

This process is further described in the *Banner Considerations Technical Guide*.

Transit

File Options Tools Help

RAD30 Report RAD30 Banner Extract and Bridge

Selection Questions Sort

Set As Defaults

What type of extract do you want to run today? \* [Dropdown]

Do you want to use the default SQL file to select the students?

**For STUDENT mode you may use the default SQL file or enter Selection criteria**  
**For STUDENT mode local/sql/bannerstudent.sql is the default SQL file**  
**For APPLICANT mode local/sql/bannerapplicants.sql will always be used**  
**For ADVISOR mode local/sql/banneradvisor.sql will always be used**  
**New audits will be generated for students with data changes**  
**New audits will be generated for applicants with data changes**  
**Be sure your CFG020 BANNER settings are correct before running the extract**

Launch Cancel

Run-time questions for RAD30 will be:

### **What type of extract do you want to run today?**

Choices are given in the drop-down list box and allow extraction of data for each of these groups:

- (ADVISOR) - Advisor
- (APPLICANT) - Applicant
- (COURSE) – Course
- (CURRRULE) – Curriculum Rules
- (EQUIV) - Equivalencies
- (ETS) - Transfer school – for Transfer Equivalency
- (MAPPINGS) - Transfer articulation
- (STAFF) - Staff
- (STUDENT) - Student
- (UCX) - Validation tables

Only one group can be processed at a time.

### **Do you want to use the default SQL file to select the students?**

If this checkbox is left empty, then the students selected will be based on the selection criteria entered under the Selection Tab. Checking this box will override the Selection criteria from the Selection Tab.

This checkbox is applicable only if STUDENT is chosen as the extract type (previous question).

### Selection tab:

When selecting students the element numbers that start with “B” specify that the searching should be done in Banner. In RAD30 all of the element numbers start with “B” so searching is always done in Banner. There is no underlying criteria specifying that the students are active students so be careful with your searches. For example, if you select on major of history you will find new students, current students and perhaps graduated students in your selection.

## Scribe Management Reports

The following reports (all prefixed with SCR) are used to help with the management of requirement blocks created with Scribe.

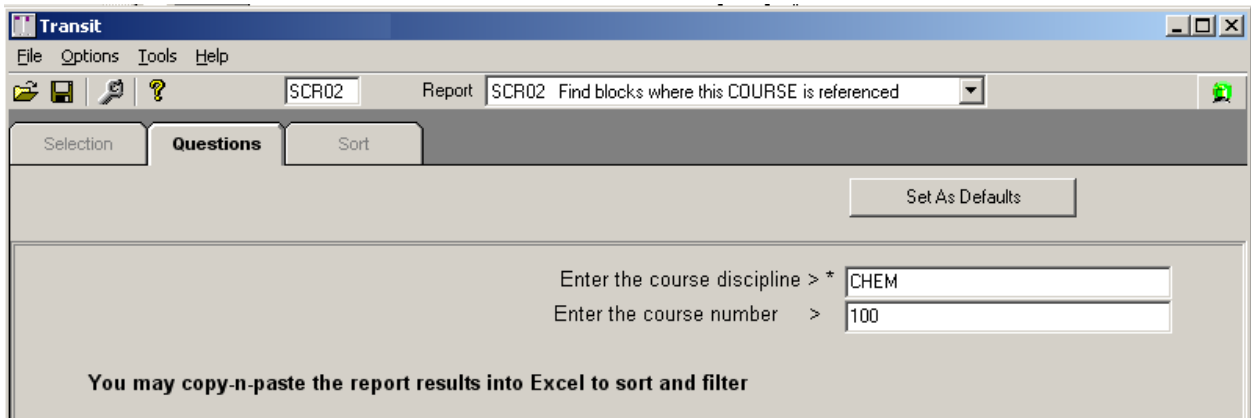
As indicated by the note appearing on the Question tab, you may copy the data into an Excel spreadsheet and sort the columns and filter as needed – the report contains tab characters allowing the data to be properly placed in discrete Excel columns.

To gain access to these SCR reports you need to have the **PTSSCRIB** key added to your security record.

### SCR02 – Find blocks where this COURSE is referenced

When conducting maintenance on your blocks in Scribe you may need to find the blocks that reference certain courses. You can do this through the **SCR02** job – this job makes use of the **dapreqcrs** script but you only need access to Transit and not the command prompt.

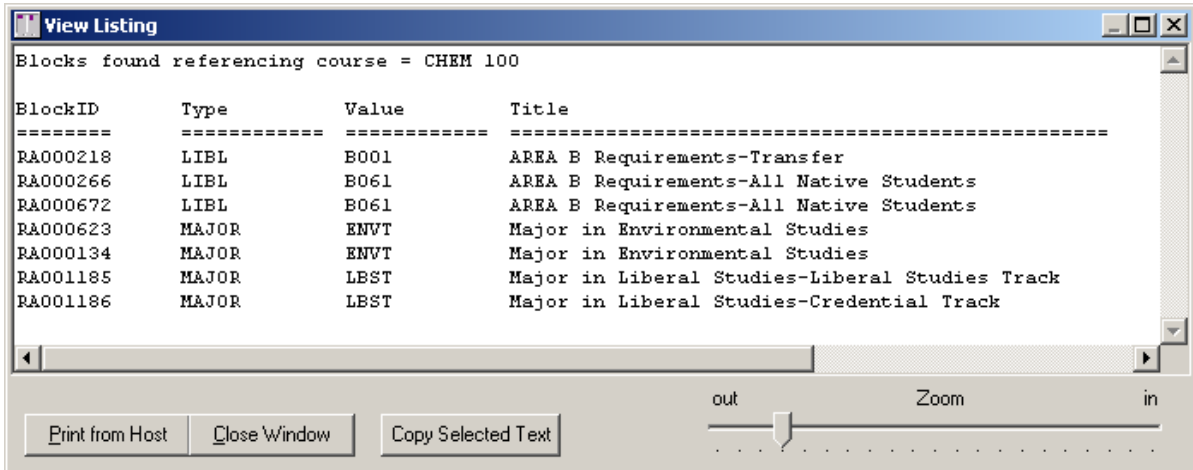
When running the job you need to enter a discipline and optionally enter a course number.



The screenshot shows the Transit application window with the following elements:

- Window title: Transit
- Menu bar: File Options Tools Help
- Toolbar: Contains icons for file operations and help.
- Report selection: A dropdown menu showing "SCR02" and a description "SCR02 Find blocks where this COURSE is referenced".
- Navigation tabs: Selection, Questions, Sort.
- Buttons: "Set As Defaults" and "Enter" (represented by a green arrow icon).
- Input fields:
  - "Enter the course discipline > \*": Text box containing "CHEM".
  - "Enter the course number >": Text box containing "100".
- Instructional text: "You may copy-n-paste the report results into Excel to sort and filter".

When the job runs the output will be sent to a Report file. Click on the View Jobs button in Transit to view the report created. The report will list the blocks that reference this course.

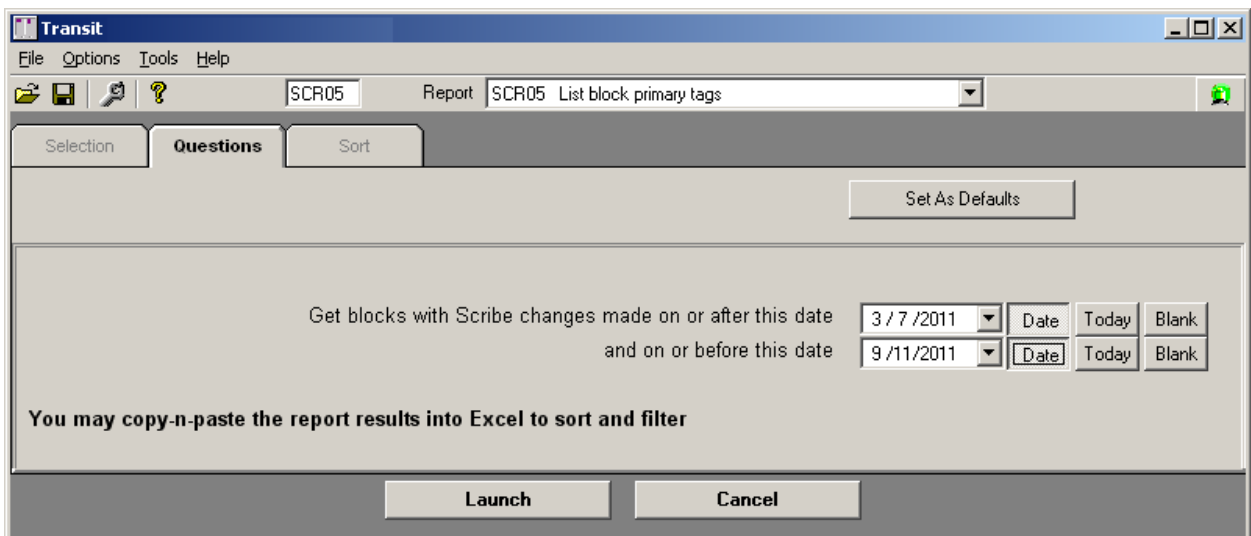


When UCX-CFG020 DAP14 'Process Equivalencies' is "Y," SCR02 will not find blocks that are scribed with a course ID that has been renumbered (CFG070) or split (CFG078). This is because SCR02 does not search the actual block text in the dap\_req\_block table; it searches the dap\_req\_crs\_dtl view which stores the parsed equivalent of the course ID in the text.

## SCR05 – List blocks changed by date range

You may want to simply get a list of all of the blocks you have in Degree Works. You can use the SCR05 job in Transit to get a simple report listing all of your blocks – sorted by the block type and value. This job makes use of the **dapreqlist** script on the server.

You can optionally specify a date range for changes made in Scribe to the blocks. You can specify the start date and leave the end date blank or simply click on "Today" for the ending date. To set both dates to blank so that no filtering is done on the modify date click on Blank on both fields. You may also click Set As Defaults to tell Transit to use these settings the next time you attempt to run the report. If you do specify a date range the report will additionally list the ModifyDate for each block selected.

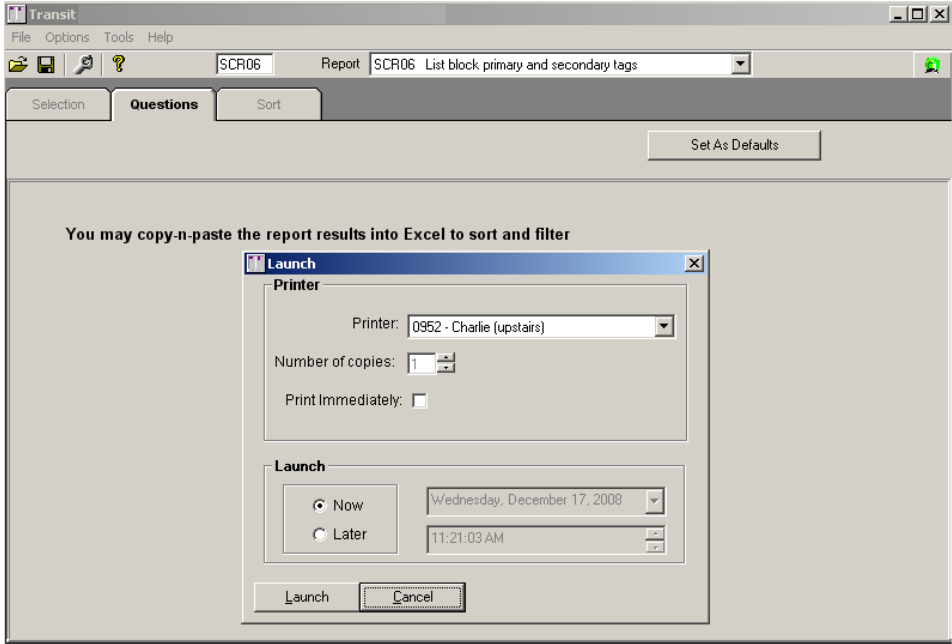


BlockID	Type	Value	Title	CatYrStart	CatYrStop	ParseStatus
RA001361	AWARD	APTS	(CUNY) APTS - Aid for Part-Time Study	1948	9999	OK
RA001371	AWARD	APTS	APTS - Aid for Part-Time Study	1962	2015	OK
RA001320	AWARD	PELL	PELL Award	1948	9999	OK
RA001358	AWARD	PELL	(CUNY) PELL Grant	1948	9999	OK
RA001372	AWARD	PELL	PELL Grant	1962	2015	OK
RA001362	AWARD	PTAP	(CUNY) PTAP - Part-time TAP	1948	9999	OK
RA001373	AWARD	PTAP	PTAP - Part-time TAP	1962	2015	OK
RA001321	AWARD	STAFFORD	STAFFORD Award	1948	9999	OK
RA001356	AWARD	STAFFORD	STAFFORD Award2	1948	9999	OK
RA001360	AWARD	TAP	(CUNY) TAP - Tuition Assistance Program	1948	9999	OK
RA001374	AWARD	TAP	TAP - Tuition Assistance Program	1962	2015	OK
RA001359	AWARD	VALLONE	(CUNY) VALLONE Scholarship	1948	9999	OK
RA001375	AWARD	VALLONE	VALLONE Scholarship	1962	2015	OK
RA001267	COLLEGE	SCI	Major in Mathematics	19941995	99999999	OK
RA001269	COLLEGE	SCI	Major in Mathematics	20042005	99999999	OK
RA001268	COLLEGE	SCI	Major in Mathematics	20102011	99999999	OK
RA001367	CONC	108	UGA Conc Example	19701971	99999999	OK
RA000600	CONC	ACCT	Accounting Option - BS in Business Administration	19941995	19951996	OK
RA000203	CONC	ACCT	Accounting Option - BS in Business Administration	19961997	19992000	OK
RA000506	CONC	ACCT	Accounting Option - BS in Business Administration	20002001	20022003	OK
RA001196	CONC	ACCT	Accounting Option - BS in Business Administration	20022003	99999999	OK

## SCR06 – List block primary and secondary tags

You may want to get a list of all of the blocks showing their primary and secondary tags. In doing this you may easily see which blocks use a Program or Student ID secondary tag for example. You may use the SCR06 job in Transit to get a report listing all of your blocks – sorted by the block type and value. Being there is so much data delivered in the report it is best to copy the data into Excel for sorting and viewing. This job makes use of the **dapreq2ndlist** script on the server.

No questions appear in Transit when SCR06 is chosen.



The report always lists all blocks.

BlockID	Type	Value	Title	CatYrStart	CatYrStop	ParseStatus	School	Degree	Coll.
RA001361	AWARD	APTS	(CUNY) APTS - Aid for Part-Time Study	1948	9999	OK			
RA001371	AWARD	APTS	APTS - Aid for Part-Time Study	1962	2015	OK			
RA001320	AWARD	PELL	PELL Award	1948	9999	OK			
RA001358	AWARD	PELL	(CUNY) PELL Grant	1948	9999	OK			
RA001372	AWARD	PELL	PELL Grant	1962	2015	OK			
RA001362	AWARD	PTAP	(CUNY) PTAP - Part-time TAP	1948	9999	OK			
RA001373	AWARD	PTAP	PTAP - Part-time TAP	1962	2015	OK			
RA001321	AWARD	STAFFORD	STAFFORD Award	1948	9999	OK			
RA001356	AWARD	STAFFORD	STAFFORD Award2	1948	9999	OK			
RA001360	AWARD	TAP	(CUNY) TAP - Tuition Assistance Program	1948	9999	OK			
RA001374	AWARD	TAP	TAP - Tuition Assistance Program	1962	2015	OK			
RA001359	AWARD	VALLONE	(CUNY) VALLONE Scholarship	1948	9999	OK			
RA001375	AWARD	VALLONE	VALLONE Scholarship	1962	2015	OK			
RA001267	COLLEGE	SCI	Major in Mathematics	19941995	99999999	OK		BS	SCI
RA001269	COLLEGE	SCI	Major in Mathematics	20042005	99999999	OK		BS	SCI
RA001268	COLLEGE	SCI	Major in Mathematics	20102011	99999999	OK		BS	SCI
RA001367	CONC	108	UGA Conc Example	19701971	99999999	OK			
RA000600	CONC	ACCT	Accounting Option - BS in Business Administration	19941995	19951996	OK			
RA000203	CONC	ACCT	Accounting Option - BS in Business Administration	19961997	19992000	OK			
RA000506	CONC	ACCT	Accounting Option - BS in Business Administration	20002001	20022003	OK			
RA001196	CONC	ACCT	Accounting Option - BS in Business Administration	20022003	99999999	OK			
RA000681	CONC	ACT	Acting/Directing Option - Major in Theatre Arts	19941995	19951996	OK			
RA000197	CONC	ACTG	Acting Option - Major in Theatre Arts	19961997	19992000	OK			
RA000561	CONC	ACTG	Acting Option - Major in Theatre Arts	20002001	99999999	OK			
RA000640	CONC	ADLT	Adult Development Option - Major in Human Dev	19941995	19951996	OK			

The secondary tags listed are School, Degree, College, Major1, Major2, Conc, Minor, Libl, Spec, Program, and Student ID, and can be viewed by using the horizontal scroll bar.

## SCR07 – List block text from Scribe

You can get the block text for your Scribe blocks based on a particular catalog year so you may save them as a backup or view on your PC in bulk. You may use the SCR07 job in Transit to get a report listing all of your blocks with all tags and text lines – sorted by the block type and value. This job makes use of the **dapblocksget** script on the server.

Transit prompts the user for a catalog year and a block type on the questions tab. This tool will find blocks whose catalog year range encompasses the catalog year specified for the block type specified. The script checks to see if the starting catalog-year is less than or equal to this value and whose ending catalog-year is greater than or equal to this value. Normally users should select the current catalog year since they don't want old blocks.

File Options Tools Help

SCR07 Repo SCR07 List block tags and text

Selection Questions Sort

Set As Defaults

Get the block text for this catalog year [ ]

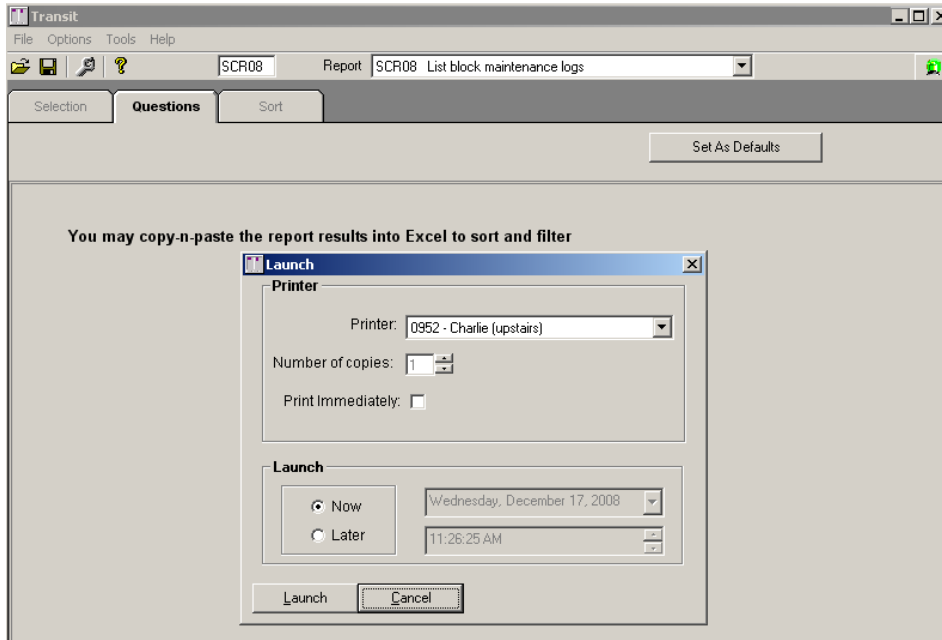
Get the block text for this block type \* MAJOR [ ]

The resulting report file may be very large and thus will take a long time to open. Since the resulting report can be very large, the block type is a required field to help reduce the size of any one report listing. You may want to run one report for DEGREE blocks, another for MAJOR blocks and so on.

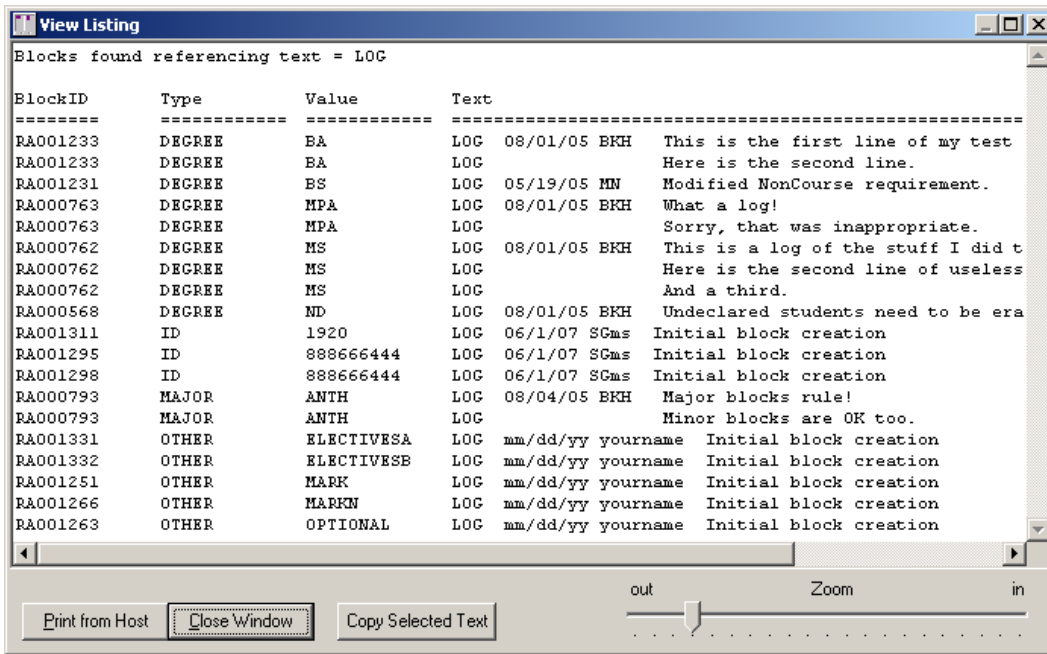
## SCR08 – List LOG entries from Scribe text

You can get a list of all of the LOG entries in your Scribe text so you may review what has changed in your blocks. You may use the SCR08 job in Transit to get a report listing all of your blocks with LOG text lines – sorted by the block type and value and text sequence number. This job makes use of the **dapreqgrep** script on the server but it only looks for lines with LOG in column 1-3 of the text line.

No questions appear in Transit when SCR08 is chosen.



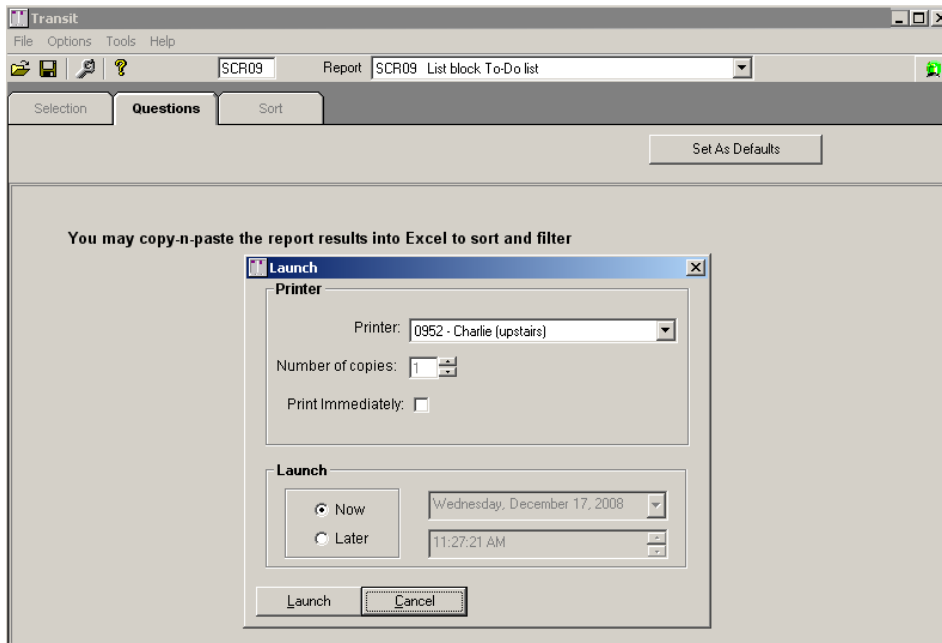
The report always lists all text lines from requirement blocks containing LOG starting in column 1.



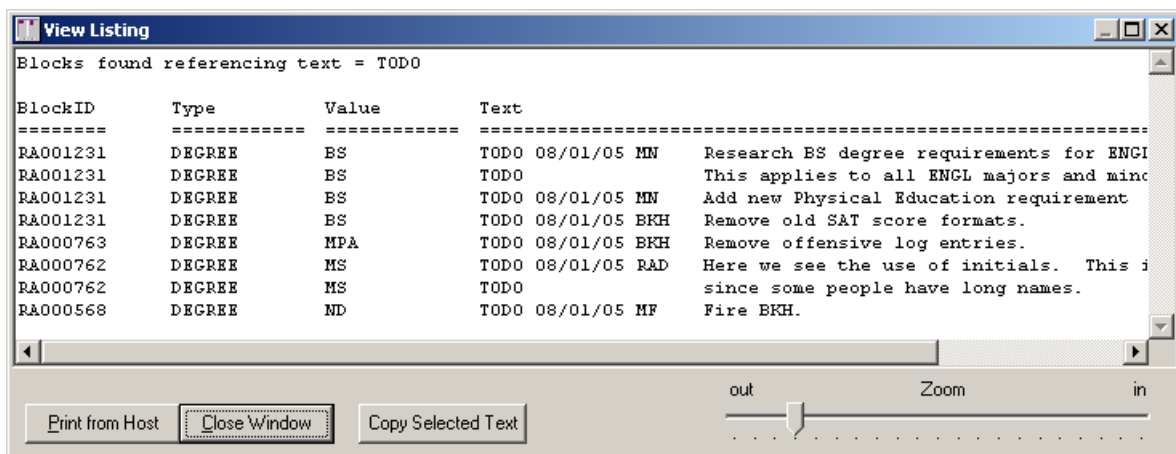
## SCR09 – List TODO entries from Scribe text

You can get a list of all of the TODO entries in your Scribe text so you may review what work still needs to be done. You may use the SCR09 job in Transit to get a report listing all of your blocks with TODO text lines – sorted by the block type and value and text sequence number. This job makes use of the **dapreqgrep** script on the server but it only looks for lines with TODO in column 1-4 of the text line.

No questions appear in Transit when SCR09 is chosen.



The report always lists all text lines from requirement blocks containing TODO starting in column 1.



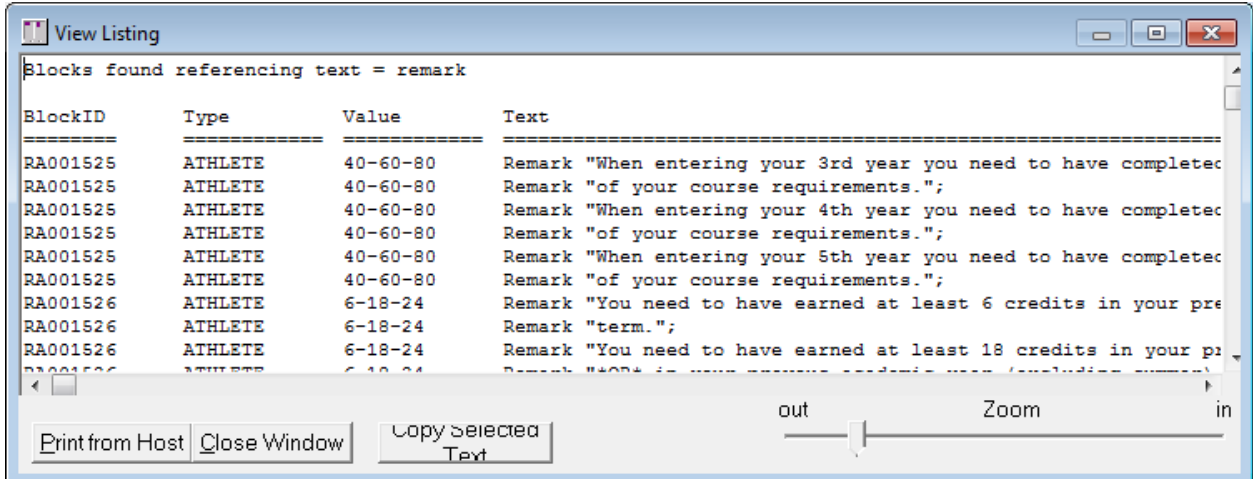
## SCR10 – Find blocks where this text is referenced

You may need to find the blocks that make use of remarks or proxy-advice or use certain values. You can do this through the SCR10 job – this job makes use of the **dapreqgrep** script but you only need access to Transit and not the command prompt.

When running the job you need to enter a text string. Remember that this is similar to doing a Find in Notepad with the “Match case” option not selected; the search is not case sensitive. If you type “remark” it will find “Remark” and “REMARK” and any other instances of the remarks.



When the job runs, the output will be sent to a Report file. Click on the View Jobs button in Transit to view the report created. The report will list the text lines containing the specified string in addition to the block information.



## SCR11 – Find-and-Replace block text

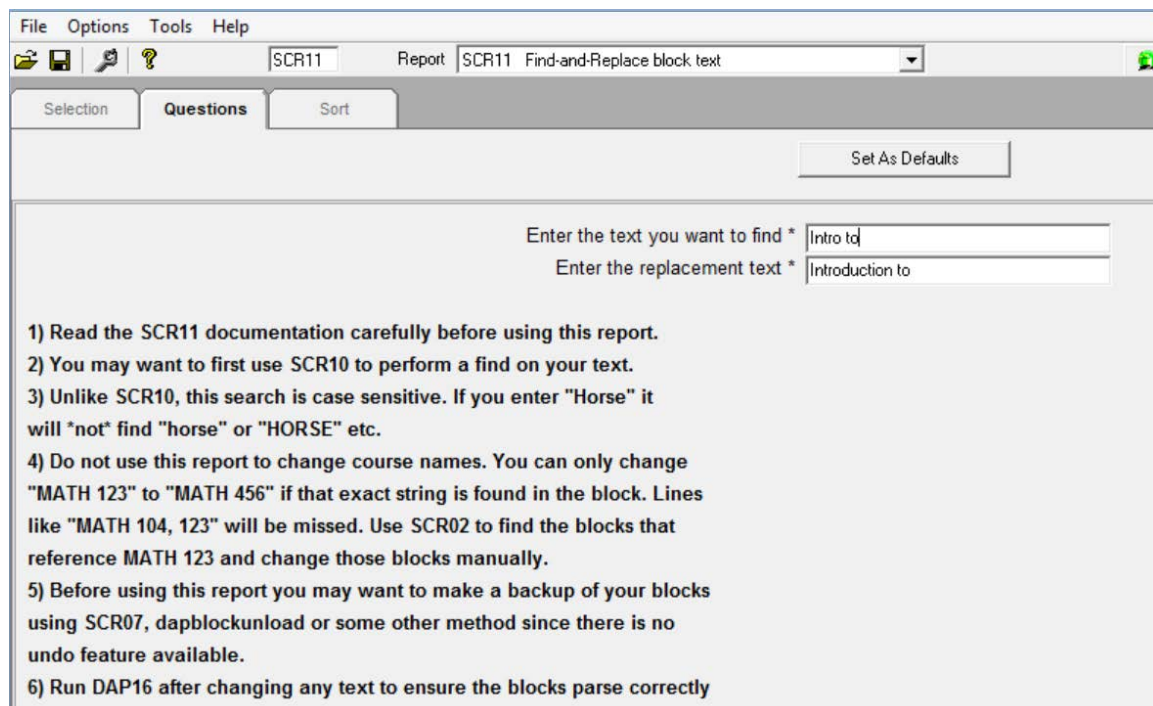
This very powerful report allows you to find any string of text up to 30 characters and replace it with a new string. The new text can be shorter or longer than the old text but the new text is also limited to 30 characters. This report makes use of the **dapreqsearchnreplace** script but you only need access to Transit and not the command prompt.

Before running SCR11 you should first run SCR10 to find the text that you want to replace to understand what will happen when you run SCR11. However, if you search for “zebra” in SCR10 it will find all lines with “zebra” and “Zebra” and “ZEBRA” because SCR10 is case insensitive. SCR11, on the other hand, is case sensitive so if you search for “zebra” it will only find the lowercase occurrences of that word. You may have to run SCR11 again for “Zebra” and then again for “ZEBRA” if you want to replace all zebras with “Horse”, let’s say.

This report searches for the text exactly as you see it in Scribe. This report should not be used to change names of courses. For example, if you have a rule of “1 Class in **HIST 102, 106**” you cannot use this report to change “HIST 106” to something else because “HIST 106” would not be found. This find-and-replace does not understand the Scribe language and thus do not think it will do any more than what the find-and-replace does in Notepad.

Because this is such a powerful tool it is possible you will end up replacing too much text or the wrong text – even if you study the results of SCR10 first. For this reason you should make a backup of your blocks using SCR07 and/or dapblockunload (available under the ADMIN report), in addition to your normal database backup, just so you have a way to undo your work in case of a bad mistake. Bottom line: there is no “undo” button in this report; if you make a mistake you will need to restore from a saved backup or fix the mistakes manually.

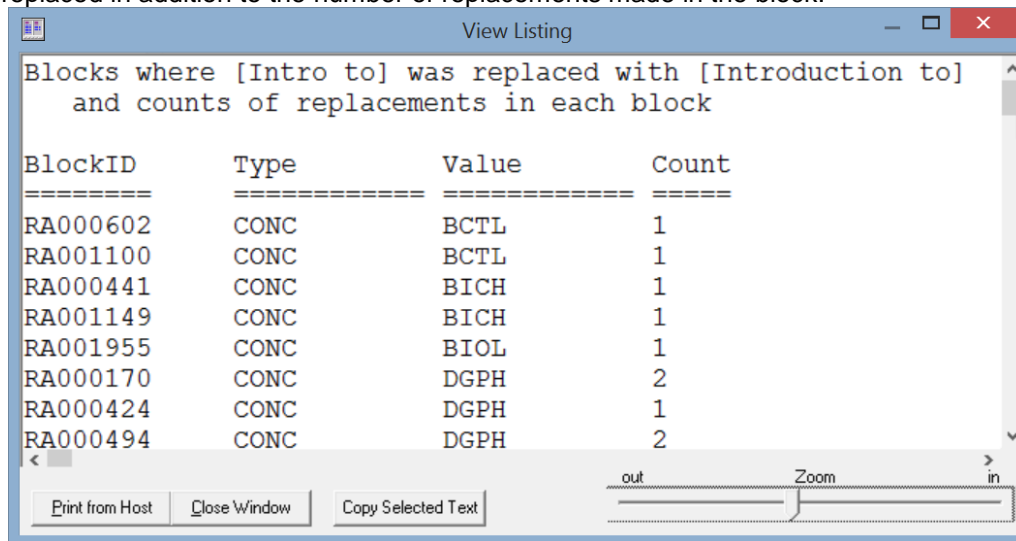
You should run DAP16 after making changes using SCR11 to 1) ensure your blocks parse successfully and 2) to allow new audits to pull in your changes.



The screenshot shows the SCR11 Find-and-Replace block text report interface. The window title is "SCR11" and the report name is "SCR11 Find-and-Replace block text". The interface includes a menu bar (File, Options, Tools, Help), a toolbar with icons for file operations, and a "Report" dropdown menu. Below the menu bar are tabs for "Selection", "Questions", and "Sort", with "Questions" selected. A "Set As Defaults" button is located below the tabs. The main area contains two input fields: "Enter the text you want to find \*" with the value "Intro tq" and "Enter the replacement text \*" with the value "Introduction to". Below the input fields is a list of instructions:

- 1) Read the SCR11 documentation carefully before using this report.
- 2) You may want to first use SCR10 to perform a find on your text.
- 3) Unlike SCR10, this search is case sensitive. If you enter "Horse" it will \*not\* find "horse" or "HORSE" etc.
- 4) Do not use this report to change course names. You can only change "MATH 123" to "MATH 456" if that exact string is found in the block. Lines like "MATH 104, 123" will be missed. Use SCR02 to find the blocks that reference MATH 123 and change those blocks manually.
- 5) Before using this report you may want to make a backup of your blocks using SCR07, dapblockunload or some other method since there is no undo feature available.
- 6) Run DAP16 after changing any text to ensure the blocks parse correctly

When the job runs, the output will be sent to a Report file. Click on the View Jobs button in Transit to view the report created. The report will list the blocks containing the string that was found and replaced in addition to the number of replacements made in the block.



## SCR9X – Prerequisite checking reports

The SCR9x suite of reports are documented in the *Technical Guide* in the *Prerequisite Checking* section.



