ellucian™

Banner Student
TRM Supplement

Release 8.6
October 2013

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

Revision History

Publication Date	Summary
October 2013	New version that supports Banner Student 8.6 software.

Contents

1 Reports and Processes

Database Reports

C Reports

Report/Module Name	Full Name	Description
Catalog		
SCRBULT	Bulletin Report	Used to print the catalog of courses with detail information including text, prerequisites, corequisites, etc.
SCRRIMS	Course Catalog Extract	Used to extract course data and provide it in an XML format readable to third parties using IMS standards.
Schedule		
SSPMFEE	Section Fee Population Process	Examines course/section characteristics defined in SSADFEE**,** determines applicable sections, and inserts appropriate records in the SSRFEES table.
SSPRDEF	Schedule Open Learning Rule Default Process	Defaults open learning rules for sections, as defined on SOAORUL, for CRNs that have been created after the rules have been established.
SSPSCHD	Schedule Purge	Purges the schedule information.
SSRASTZ	Available Seats to Zero Process	Used to modify the available seats for a CRN by setting the number of available seats to zero.
SSRATSQ	Room Attribute Sequence Update Process	Used to automatically populate the sequence number of the room attribute information in the STVRDEF table.
SSRROLL	Term Roll	Rolls Sections of one term to another term.
SSRSECT	Class Schedule	Lists all sections for given term and/or part of term.
SSRSCMT	Update Building/Room Process	Reads the Scheduler Work Table (SSTSCHW) entries and updates the SSRMEET table with the room assignments generated in Schedule25

Report/Module Name	Full Name	Description
SSRSCPR	Schedule Work Table Purge Process	Used to delete Scheduler Work Table (SSTSCHW) data based on the term and/or campus specified in the parameters.
SSRSCRM	Schedule 25 Work File Creation Process	Used to create the building, room, partition, department (subject), class descriptor, and control files required to run the scheduling tool.
SSRSCUP	Scheduler Work Table Update Process	Reads the three Schedule25 export files (`sortdp.dat`, `losers.dat`, `notposs.dat`),updating the existing records in the Scheduler Work Table (SSTSCHW).
SSRTALY	Scheduled Section Tally	Produces detailed list of section information.
General Person		
SPRPDIR	Person Directory	Prints all persons by type (recruit, applicant, student, faculty).
Support Services		
SERLOAD	Support Services Load Process	Used to create service groups from a user-defined population selection.
Faculty Load		
SIPASGN	Faculty Load Purge	Purges instructional, non-instructional, and workload rules for a term.
SIRASGQ	Faculty Schedule Report	Used to print an instructor's schedule, with detailed information.
SIRCTAL	Faculty Load Contract Analysis Report	Used to calculate the Faculty Load contract analysis.
SIRTRAL	Faculty Load Term Analysis Report	Used to calculate and print the Faculty Load term analysis.
Location Management		
SLPHOUS	Housing Purge	Purges housing applications, housing, meal plan, and phone assignments for the user specified terms and activity dates.
SLRBACS	ACS Housing Interface	Produces file for ACS HIS for addresses of housing applicants.

Report/Module Name	Full Name	Description
SLRDADD	Dormitory Address Creation	Used to create dormitory addresses from the housing assignments.
SLRFASM	Batch Room/Meal/Phone Assessment	Used to assess room, meal, and phone charges in a batch mode.
SLRHLST	Active Housing Assignments Report	Lists all the active housing assignments.
SLRROLL	Assignment Roll Process	Used to roll the term's housing, meal, and phone assignments forward to a future term.
SLRSCHD	Batch Scheduler Report	Used to create housing assignments in batch based on the applicant's preferences.
Recruiting		
SRREMAL	Electronic Prospect Email Process	Used for sending out a mass mailing as an email letter to electronic prospects.
SRRENRH	Rec/Enroll Analysis - How Learned Report	Produces a statistical report of recruits, applicants, and enrollees by a How I Learned code for a term range.
SRRENRL	Recr/Enrl Analysis - Source/Recr Report	Produces a statistical report of recruits, applicants, and enrollees by source and recruiter for a term range.
SRRINQR	Recruits Never Applied to Inst. Report	Prints or deletes all recruits that expressed an interest in the institution but never applied.
SRRPREL	Migrate Electronic Prospects Process	Used to create or update Banner® recruiting records for the records in the Search Tape View (SRVPREL).
SRRRICC	Recruiter Integration Crosswalk Process	Generates academic programs names from existing curriculum data using a defined business rule.
SRRRPRO	Recruiter Prospect Provisioning Process	Used to provision Recruiter with Banner prospects, for further processing in Recruiter.
SRRRVAL	Recruiter Validation Provisioning Process	Used to provision Recruiter with validation codes from Banner to ensure the same validation codes are used in the Recruiter system as in Banner when possible.
SRRSRIN	Prospect Tapeload Matching Process	Used to determine if a match exists between a record on the Search Tape View (SRVPREL) and Banner production data when loading search tape records en masse.

Report/Module Name	Full Name	Description
SRTLOAD	Electronic Prospect Load	Loads data from an input file to temporary tables. Also creates an audit report detailing the status of each record on the input file.
SRTPURG	Electronic Prospect Purge	Used to delete records from the search tape temporary tables and designate which records to purge.
Admissions		
SAPADMS	Admission Purge Report	Purges all the Admission records for a selected term.
SARACTM	Admissions Count by College/ Major	Prints admission application count by college/major.
SARADMS	Admissions Application Report	Prints admissions application report by ID or name and term, application number.
SARAMDP	AMCAS Date Purge	Used to purge the flat file of data that was electronically submitted to AMCAS.
SARAMXF	AMCAS Extract File Process	Used to create a flat file of data to be electronically submitted to AMCAS.
SARBDSN	Admissions Decision Calculation Report	Produces batch automated admissions decision.
SARDCBT	Rating Audit Report	Used to view applications that have been updated on the SAADCBT for a specific date or dates.
SARDCSN	Admissions Decision Criteria Report	Prints admissions decision criteria report and term roll with delete option.
SAREMAL	Batch Email for Elec. Apps. Process	Used for sending out a mass mailing as an email letter to electronic applicants.
SARETMT	Elec App Verify/Load Process	Used for matching electronic applicants to Banner production records by interface code, and SSN and Last Name, or all criteria.
SARETPG	Electronic App Purge Process	Used to purge old data from the electronic application holding tables by date range, application type, source, and term.
SAR189U	TS189 Upload to Banner	Used to receive an electronic (EDI) admissions application in SPEEDE format and to automate the entry of the admissions application data into the Banner Admissions module.

Report/Module Name	Full Name	Description
SARRATE	Admission Calculation Rating Report	Used to calculate the admissions rating for an ID or population.
General Student		
SGPCOOP	Cooperative Education Purge	Purges all the cooperative education data for all students.
SGPBLCK	Student Block Load Process	Associates a group of students defined through population selection to a block code for an effective term.
SGPHOLD	Hold Purge	Purges all expired holds which exist on the database.
SGPSTDN	General Student Purge	Purges the general student records for any non-registered students.
SGRCHRT	Cohort Load Process	Assigns cohort codes to a group via population selection.
SGRKNOW	Student Right to Know Report	Produces graduation/completion rates by cohort and sport for race and gender.
SGRSTDN	Student Report	Prints student list by term in alpha or ID order.
SGRVETN	Veteran Report	Prints veteran list by term.
Registration		
SFPAGRD	Auto Grade Assignment Process	Used to assign an incomplete or institutionally defined letter grade to long-standing, ungraded, open learning registration records.
SFPBLCK	Course Request Load Process	Tracks student populations by block schedule codes for effective and report terms.
SFPENRL	Enrollment Verification Request Purge	Used to purge all previously requested enrollment verification requests.
SFPFAUD	Purge Fee Assessment Audit Process	Used to purge audit history records from the database.
SFPREGS	Registration Purge	Purges all registration records for a selected term, as well as time status history.
SFPWAIT	Waitlist Enrollment Purge	Purges waitlist enrollment.

Report/Module Name	Full Name	Description
SFPWLRO	Waitlist Priority Reorder Process	Used to reorder positions in the waitlist queue based on the rules defined on SOAWLTC.
SFRBWLP	Batch Waitlist Notification Process	Used to produce waitlist notifications by term for new available seats based on entries in the SFRCOLW collector table.
SFRENRL	Enrollment Verification Request	Produces the pre-processed enrollment verification requests.
SFRFASC	Registration Fee Assessment Process	Used to run batch fee assessment.
SFRFEES	Fee Assessment Report	Used to assist in trouble shooting and debugging fee assessment processing.
SFRFFMN	Feedback Monitor Students Process	Used to find students registered for a course in a specific term that are required to be monitored by faculty members.
SFRFFPG	Faculty Feedback Purge Process	Used to purge records for a term and session description for the start and end dates of the session control record.
SFRGEED	Gainful Employment Submittal Process	Used to retrieve data for gainful employment reporting based on processing rules.
SFRHCNT	Unduplicated Headcount Report	Prints headcount totals by level and major for a term, for students with a registration status and effective headcount activated.
SFRLINK	Unsatisfied Links Report	Produces a list of students who have unsatisfied or missing section links for a term.
SFRNSLC	Clearinghouse Extract Report	Extracts student enrollment information for reporting to the National Student Loan Clearinghouse (NSLC).
SFRRGAM	Registration Admin Messages Report	Tracks registration errors for administrative purposes, checks for in-progress prerequisites errors, views any potential schedule conflicts due to changes in class schedule, finds students who took in-progress prerequisites that may have not met the requirements.
SFRRNOP	Registered, Not Paid Process	Prints/purges all students registered but not paid for a term and their time status history.
SFRSCHD	Student Schedule	Prints student schedule for term.

Report/Module Name	Full Name	Description
SFRPINI	Compliance Listener Start Up Process	Initializes the SFRPIPE process to run in the background, listening for Oracle pipe calls to execute compliance for registration prerequisite processing.
SFRPIPE	Compliance Pipe Process	Used as a listening agent for Oracle pipes, to initiate compliance to perform prerequisite registration processing.
SFRSLST	Class Roster	Prints class list for term or part of term.
SFRSSCR	NSLDS SSCR Process	Reads and processes NSLDS Student Status Confirmation Report (SSCR) Roster and Error Notification files.
SFRTMST	Time Status Calculation Update Process	Calculates student enrollment time statuses in batch mode and updates/inserts time status history records.
Academic History		
SHPTAEQ	Transfer Articulation Purge	Used to purge transfer articulation information.
SHPTRTC	Academic Transcript Request Purge	Used to purge the academic transcript request information.
SHRASTD	Calculate Academic Standing Report	Used to calculate academic standing for a selected term and group of students.
SHRCATT	Commencement Attendance Report	Used to print ceremony type and attendees, and cap, gown, hood order, pickup, and return information.
SHRCGPA	Calculate GPA Report	Used to calculate GPA for a selected term and group of students.
SHRCINC	Incomplete Grade Process	Used to automatically update incomplete grades to final grades.
SHRCOMM	Commencement Report	Prints camera-ready commencement program report.
SHRCONV	Calculate GPA Conversion Process	Calculates and stores campus GPAs calculated by term and level of the students.
SHRDEGS	Degree Status Update Report	Updates the existing degree status code to a new user-defined degree status.
SHRDEGV	Degree Verification Process	Used to collect degree data for a student at an institution for the NSC.

Report/Module Name	Full Name	Description
SHREDII	Electronic Data Interchange Institutions	Used to upload EDI Capable Institution file for matching of FICE codes.
SHREDIP	Upload of EDI Transcript	Used to load transcript data into interim Oracle tables in Banner, after the FLAT130 file produced by EDI.Smart has been uploaded to the host where Banner resides.
SHREDIR	Electronic Data Interchange Reconciliation	Used to reconcile EDI.Smart disposition files with transcript requests in the collector table.
SHREDIY	Electronic Data Interchange Extract	Used to produce Banner transcripts to be sent via EDI.Smart. Reads requests from the collector file and generates the flat file entries for the transcripts along with control and log file entries.
SHRETRP	Electronic Transcript Upload Purge Process	Used to purge electronic transcript data, after electronic transcripts have been processed in the Banner Student System.
SHRGPAC	GPA Recalculation Report	Used to calculate GPA for selected term for student or population selection.
SHRGRDE	Grade Mailer	Prints grade mailers by type (original, revised, and duplicate).
SHRIACT	IPEDS Total Activity Report	Prints institution's total student activity, required by Dept. of Education.
SHRIAGE	IPEDS Summary by Age Report	Used to produce the summary by age portion of the Fall Enrollment Reports.
SHRICIP	IPEDS Completion Report	Prints parts A, B, C, and D comprising IPEDS Completion Report.
SHRIETH	IPEDS Ethnic Status Report	Prints report by racial/ethnic category, attendance period, level, and sex of student.
SHRIGRS	Graduation Rate Survey Report	Tracks graduation, transfer, and withdrawal rates by cohort and with athletic aid by type of institution for IPEDS reporting.
SHRIPDS	IPEDS File Generation Process	Produces generation extract file, which is used to produce the Fall Enrollment Reports.
SHRIRES	First Time Residency Report	Prints a report on first time student's states of residency.

Report/Module Name	Full Name	Description
SHRPREV	Progress Evaluation Process	Used to determine progress evaluation and combined academic standing codes and print a report of the results.
SHRROLL	Grade Roll Process	Rolls grades entered on the Class Roster Form and creates Academic History.
SHRROUT	Roll Learner to Outcome Process	Used to roll learner curriculum records to the outcome curriculum and then insert new degree sequence records.
SHRRPTS	Repeat/Equivalency Course Check	Used to flag repeated courses.
SHRTAEQ	Transfer Equivalency Worksheet	Used to review the transfer equivalency for a student.
SHRTECA	Transfer Equivalency Catalog	Provides a listing of transfer courses entered for each institution.
SHRTPOP	Transcript Population Creation Process	Used to select a population from the student body for whom transcripts are to be printed.
SHRTRTC	Academic Transcript	Used to produce an academic transcript.
SHRTYPE	Student Type Update	Used to update the student type after completing the Grade Roll Process.
CAPP		
SMPCPRG	Compliance Purge Process	Used to purge compliance records.
SMPCSAP	SAP Purge Process	Used to purge compliance data for satisfactory academic progress.
SMRBCMP	Batch Compliance Process	Creates compliance and hardcopy requests from population, or performs outstanding compliance requests and produces outstanding hardcopy output. Calls *both* SMRCMPL and SMRCRLT.
SMRCMPL	Program Compliance Report	Checks courses taken by student against curriculum and program requirements.
SMRCRLT	Compliance Print Process	Prints hardcopy compliance output.

Report/Module Name	Full Name	Description
SMRRLST	Compliance Rule Report	Produces a list of compliance rules that have been set up at your institution for rule within a rule for areas (SMAAREA) and groups (SMAGROP).
Overall		
SOPAPPT	Appointment Purge Process	Purges all appointment records for specified start and end dates.
SOPLCCV	Learner Curriculum Conversion Process	Used to read the input population and run the curriculum conversion process.
SOPLCPG	Learner Curriculum Purge Process	Used to purge SORLCUR rows and/or SORLFOS rows for Admissions and Recruiting records.
SOPMAUD	Purge Mass Entry Audit Process	Used to purge mass entry audit information displayed in SOAMAUD and keep the SORMAUD and SOBMAUD table sizes from getting too large.
SOPSATS	SAT Recentering Process	Recenters SAT scores for college board compliance.
SORAINF	Administrator Assignments Process	Allows institutions to use the rules defined on SOAADAS to assign administrators to recruit and applicant records.
SORCPLN	Communication Plan Processing Report	Shows person's movement from old to new communication plan.
SOREMAL	Student Email Process	Used for sending out a mass mailing as an email letter using population selection on IDs.
SORHSRP	Admissions High School Report	Prints totals and percentage for applicants, students, enrollment and GPA.
SORLCHG	Non-Destructive Curric Update Report	Used to show non-destructive updates made to curriculum and/or field of study records.
SORMEBP	Process Mass Entry Report	Used to execute mass entry updates that have been held for job submission processing using the Results window of a mass entry form.
SORSBSM	Source/Background Summary Report	Used to collect source/background institution summary information similar to what is found on SOASBSM.
SURDELT	Communication Removal Process	Used to perform a mass removal of mail records.

Report/Module Name	Full Name	Description
SURLOAD	Communication Load Process	Used to perform a mass entry of mail records.
Course Request and Scheduling		
SFPCREQ	Course Request Edit Report	Lists course request transactions containing errors, with error messages, and updates valid transactions for course request processing.
SFPFREQ	Course Request Update Report	Used to load course request and schedule data to registration tables and to update section counts in Banner Student System.

COBOL Reports

Report/Module Name	Full Name	Description
Location Management		
SLRSCHE	Batch Housing Schedules Report	Processes housing assignments based on information on SLARMAP.
Course Request and Scheduling		**Note:** C% and D% reports for Course Request and Scheduling are generated by SCT% reports.
C1501	Course Schedule Load Control Report	Lists input/output file counts in program Load Course Database to Disk.
C2001	Course Request Update Control Report	Produces printer listing of file, transaction, and summary counts.
C2002	Course Request Audit Trail	Produces report of students with request set errors.
C3001	Print Requests Control Report	Produces printer listing of file counts in program Print Course Request Reports.
C3002	Section Request Tally/ Closed Section Tally	Lists sections on Reformatted Course Interface File and students requesting each section. Use to measure student demand for courses and specific sections.
C3003	Section Request List	Lists sections on Reformatted Course Interface File and students requesting each section. Use for quick identification of students requesting specific sections, and students in sections which are to be canceled or have restricted enrollment.

Report/Module Name	Full Name	Description
C3004	Student Request List	Lists request set of each student prior to scheduling and any errors found.
C3005	Request Segregated Error List	Produces printer listing of each registering student and course requests for that student with specific type of selected error in their request set.
C3007	Acknowledgment of Pre-Registration	Lists all section requests and error messages for pre-registered students.
C3008	Valid Requested Hours	Lists registering students and valid requests hours.
D0601	Student Scheduling Control Report	Lists summary totals and results of current Student Scheduling Cycle for evaluation of scheduling run.
D0603	Scheduling Results Matrix	Identifies counts of students by number of requests submitted versus number of requests satisfied for evaluation of scheduling run.
D3001	Print Scheduling Control Report	Produces printer listing of file counts in Print Scheduling Reports Program.
D3002	Scheduling Enrollment Tally	Lists capacity, total requests, and enrolled counts for each section, as a result of scheduling, for scheduling trial run modifications and/or late registration.
D3003	Section Enrollment List/ Unscheduled Student By Section	Lists each section in Course Master File. Use as preliminary class list for Section Enrollment List. Use as wait list for drop and add for Unscheduled Student By Section.
D3008	Student Schedule Analysis	Lists counts of students by class level within major for evaluation of schedule type.
SCTC1000	Course Interface Extract	Used to extract course schedule data from Banner Student System.
SCTC1500	Course Schedule Load	Creates tables with restriction and relative course information for each section extracted from Banner Student System.
SCTC2000	Course Request Update	Used to begin Course Request Process and to produce audit trail/Change Register for request validation and error detection.
SCTC3000	Course Request Extract	Used to generate other reports for Course Request (edit mode) and Scheduling (update mode).

Report/Module Name	Full Name	Description
SCTD0600	Student Schedule Update	Used to begin student scheduling.
SCTH1000	Person Interface Extract	Used to extract course request and bio/demo data from Banner Student System.
SADASORT	CRS Sort Program	Used to sort files on specified order.
SCTSORT	CRS Sort Program	Used to sort files on specified order.

Oracle Report

Report/Module Name	Full Name	Description
Admissions		
SARETBL	Electronic Application Report	Used to view information about a student's electronic admissions application.

Java Reports

Report/Module Name	Full Name	Description
Catalog		
SCRCATE	Course Catalog Data Extract Process	Used to extract course catalog data and create an XML output file for download or third party use.
Academic History		
SHRPESE	PESC/XML Transcript Export Process	Used to create electronic transcripts in XML format by producing a `.xml` file that can be read by the receiving institution.
SHRPESI	PESC/XML Transcript Import Process	Used to import XML transcript files into an institution from other institutions.
SHRTCIM	Transfer Catalog Data Import Process	Used to import an XML extract file of course catalog data into a Banner database and compare incoming records with transfer course records in the system.

Report and Process Attributes Matrix

Report and Process Attributes Legend

Item	Description
Report or Process	The report/batch process name.
Language	Identifies the language for the process - COBOL, C, RPT, SQL, or PL/SQL, etc.
Update/Query	Does the process update any tables, or is it strictly a query-only report?
Audit	Can you run the update process in Audit Mode, so that you can produce the report without an update taking place (Yes or No)? **Note:** *Yes* appears in this column only if the process permits both update and audit mode. If the report is query only, *Yes* does not appear in this column.
Job Submission	Can you run the process via job submission (Yes or No)?
Sleep/Wake	Is the process used in conjunction with Sleep/Wake (Yes or No)?
Off Peak	Does the company recommend that you defer this program to an off peak processing time (late night, weekends) for performance reasons (Yes or No)?
Restart	If the process aborts or is terminated after the process is initiated, are special procedures required to restart the process without any adverse consequences (Yes or No)? **Note:** *Yes does not appear in this column if the job can be restarted without special procedures.* If *Yes* appears, refer to the "Restart" section of this chapter for more information regarding recovery procedures.

Report and Process Attributes

Banner Student System

Report or Process	Language	Update/ Query	Audit	Job Sub	Sleep/ Wake	Off Peak	Restart
SAPADMS	C	Yes	Yes	Yes			Yes
SARACTM	C			Yes			
SARADMS	C			Yes			
SARAMDP	C	Yes	Yes	Yes			
SARAMXF	C	Yes	Yes	Yes			
SARBDSN	C	Yes		Yes			Yes

Report or Process	Language	Update/ Query	Audit	Job Sub	Sleep/ Wake	Off Peak	Restart
SARDCBT	C			Yes			
SARDCSN	C	Yes		Yes			Yes
SAREMAL	C			Yes			
SARETBL	Oracle			Yes			
SARETMT	C	Yes	Yes	Yes			
SARETPG	C	Yes	Yes	Yes			
SARRATE	C	Yes	Yes	Yes			
SAR189U	C			Yes			
SCRBULT	C			Yes			
SCRCATE	Java	Yes		Yes			
SCRRIMS	C			Yes			
SCROIMS *	C						
SCRRFUN *	C						
** These are not stand-alone programs and cannot be run individually. They create object files which are linked to SCRRIMS.*							
SERLOAD	C	Yes		Yes			
SFPAGRD	C	Yes	Yes	Yes			
SFPBLCK	C	Yes		Yes			
SFPENRL	C	Yes	Yes	Yes			
SFPFAUD	C	Yes	Yes	Yes			
SFPREGS	C	Yes	Yes	Yes			Yes
SFPWAIT	C	Yes	Yes	Yes			
SFPWLRO	C	Yes		Yes			
SFRBWLP	C	Yes		Yes	Yes		
SFRENRL	C	Yes		Yes			

Report or Process	Language	Update/ Query	Audit	Job Sub	Sleep/ Wake	Off Peak	Restart
SFRFASC	C	Yes	Yes	Yes		Yes	
SFRFEES	C			Yes			
SFRFFMN	C			Yes			
SFRFFPG	C	Yes		Yes			
SFRGEED	C	Yes		Yes			
SFRHCNT	C			Yes			
SFRLINK	C			Yes			
SFRNSLC	C	Yes	Yes	Yes			
SFRPINI	C						
SFRPIPE	C						
SFRRGAM	C	Yes		Yes			
SFRRNOP	C	Yes		Yes			
SFRSCHD	C	Yes		Yes	Yes		
SRRSLST	C	Yes		Yes			Yes
SFRSSCR	C	Yes	Yes	Yes			
SFRTMST	C	Yes	Yes	Yes			
SGPBLCK	C	Yes		Yes			
SGPCOOP	C	Yes	Yes	Yes			
SGPHOLD	C	Yes	Yes	Yes			
SGPSTDN	C	Yes	Yes	Yes			
SGRCHRT	C	Yes		Yes			
SGRKNOW	C			Yes			
SGRSTDN	C			Yes			
SGRVETN	C			Yes			

Report or Process	Language	Update/ Query	Audit	Job Sub	Sleep/ Wake	Off Peak	Restart
SHPTAEQ	C	Yes	Yes	Yes			
SHPTRTC	C	Yes	Yes	Yes			
SHRASTD	C	Yes	Yes	Yes			
SHRCATT	C	Yes		Yes			
SHRCGPA	C	Yes		Yes			Yes
SHRCINC	C	Yes	Yes	Yes			
SHRCOMM	C			Yes			
SHRCONV	C	Yes		Yes		Yes	Yes
SHRDEGS	C	Yes	Yes	Yes			
SHRDEGV	C			Yes			
SHREDII	C	Yes	Yes	Yes			
SHREDIP	C	Yes		Yes			
SHREDIR	C	Yes	Yes	Yes			
SHREDIY	C	Yes		Yes			
SHRETRP	C	Yes	Yes	Yes			
SHRGPAC	C	Yes		Yes			Yes
SHRGRDE	C	Yes		Yes			Yes
SHRIACT	C			Yes		Yes	
SHRIAGE	C			Yes			
SHRICIP	C			Yes			
SHRIETH	C			Yes			
SHRIGRS	C			Yes			
SHRIPDS	C	Yes		Yes		Yes	
SHRIRES	C			Yes			

Report or Process	Language	Update/ Query	Audit	Job Sub	Sleep/ Wake	Off Peak	Restart
SHRPESE	Java			Yes	Yes		
SHRPESI	Java			Yes	Yes		
SHRPREV	C	Yes	Yes	Yes			
SHRROLL	C	Yes		Yes			Yes
SHRROUT	C	Yes	Yes	Yes			
SHRRPTS	C	Yes	Yes	Yes			Yes
SHRTAEQ	C			Yes			
SHRTCIM	Java	Yes	Yes	Yes			
SHRTECA	C			Yes			
SHRTPOP	C	Yes		Yes			
SHRTRTC	C	Yes		Yes	Yes	Yes	
SHRTYPE	C	Yes	Yes	Yes			
SIPASGN	C	Yes	Yes	Yes			Yes
SIRASGQ	C			Yes			
SIRCTAL	C			Yes			
SIRTRAL	C			Yes			
SLPHOUS	C	Yes	Yes	Yes			
SLRBACS	C	Yes	Yes	Yes			
SLRDADD	C	Yes		Yes			Yes
SLRFASM	C	Yes		Yes			Yes
SLRHLST	C			Yes			
SLRROLL	C	Yes	Yes	Yes			Yes
SLRSCHD	C	Yes	Yes	Yes			
SLRSCHE	COBOL	Yes		Yes			

Report or Process	Language	Update/ Query	Audit	Job Sub	Sleep/ Wake	Off Peak	Restart
SMPCPRG	C	Yes	Yes	Yes			
SMPCSAP	C	Yes	Yes	Yes			
SMRBCMP	C	Yes		Yes			Yes
SMREVAL ****	C						
SMRGEVL ****	C						
SMRJOBS ****	C						
SMRRULE ****	C						
SMRCMPL	C	Yes		Yes			
SMRSELS ****	C						
***** *These are not stand-alone programs and cannot be run individually. They create object files which are linked to SMRBCMP and SMRCMPL.*							
SMRCRLT	C			Yes			
SMRRLST	C			Yes			
SOPAPPT	C	Yes	Yes	Yes			Yes
SOPLCCV	C	Yes	Yes	Yes			
SOPLCPG	C	Yes	Yes	Yes			
SOPMAUD	C	Yes	Yes	Yes			
SOPSATS	C	Yes	Yes	Yes			
SORAINF	C	Yes	Yes	Yes			
SORCPLN	C	Yes		Yes			
SOREMAL	C	Yes	Yes	Yes			
SORHSRP	C			Yes			
SORLCHG	C			Yes			
SORMEBP	C	Yes	Yes	Yes			
SORSBSM	C			Yes			

Report or Process	Language	Update/ Query	Audit	Job Sub	Sleep/ Wake	Off Peak	Restart
SPRPDIR	C	Yes		Yes			
SRREMAL	C			Yes			
SRRENRH	C			Yes			
SRRENRL	C			Yes			
SRRINQR	C	Yes		Yes			Yes
SRRPREL	C	Yes		Yes			
SRRRICC	C	Yes		Yes			
SRRRPRO	C	Yes		Yes			
SRRRVAL	C	Yes		Yes			
SRRSRIN	C			Yes			
SRTLOAD	C	Yes	Yes	Yes			
SRTPURG	C	Yes		Yes			
SSPMFEE	C	Yes	Yes	Yes			
SSPRDEF	C	Yes	Yes	Yes			
SSPSCHD	C	Yes	Yes	Yes			
SSRASTZ	C	Yes	Yes	Yes			
SSRATSQ	C	Yes		Yes			
SSRROLL	C	Yes	Yes	Yes			
SSRSCMT	C	Yes	Yes	Yes			
SSRSCPR	C	Yes	Yes	Yes			
SSRSCRM	C			Yes			
SSRSCUP	C	Yes	Yes	Yes			
SSRSECT	C			Yes			
SSRTALY	C			Yes			

Report or Process	Language	Update/ Query	Audit	Job Sub	Sleep/ Wake	Off Peak	Restart
SURDELT	C	Yes	Yes	Yes			
SURLOAD	C	Yes		Yes			

Course Request and Scheduling

Report or Process	Language	Update/ Query	Audit	Job Sub	Sleep/ Wake	Off Peak	Restart
SCTC1000	COBOL			Yes			
SCTC1500	COBOL	Yes		Yes			
SCTC2000	COBOL			Yes			
CRQC3000	COBOL	Yes		Yes			
SCHC3000	COBOL	Yes		Yes			
SCTD0600	COBOL	Yes		Yes			
SCTH1000	COBOL			Yes			
SADASORT	COBOL			Yes			
SCTSORT	COBOL			No			
SFPCREQ	C	Yes		Yes			Yes
SFPFREQ	C	Yes		Yes			

Restart Procedures

Certain batch jobs perform database commits at intermediate steps during processing. If one of these processes terminates abnormally, you must restore the database to its original state prior to running the job again.

Course Request and Scheduling jobs are order dependent. To restart a process that has terminated abnormally, you must re-run the preceding processes.

Certain FTE jobs perform database commits at intermediate steps during processing. If one of these processes terminates abnormally, you must restore the database to its original state prior to running the job again.

SQL*Plus Scripts

The following describes the SQL*Plus scripts that are provided.

Please note that the user running the scripts below must have the necessary privileges to execute them. For more details on the following scripts, see SREADME.DOC.

Script	Description
SCOMMENT.SQL	Comment On Column statements for Banner Student tables.
SFRGRUP.SQL	Sample script of how to generate group codes for use with the time ticketing function of registration.
SFRTFAN.SQL	Produces a sample report for tuition and fee analysis view (SBVTFAN).
SLFEERL.SQL	Procedure to roll location management fee assessment rules from one term to another.
SPRIOR.SQL	Script used to set Student Priorities in Course Request and Schedule.
STBLCNT.SQL	Count of Student System tables by module.
STUALUG.SQL	Grants for Advancement tables.
STUFIMG.SQL	Grants for Finance tables.
STUGENG.SQL	Grants for General tables.
STUMEET.SQL	Adjusts number of meeting days in meeting time table.
STUPAYG.SQL	Grants for Human Resources tables.
STURESG.SQL	Grants for Financial Aid tables.
STUTRAG.SQL	Grants for Accounts Receivable tables.
XREFCOL2.SQL	This script will select STVSBGI records. These records will be inserted into SORXREF to establish the relationship between the STVSBGI value as an EDI value to the STVSBGI value as a Banner value.
XREFCOLL.SQL	This script will select STVSBGI records. These records will be inserted into SORXREF to establish the relationship between the STVSBGI value as an EDI value to the STVSBGI value as a Banner value.

Script	Description
XREFDRP.SQL	This script will drop the backup of the 2.0.10 version of the SORXREF table. This will be run after the 2.0.10 SORXREF data has been converted.
XREFETHN.SQL	This script will select STVSBGI records. These records will be inserted into SORXREF to establish the relationship between the STVSBGI value as an EDI value to the STVSBGI value as a Banner value.
XREFHSC2.SQL	This script will select STVSBGI records. These records will be inserted into SORXREF to establish the relationship between the STVSBGI value as an EDI value to the STVSBGI value as a Banner value.
XREFHSCH.SQL	This script will select STVSBGI records. These records will be inserted into SORXREF to establish the relationship between the STVSBGI value as an EDI value to the STVSBGI value as a Banner value.
XREFINS.SQL	This script reads records from the SORXREF_TEMP table. For each record not currently in SORXREF, based upon table name, EDI qualifier, and EDI code, a record will be inserted into SORXREF.
XREFMAJR.SQL	This script will select STVSBGI records. These records will be inserted into SORXREF to establish the relationship between the STVSBGI value as an EDI value to the STVSBGI value as a Banner value.
XREFNATN.SQL	This script will select STVSBGI records. These records will be inserted into SORXREF to establish the relationship between the STVSBGI value as an EDI value to the STVSBGI value as a Banner value.
XREFSTAT.SQL	This script will select STVSBGI records. These records will be inserted into SORXREF to establish the relationship between the STVSBGI value as an EDI value to the STVSBGI value as a Banner value.
XREFUPD.SQL	This script reads records from the SORXREF_TEMP table. For each record not currently in SORXREF, based upon table name, EDI qualifier, and EDI code, a record will be inserted into SORXREF.

Sleep/Wake-up

Banner provides two different mechanisms for running jobs in a cyclical or "sleep/wake-up" manner.

Setting Up Sleep/Wake Processes Using Method One

The first method uses OS command scripts and an SQL*Plus script to cause the job to run in a cyclical fashion. These jobs must be submitted from the operating system prompt and must be terminated manually. To compile programs to run in this fashion, you must define the `NO_SLEEP_SW` as a pre-compiler directive to exclude the code used by the second technique.

UNIX

The first command procedure, `sleepunx`, prompts for parameters needed by the second procedure and SQL*Plus scripts, `sleepunx.shl` and `sleepunx.sql` respectively. This procedure then starts (or submits) `sleepunx.shl`, which in turn starts `sleepunx.sql`. The SQL*Plus script `sleepunx.sql` will spool OS-specific commands to run the job into a file, provided there is actually work to do as determined by the parameters previously entered.

When the SQL*Plus script exits, `sleepunx.shl` executes the "spool" file. The parameters needed by the program are contained in a `XXXXXXX.dat` file which are read via input redirection when the job executes. The second command procedure `sleepunx.shl` then "sleeps" for the specified interval, awakes, and loops back to start the SQL*Plus script again.

VMS

This is essentially the same as for UNIX. The script names are `sleep.com`, `sleepdec.com`, and `sleepdec.sql`. Command input redirection is accomplished by defining `sys$input` as the `.dat` file. The “sleeping” is done with the “wait” command.

CMS

In this environment, there is only one command script, `sleepcms.exec`. Each job run in sleep/wake mode must have its own `.exec` script. The command script `sleepcms.exec` is provided as the model. You also need to set up separate CMS accounts for each job or you may combine the execs to run sequentially on a single account. The `.dat` file is read via input redirection. “Sleeping” is performed using the CP SLEEP command.

Setting Up Sleep/Wake Processes Using Method Two

Note

The following Banner systems and processes are valid for the Sleep/Wake processing described in this section:

Banner Student

Report/Process	Description
SFRSCHD	Student Schedules
SHRTRTC	Academic Transcript

Accounts Receivable Module

Report/Process	Description
TGRRCPT	Account Receipt
TSRCBIL	Student Billing Statement (Invoices)
TGRMISC	Miscellaneous Receipt

1. Define printer and print command on the Printer Validation Form (GTVPRNT). In the **(Printer) Code** field, enter a name to reference each specific printer that may be used for printing output from sleep/wake processing. In the **(Printer) Command** field, enter the correct operating system print command as it would normally be entered from the command line prompt, substituting an @ (at sign) as the place holder for the filename to be printed.

Operating System	Print Command
UNIX example:	`lp -d talaris1 @`
VMS example:	`print/queue=ln01 @`

2. On the appropriate System Distribution Initialization Information Form (SOADEST for Banner Student or TOADEST for Accounts Receivable), enter the printer code from GTVPRNT that should be identified with the collector table rows that will be inserted to the appropriate tables when online application forms create a request for output that can be generated by sleep/wake processing.

Note

The collector tables are as follows:

Process	Collector Table
SFRSCHD	SFRCBRQ
SHRTRTC	SHTTRAN

Process	Collector Table
TGRMISC	TBRCMIS
TGRRCPT	TBRCRCP
TSRCBIL	TBRCBRQ

3. On the Process Submission Control Form (GJAPCTL), for the valid sleep/wake jobs listed previously, enter the correct response for the parameter that specifies that the job should be processed for collector table entries. Refer to the documentation for each specific process to determine the appropriate response in each case (correct responses may be *COLLECTOR*, *Y*, *%*, etc.). In addition, each sleep/wake job has a printer code parameter. You must specify exactly the same code for this parameter answer that was entered on either SOADEST or TOADEST. A value of *Y* should be entered for the run in sleep/wake mode parameter, and a number of seconds should be specified for the sleep/wake interval (cycle) for each process.

Note

Do not enter the printer code in the top section of GJAPCTL; only enter it in the parameter section of the form. ■

4. The Sleep/Wake Maintenance Form (GJASWPT) should be used to stop the sleep/wake process or to change the sleep interval. A process name and printer code must be entered in the Key Information. A List of Values is available in each field to see the valid list of processes and printer codes that have ever been submitted for sleep/wake processing.

   To stop the process, enter *N* in the **Continue to Run f**ield and do a Save function. The job will not stop immediately, but rather will stop after the next time the process "wakes up" and finishes the next processing cycle. To change the sleep interval, enter the desired interval in the **Next Cycle Time (seconds)** field and perform a Save function.

   The GJASWPT form can also be used to view statistics regarding how many rows were processed for the most recent wakeup cycle and the total number of rows processed since the process was initiated. You can also determine if the processes terminated abnormally. by viewing the **Abnormal Termination** field. If there is a *Y* in **Abnormal Termination**, something caused the process to fail. You should review log files to determine the cause.

Operating Systems Without Sleep/Wake-up Commands

Operating systems which do not have sleep commands, or whose sleep commands may not be executed by user programs, must use the first method.

Collector File Entries

The following forms can be used to generate the collector file entries:

Form	Description/Generate	Report/ Process	Description
SFAREGS	Student Course Registration Form		
	Can generate:	SFRSCHD	Student Schedule
	Can generate:	TSRCBIL	Student Invoice
SHARQTC	Transcript Request Form		
	Can generate:	SHRTRTC	Immediate Academic Transcript
TGARCPT	Receipt Form		
	Can generate:	TGRRCPT	A/R Receipt
TFAAREV	Account Detail Query Form		
	Can generate:	TGRRCPT	A/R Receipt
TFADETF	Foreign Currency Detail Form		
	Can generate:	TGRRCPT	A/R Receipt
TFADETL	Account Detail Form		
	Can generate:	TGRRCPT	A/R Receipt
TFAMASS	Billing Mass Data Entry Form		
	Can generate:	TGRRCPT	A/R Receipt
TFAMDET	Billing Mass Entry Detail Form		
	Can generate:	TGRRCPT	A/R Receipt
TSAAREV	Currency Code Calculation Form		
	Can generate:	TGRRCPT	A/R Receipt
TSADETF	Student Foreign Currency Detail Form		
	Can generate:	TGRRCPT	A/R Receipt
TSADETL	Student Account Detail Form		
	Can generate:	TGRRCPT	A/R Receipt

Form	Description/Generate	Report/ Process	Description
TSAMASS	Billing Mass Data Entry Form Can generate:	TGRRCPT	A/R Receipt
TSASPAY	Student Payment Form Can generate:	TGRRCPT	A/R Receipt
TSAMISC	Miscellaneous Transaction Form Can generate:	TGRMISC	Miscellaneous Receipt
TFAMISC	Miscellaneous Transaction Form Can generate:	TGRMISC	Miscellaneous Receipt

2 Interfaces

Interfaces with External User Systems

Banner Data Load

Please refer to the *Banner® Student User Guide* for more information on interfaces with external user systems.

Chapter 9, “Recruiting” provides information on data load processing. The procedural documentation can be found in the “Procedures” section of the “Recruiting” chapter under these topics.

- “Data Load and Match Processing”
- “Search and Test Score Data Load”

Chapter 10, “Admissions”, provides information on AMCAS data load processing. The procedural documentation can be found in the “Procedures” section of the “Admissions” chapter under the “AMCAS (American Medical College Application Service) Load Procedures Using SRTLOAD” topic.

Process flows for data load can be found in Chapter 3, “Process Flows”, of the *Banner Student User Guide*.

Banner Recruiter Integration

Chapter 9, “Recruiting” provides information on the interface of Banner Student with Ellucian Recruiter. The procedural documentation can be found in the “Procedures” section of the “Recruiting” chapter under the “Banner Recruiter Integration” topic.

Interfaces within Banner

Please refer to the *Banner Student User Guide* and other Banner user guides for more information on interfaces within Banner.

Chapter 16, “Interfaces”, provides information on the interfaces between Banner Student and Banner Human Resources, as well as between Banner Student and Banner Advancement.

Note

The documentation for the Banner Accounts Receivable Interface to Banner Finance and the Banner Accounts Receivable Interface to Banner Financial Aid is contained in the *Banner Accounts Receivable User Guide*. Please refer to this guide for the interface information. ■

3 APIs

API Disclaimer

Warning

Please be advised that several APIs are currently intended to only support internal operations. To ensure data integrity, these APIs are not supported when called by external applications or interfaces to manipulate data. The recommendation for external applications is to use message level integration to integrate with these entities in Banner®. The following APIs come under this disclaimer:

- `gb_stvterm`
- `sb_course_registration`
- `sb_enrollment`
- `sb_term`
- `sp_grading`

Overview

Application Programming Interfaces (APIs) are used to facilitate the integration of Banner with other applications on campus and simplify code by encapsulating business logic in database packages. An API is a central program that creates, updates, and deletes data. APIs also execute and validate business rules before inserting or updating information.

Optimistic locking

The APIs used with faculty feedback processing use the Optimistic Locking strategy rather than the Pessimistic Locking strategy, which is used by the older Banner APIs. Optimistic Locking strategy is managed by the APIs using the Version column in each table. The Version column value is increased each time an API updates a record. On any delete or update, the version number of the record in the database must match the version number passed into the database. If it does not, an exception is raised.

APIs Used in Banner Student

The following tables and forms use APIs to process data in Banner Student. The form listed next to the table in this chart is the representative source used to build the API

validation and business rules. The APIs replace the corresponding code in the Banner forms.

Note

Most of the APIs support create, update, and delete signatures. Exceptions, such as queries, are noted under Task Performed.

Table	Object	API Object Name	API Entity Name	Task Performed
SARADAP	SAAADMS	sb_admissions_application	ADMISSIONSAPPLICATION	Used to maintain admissions applications
SARAPPD	SAAQUIK	sb_application_decision	APPLICATION_DECISION	Used in application decision process to consider separate decision codes on multiple applications for the same applicant within the same term
SCBCRSE	SCACRSE	sb_course	COURSE	Used to maintain basic course catalog information
SCBCRSE	SCRCATE	sp_catalog_export	COURSE_CATALOG	Used to extract course, level, schedule type, attribute, and description data from course catalog tables
SCRCLBD	SCACLBD	sb_course_labor	COURSE_LABOR	Used to query, update, create, or delete course labor distribution records (SCRCLBD table)
SCRINTG	SCADETL	sb_catlg_integration_partner	CATLG_INT_PARTNER	Used to maintain courses for integration partners process
SCRMEXC	SCAMEXC	dml_scrmexc	CRSE_MUTUAL_EXCLUSION	Used to support processing for records in the SCRMEXC table
SCRMEXC	SCAMEXC	sb_crse_mutual_exclusion	CRSE_MUTUAL_EXCLUSION	Used to process mutually exclusive courses for registration
SCRRATT	SCARRES	sb_crse_atts_restriction	CRSE_ATTS_RESTRICTION	Used to process student attribute restrictions defined at the Catalog level
SCRRCHR	SCARRES	sb_crse_chrt_restriction	CRSE_CHRT_RESTRICTION	Used to process cohort restrictions defined at the Catalog level

Table	Object	API Object Name	API Entity Name	Task Performed
SCRRDEP	SCARRES	sb_crse_dept_restriction	CRSE_DEPT_RESTRICTION	Used to process department restrictions defined at the Catalog level
SFBETRM	SFAREGS	sb_enrollment	ENROLLMENT	Used to maintain learner enrollment status
SFBFFSC	BWLKFDAD, BWLKFDBK	sb_feedback_session	FEEDBACK_SESSION	Used to process faculty feedback session information in Banner Faculty and Advisor Self-Service.
SFRCOLW	SFKWLAT, SFPWAIT	sb_wl_collector	WL_COLLECTOR	Used to maintain sections by term and CRN with new available seats that a waitlisted student can register for and be counted in enrollment
SFRENPO	SFAEPRT	sb_enr_ver_payment	ENR_VER_PAYMENT	Used to define special payment detail code options available for processing self-service enrollment verification requests
SFRENSP	SFAREGS	sb_studypath_enrollment	STUDYPATH_ENROLLMENT	Used to process the enrollment status for the study path
SFRENSS	SFAEPRT	sb_enr_ver_services	ENR_VER_SERVICES	Used to define special services available for processing self-service enrollment verification requests
SFRFFBK	BWLKFDAD, BWLKFDBK	sb_faculty_feedback	FACULTY_FEEDBACK	Used to process faculty feedback information in Banner Faculty and Advisor Self-Service.
SFRFFST	BWLKFDAD, BWLKFDBK	sb_estimated_grades	ESTIMATED_GRADES	Used to process estimated grades in Banner Faculty and Advisor Self-Service.
SFRRMSG	SFARMSG	sb_registration_msg	REGISTRATION_MSG	Used to return error messages for registration process by message code and sequence number

Table	Object	API Object Name	API Entity Name	Task Performed
SFRSTCR	SFAREGS	sb_course_ registration	REGISTRATION	Used to query on student course registration records
SFRSTCR	N/A	sp_grading	GRADING	Used to support existing midterm and final grade exchange between WebCT and Banner
SFRSTSH	SFASCPR	sb_scp_timestatus_ hist	SCP_TIMESTATUS_HIST	Used to process time status history records for student centric periods
SFRSTSL	SFASTSR	sb_scp_ruletime stat_lvl	SCP_RULETIMESTAT_LVL	Used to process time status rules by level for student centric periods
SFRSTST	SFASTSR	sb_scp_ruletime stat	SCP_RULETIMESTAT	Used to process time status rules for student centric periods
SFRWLNT	SFKWLAT	sb_wl_notifica tion	WL_NOTIFICATION	Used to maintain students notified of available seats in a section by term, CRN, and PIDM
SGBAPRG	SGAAPRG	sb_athletic_acad_ prog	ATHLETIC_ACAD_PROG	Used to process academic progress information for an athlete for athletic compliance
SGBDCPR	SGAAPRG	sb_athletic_ degree_comp	ATHLETIC_DEGC_COMP	Used to process degree completion information for an athlete for athletic compliance
SGBSTDN	SGASTDN	sb_learner	LEARNER	Used to maintain learner information
SGRATAD	SGASPRT	sb_athletic_ admissions	ATHLETIC_ADMISSIONS	Used to process admissions information for an athlete for athletic compliance
SGRATCT	SGASPRT	sb_athletic_ comments	ATHLETIC_COMMENTS	Used to process comment information for an athlete for athletic compliance
SGRATHA	SGASPRT	sb_athletic_ attributes	ATHLETIC_ATTRIBUTES	Used to process attribute information for an athlete for athletic compliance

Table	Object	API Object Name	API Entity Name	Task Performed
SGRATHC	SGASPRT	sb_athletic_competition	ATHLETIC_COMPETITION	Used to process competition information for an athlete for athletic compliance
SGRATHE	SGASPRT	sb_athletic_eligibility	ATHLETIC_ELIGIBILITY	Used to process eligibility information for an athlete for athletic compliance
SGRATHT	SGASPRT	sb_athletic_transfer	ATHLETIC_TRANSFER	Used to process transfer information for an athlete for athletic compliance
SGRSPRT	SGASPRT	sb_athletic_compliance	ATHLETIC_COMPLIANCE	Used to process sport information for an athlete for athletic compliance
SGRSTSP	SOQOLIB, SGASTDN, SFAREGS	sb_studypath	STUDYPATH	Used to process study paths in general student and registration
SHBGADR	SHAGADR	sb_gradapp_displayrule	SB_GRADAPP_DISPLAY RULE	Used to track graduation application display rules
SHBGADS	SHAGADS	sb_gradapp_selrule	SB_GRADAPP_SELRULE	Used to track graduation application display rule selection
SHBGAPP	SHAGAPP	sb_gradapp	SB_GRADAPP	Used to track graduation applications
SHBGELR	SHAGELR	sb_gradapp_eligible	SB_GRADAPP_ELIGIBLE	Used to track graduation application eligibility rules selection
SHBHEAD	SHRPESI	sb_edi_header	EDI_HEADER	Used to access imported XML transcript records for the document master header record
SHBTATC	SHATATC, SHATATR, SHRTCIM	sb_transfer_crse	TRANSFER_CRSE	Used to process transfer courses
SHRASES	SHRPESI	sb_edi_acad_sess	EDI_ACAD_SESSION	Used to load imported XML transcript academic session data into SHRASES

Table	Object	API Object Name	API Entity Name	Task Performed
SHRAUDE	SHAEDIS, SHRPESI	sb_edi_acrec_ude	EDI_ACREC_UDE	Used to access imported XML transcript records for academic record UDE data
SHRCRSR	SHRPESI	sb_edi_course	EDI_COURSE	Used to access imported XML transcript records for course UDE data
SHRDGMR	SHADEGR	sb_learneroutcome	LEARNEROUTCOME	Used to maintain program outcome, degree, and award information
SHREDIS	SHRPESI	sb_edi_status	EDI_STATUS	Used to access imported XML transcript records for EDI status data
SHREPTD	SHRPESI	sb_pesc_status_export	PESC_STATUS_EXPORT	Used to access imported XML transcript records for transcript export status
SHRGADA	SHAGADR	sb_gradapp_graddate	SB_GRADAPP_GRADDATE	Used to track graduation application graduation data availability
SHRGAPO	SHAGADR	sb_gradapp_paymentopts	SB_GRADAPP_PAYMENTOPTS	Used to track graduation application payment options
SHRHDR4	SHRPESI	sb_edi_doc_header	EDI_DOC_HEADER	Used to access imported XML transcript records for the document identification header
SHRICMT	SHATATR	sb_transfer_equiv_cmt	TRANSFER_EQUIV_CMT	Used to process equivalent course comments
SHRIDEN	SHRPESI	sb_edi_identification	EDI_IDENTIFICATION	Used to access imported XML transcript records for person identification data
SHRIMMU	SHRPESI	sb_edi_immunization	EDI_IMMUMIZATION	Used to load immunization data from imported XML transcript
SHRINCG	SHAINCG	sb_incmp_grading	INCMP_GRADING	Used to define incomplete grade rules

Table	Object	API Object Name	API Entity Name	Task Performed
SHRIPTD	SHRPESI	sb_pesc_status_imp	PESC_STATUS_IMP	Used to access imported XML transcript records for transcript import status
SHRMEDI	SHREDIS, SHRPESI	sb_edi_medical	EDI_MEDICAL	Used to access imported XML transcript records for medical data
SHRSTSC	SHAEDIS, SHRPESI	sb_edi_subtest_score	EDI_SUBTEST_SCORE	Used to access imported XML transcript records for subtest score data
SHRSTST	SHAEDIS, SHRPESI	sb_edi_subtest	EDI_SUBTEST	Used to access imported XML transcript records for subtest data
SHRSUDE	SHAEDIS, SHRPESI	sb_edi_stud_ude	EDI_STUD_UDE	Used to access imported XML transcript records for student UDE data
SHRSUMA	SHRPESI	sb_edi_acad_summ	EDI_ACAD_SUMM	Used to load imported XML transcript academic summary data into SHRSUMA
SHRSUMS	SHRPESI	sb_edi_acad_sess_sum	EDI_ACAD_SESS_SUM	Used to load imported XML transcript academic session summary data into SHRASES
SHRTATC	SHATATR	sb_transfer_equiv	TRANSFER_EQUIV	Used to process equivalent courses
SHRTCAT	SHATATC, SHATATR, SHRTCIM	sb_transfer_crse_attr	TRANSFER_CRSE_ATTR	Used to process transfer course attributes
SHRTCMT	SHATATC, SHATATR	sb_transfer_crse_cmt	TRANSFER_CRSE_CMT	Used to process transfer course comments
SHRTEST	SHAEDIS, SHRPESI	sb_edi_test	EDI_TEST	Used to access imported XML transcript records for test data
SHRTRAT	SHATATR	sb_transfer_equiv_attr	TRANSFER_EQUIV_ATTR	Used to process equivalent course attributes

Table	Object	API Object Name	API Entity Name	Task Performed
SHRTRNM	SHATPRT	sb_transcript_name	TRANSCRIPT_NAME	Used to maintain define name to be printed on transcript for name hierarchy and transcript type
SHRTRNM	SHARQTC	sb_transcript_req_name	TRANSCRIPT_REQ_NAME	Used to maintain identify name hierarchy for transcript request
SHRTRNS	SHRTRNS	sb_transcript_name_source	TRANSCRIPT_NAME_SOURCE	Used to maintain restrictions for name sources available for selecting names to be printed on transcript
SHRTUDE	SHAEDIS, SHRPESI	sb_edi_trans_ude	EDI_TRANS_UDE	Used to access imported XML transcript records for transcript UDE data
SIBINST	SIAINST	sb_faculty	FACULTY	Used to maintain faculty member and advisor information
SIRASGN	SIAASGN	sb_facassignment	FACULTYASSIGNMENT	Used to maintain course assignments for a faculty member or advisor
SLRBMAP	SLARMAP	sb_roompreference	ROOMPREFERENCE	Used to maintain dorm room and meal plan applications
SLRMASG	SLAMASG	sb_mealassignment	MEALASSIGNMENT	Used to maintain meal assignments
SLRPASG	SLAPASG	sb_phoneassignment	PHONEASSIGNMENT	Used to maintain telephone assignments
SLRRMAT	SLARMAT	sb_roommate	ROOMMATE	Used to maintain roommate applications
SMRCMXL	SMACACT, SMRCPRG	sb_compliance	COMPLIANCE_XML	Used to support XML compliance processing
SMRCXML	SMACACT, SMRCPRG	dml_smrcxml	COMPLIANCE_XML	Used to support processing for records in the SMRCXML table

Table	Object	API Object Name	API Entity Name	Task Performed
SOBCACT	SORCACT	sb_learnercurric status	LEARNERCURRICSTATUS	Used to maintain learner curriculum activity and status rules
SOBCSPR	SOAWLTC	sb_crse_select_pri	CRSE_SELECT_PRI	Used to maintain priority rules to be applied to waitlist course selection criteria by sequence number
SOBEXCR	SOAWLTC	sb_wl_exclusion	WL_EXCLUSION	Used to maintain waitlist exclusion rules by sequence number that prevent students from being added to a course waitlist if they fall into any of the specified exclusions
SOBLMOD	SOACTRL	sb_learnermodule	LEARNERMODULE	Used to maintain learner module rules
SOBMAUD	SOAMAUD	sb_mass_entry_form	MASS_ENTRY_FORM	Used to store the mass entry session header information
SOBROPR	SOAWLTC	sb_wl_reordering	WL_REORDERING	Used to maintain waitlist reordering rules by sequence number to be applied to students registered in the waitlist queue
SOBSCPS	SOASCPT	sb_centricperiod_ term	CENTRICPERIOD_TERM	Used to process student centric periods associated with terms
SOBTERM	SOATERM	sb_term	TERM	Used to maintain controls for a specific term - no create allowed, only update and delete
SOBWLTC	SOAWLTC	sb_wl_term_control	WL_TERM_CONTROL	Used to configure waitlist by term
SORCSCP	SOACSCP	sb_scp_continuant	SCP_CONTINUANT	Used to process continuant student centric periods
SORCSTS	SAAEAPS	sb_curriculum_ status_event	CURRICULUM_STATUS_ EVENT	Used to house values for curriculum events where field of study is copied, and new record has new curriculum status (STVCSTS) value

Table	Object	API Object Name	API Entity Name	Task Performed
SORDLIM	SORDLIM	sb_datafile_delimiter	FILEDELIMETER	Used to maintain datafile file delimeters
SORFACS	SOAFACS	sb_facsec_access	FACSEC_ACCESS	Used to control faculty member and advisor access to self-service processes and student information
SORFTRM	SOATERM	sb_fac_sec_dates	FAC_SEC_DATES	Used to process availability of main menu and term selection in Self-Service by date for faculty members
SORHCUR	SOIHCUR	sb_curriculum_history	CURRICULUM_ARCHIVE	Used to write copy of purged curriculum data to SORHCUR for stored archive
SORHFOS	SOIHCUR	sb_fieldofstudy_archive	FIELDOFSTUY_ARCHIVE	Used to write copy of purged field of study data to SORHFOS for stored archive
SORHSCH	SOAHSCH	sb_highschool	HIGHSCHOOL	Used to create high school records
SORLCDF	SOQOLIB	sb_curriculum_default	USER_CURRICULUM_DEFAULT	Used to house default values defined by leaner module for curriculum/field of study records for improved data entry
SORLCUR	SOQOLIB	sb_curriculum	CURRICULUM	Used to maintain learner curriculum information
SORLFOS	SOQOLIB	sb_fieldofstudy	FIELDOFSTUDY	Used to maintain learner curriculum field of study information
SORLMFS	SOACTRL	sb_fieldofstudy_allowed	FIELDOFSTUDY_ALLOWED	Used to define maximum current/active field of study allowed for a curriculum and learner module, as well as enrollment verification/ transcript self-service delivery options
SORMAUD	SOPMAUD	sb_mass_entry	MASS_ENTRY	Used to store the mass entry selection information

Table	Object	API Object Name	API Entity Name	Task Performed
SORMECL	SOAMAUD	sb_mass_entry_column	MASS_ENTRY_COLUMN	Used to store the mass entry columns and values for search and update processing
SORSCPT	SOASCPT	sb_centricperiod	CENTRICPERIOD	Used to retrieve effective student centric period, term, and student type for given student for specific term, as well as retrieve previous and latest time status codes for student for specific student centric period
SORSPNM	SOACTRL	sb_studypath_name	STUDYPATH_NAME	Used to parse study path description that is viewable by students and faculty
SORTEST	SOATEST	sb_testscore	TESTSCORE	Used to create test score records
SORTSPC	SOATEST	sb_test_percentile	TEST_PERCENTILE	Used to maintain student test score percentile details for master test records in SORTEST
SORWSCR	SOAWSCR	sb_pagefieldconfig	PAGEFIELDCONFIG	Used to maintain Banner 9.0 Self-Service registration page and field display configuration definitions
SRBRECR	SRARECR	sb_recruit	RECRUIT	Used to maintain prospective applicant information
SRRTPTS	SRATPTS	sb_datafile_test_score	FILETESTSCORE	Used to maintain date origins for STVTESC codes
SSBSECT	SSASECT	sb_section	SECTION	Used to maintain the schedule of classes as defined in the course catalog
SSBWLSC	SSAWLSC	sb_wl_section_ctrl	WL_SECTION_CTRL	Used to maintain the waitlist configuration for each section by term and CRN

Table	Object	API Object Name	API Entity Name	Task Performed
SSRCLBD	SSACLBD	sb_section_labor	SECTION_LABOR	Used to query, update, create, or delete course labor distribution records (SSRCLBD table)
SSRRATT	SSARRES	sb_sect_atts_ restriction	SECT_ATTS_RESTRICTION	Used to process student attribute restrictions defined at the Schedule level
SSRRCHR	SSARRES	sb_sect_chrt_ restriction	SECT_CHRT_RESTRICTION	Used to process cohort restrictions defined at the Schedule level
SSRRDEP	SSARRES	sb_sect_dept_ restriction	SECT_DEPT_RESTRICTION	Used to process department restrictions defined at the Schedule level
STVATTS	STVATTS	sb_stvatts	SB_STVATTS	Used to process student attribute codes
STVCHRT	STVCHRT	sb_stvchrt	SB_STVCHRT	Used to process cohort codes
STVCKSR	SAAADMS, SAAACKL	sb_stvcksr	CHECKLIST_SOURCE	Used to track source of checklist items
STVCKST	SAAADMS, SAAACKL	sb_stvckst	CHECKLIST_STATUS	Used to track status of checklist items
STVCMTT	STVCMTT	sb_stvcmtt	SB_STVCMTT	Used to validate STVCMTT for athletic compliance processing
STVDCPR	STVDCPR	sb_stvdcpr	SB_STVDCPR	Used to validate STVDCPR for athletic degree completion processing
STVFFVA	BWLKFDAD, BWLKFDBK	sb_feedback_codes	FEEDBACK_CODES	Used to process feedback codes in Banner Faculty and Advisor Self-Service.
STVGADR	STVGADR	sb_stvgadr	SB_STVGADR	Used to track graduation application display rule codes
STVGAST	STVGAST	sb_stvgast	SB_STVGAST	Used to track graduation application status codes
STVMECL	STVMECL	sb_stvmecl	STVMECL	Used to validate the mass entry search and update columns

Table	Object	API Object Name	API Entity Name	Task Performed
STVNIST	STVNIST	sb_stvnist	STVNIST	Used to validate STVNIST_CODE field and return related description
STVSAAT	STVSAAT	sb_stvsaat	SB_STVSAAT	Used to validate STVSAAT for athletic compliance processing
STVSAEL	STVSAEL	sb_stvsael	SB_STVSAEL	Used to validate STVSAEL for athletic compliance processing
STVSAQS	STVSAQS	sb_stvsaqs	SB_STVSAQS	Used to validate STVSAQS for athletic compliance processing
STVSARE	STVSARE	sb_stvsare	SB_STVSARE	Used to validate STVSARE for athletic compliance processing
STVSARX	STVSARX	sb_stvsarx	SB_STVSARX	Used to validate STVSARX for athletic compliance processing
STVSATR	STVSATR	sb_stvsatr	SB_STVSATR	Used to validate STVSATR for athletic compliance processing
STVSCPC	STVSCPC	sb_stvscpc	SB_STVSCPC	Used to validate student centric period cycle codes
STVSTSP	STVSTSP	sb_stvstsp	SB_STVSTSP	Used to validate the study path status for the student
STVTRNS	STVTRNS	sb_stvtrns	SB_STVTRNS	Used to house cursors/ functions and retrieve validation table data from transcript name source

APIs Used in Banner General with Banner Student Forms and Tables

The following Student tables and forms use APIs to process data in Banner General and Banner Student.

Table	Object	API Object Name	API Entity Name	Task Performed
SLBBLDG	SLABLDG	gb_bldgdefinition	BLDGDEFINITION	Used to maintain building information
SLBRDEF	SLARDEF	gb_roomdefinition	ROOMDEFINITION	Used to maintain room information by building
SLRRASG	SLARASG	gb_roomassignment	ROOMASSIGNMENT	Used to maintain dorm room assignments
SORCONC	SIAFDEG, SOAPCOL	gb_pcol_conc	PCOL_CONCENTRATION	Used to maintain a person's educational background, including institutions attended, degrees received at each institution, and majors, minors, and areas of *concentration* at each institution
SORDEGR	SAADCRV, SIAFDEG, SOAPCOL, SRAQUIK	gb_pcol_degree	PCOL_DEGREE	Used to maintain a person's educational background, including institutions attended, *degrees* received at each institution (majors, minors, and areas of concentration)
SORMAJR	SIAFDEG, SOAPCOL	gb_pcol_major	PCOL_MAJOR	Used to maintain a person's educational background, including institutions attended, degrees received at each institution (*majors*, minors, and areas of concentration)
SORMINR	SIAFDEG, SOAPCOL	gb_pcol_minor	PCOL_MINOR	Used to maintain a person's educational background, including institutions attended, degrees received at each institution (majors, *minors*, and areas of concentration)

Table	Object	API Object Name	API Entity Name	Task Performed
SORPCOL	SAADCRV, SHAEDIS, SIAFDEG, SOAPCOL, SRAQUIK	gb_pcol	PRIOR_COLLEGE	Used to maintain a person's educational background, including institutions attended, degrees received at each institution (majors, minors, and areas of concentration)
SPBPERS	SPAPERS	gb_bio	BIO	Used to maintain biographic/ demographic information for an individual
SPRADDR	SPAIDEN	gb_address	ADDRESS	Used to maintain address information
SPREMRG	SPAEMRG	gb_emergency_ contact	EMERGENCY_CONTACT	Used to maintain emergency contact information for an individual
SPRHOLD	SOAHOLD	gb_hold	HOLD	Used to place or remove holds on an account
SPRIDEN	SPAIDEN	gb_identification	IDENTIFICATION	Used to maintain person and non-person biographic/ demographic information
SPRMEDI	GOAMEDI	gb_medical	MEDICAL	Used to maintain information about medical conditions of people at your institution, including students, faculty, and staff
SPRTELE	SPATELE	gb_telephone	TELEPHONE	Used to maintain telephone information
SSRMEET	SSASECT	gb_classtimes	CLASSTIMES	Used to maintain section and event meeting times
STVADMR	STVADMR	gb_stvadmr	PRIOR_COLLEGE	Used to check for existence of admission request code information for a prior college
STVHONR	STVHONR	gb_stvhonr	PCOL_DEGREE	Used to check for existence of institutional honors for a prior degree
STVMEDI	STVMEDI	gb_stvmedi	MEDICAL	Used to check for existence of medical information

Table	Object	API Object Name	API Entity Name	Task Performed
STVPENT	STVPENT	gb_stvpent	VISA	Used to check for existence of port of entry information
STVSBGI	STVSBGI	gb_stvsbgi	PRIOR_COLLEGE	Used to maintain general information, such as address, comments, and contacts, about a source or background institution for a prior college
STVTERM	STVTERM	gb_stvterm	STVTERM	Used to query on term validation information
STVVTYP	STVVTYP	gb_stvvtyp	VISA	Used to check for existence of visa type record

Curriculum Conversion Using Functions and APIs

Four of the functions in package SOKLCUR (f_convert_recruit, f_convert_applicant, f_convert_learner, and f_convert_outcome) are used to call APIs to perform the insert activity. The functions can be launched from a batch process or when the a form with curriculum data is initially opened at your institution. This includes: SRARECR, SRAQUIK, SAAADMS, SAAQUIK, SGASTDN, SFAREGS, and SHADEGR. These forms will check to see if the curriculum data has been converted, and if it has not, the function will be executed for the PIDM.

The table below explains how the functions convert the curriculum data using APIs.

Function Name	Table Read	Conversion Step	API Task	Data Converted into Table
f_convert_recruit	SRBRECR	Convert primary curriculum	Insert recruiting curriculum base	SORLCUR
			Insert recruiting curriculum study entry	SORLFOS
f_convert_applicant	SARADAP	Convert primary curriculum	Insert admissions curriculum base	SORLCUR
			Insert admissions curriculum study entry	SORLFOS

Function Name	Table Read	Conversion Step	API Task	Data Converted into Table
		Convert secondary curriculum	Insert admissions curriculum base	SORLCUR
			Insert admissions curriculum study entry	SORLFOS
f_convert_learner	SGBSTDN	Convert primary curriculum	Insert learner curriculum base	SORLCUR
			Insert learner curriculum study entry	SORLFOS
		Convert secondary curriculum	Insert learner curriculum base	SORLCUR
			Insert learner curriculum study entry	SORLFOS
f_convert_outcome	SHRDGMR	Convert primary curriculum	Insert history curriculum base	SORLCUR
			Insert history curriculum study entry	SORLFOS
		Convert secondary curriculum	Insert history curriculum base	SORLCUR
			Insert history curriculum study entry	SORLFOS

4 Selecting and Maintaining Concurrent Curricula Data

Curriculum Processing Overview

There are two methods used for selecting and maintaining concurrent curricula data.

- The pre-Banner® 7.0 method used to select the current curriculum data selected the base record (SRBRECR, SARADAP, SGBSTDN, SHRDGMR) and pulled curriculum data from that record. The SRBRECR, SARADAP, SGBSTDN, and SHRDGMR tables also contained the curriculum priority as part of the column definitions.
- In Banner 7.X and forward, the curriculum data is maintained in the SORLCUR and SORLFOS concurrent curriculum tables. Multiple curriculums are allowed, and the priority is a value in a column. The process used to select the primary curriculum involves selecting the first curriculum row when reading the curriculum rows in priority order.

The following table cross-references the old primary curriculum columns to the new columns.

Pre-Banner 7.0 Columns	Banner 7.X Columns
SARADAP_TERM_CODE_ENTRY	SORLCUR_TERM_CODE
SRBRECR_TERM_CODE	
SGBSTDN_TERM_CODE_EFF	
SHRDGMR_TERM_CODE_STUREC	
SARADAP_APPL_NO	SORLCUR_KEY_SEQNO
SRBRECR_ADMIN_SEQNO	
SHRDGMR_SEQ_NO	

Pre-Banner 7.0 Columns	Banner 7.X Columns
SARADAP_LEVL_CODE SARADAP_LEVL_CODE_2 SGBSTDN_LEVL_CODE SGBSTDN_LEVL_CODE_2 SHRDGMR_LEVL_CODE SRBRECR_LEVL_CODE	SORLCUR_LEVL_CODE
SARADAP_COLL_CODE_1 SARADAP_COLL_CODE_2 SGBSTDN_COLL_CODE_1 SGBSTDN_COLL_CODE_2 SHRDGMR_COLL_CODE_1 SHRDGMR_COLL_CODE_2 SRBRECR_COLL_CODE	SORLCUR_COLL_CODE
SARADAP_CAMP_CODE SARADAP_CAMP_CODE_2 SGBSTDN_CAMP_CODE SGBSTDN_CAMP_CODE_2 SHRDGMR_CAMP_CODE SHRDGMR_CAMP_CODE_2 SRBRECR_CAMP_CODE	SORLCUR_CAMP_CODE
SARADAP_DEGC_CODE_1 SARADAP_DEGC_CODE_2 SGBSTDN_DEGC_CODE_1 SGBSTDN_DEGC_CODE_2 SHRDGMR_DEGC_CODE SRBRECR_DEGC_CODE	SORLCUR_DEGC_CODE
SARADAP_PROGRAM_1 SARADAP_PROGRAM_2 SGBSTDN_PROGRAM_1 SGBSTDN_PROGRAM_2 SHRDGMR_PROGRAM SRBRECR_PROGRAM_1	SORLCUR_PROGRAM

Pre-Banner 7.0 Columns	Banner 7.X Columns
SARADAP_TERM_CODE_CTLG_1 SARADAP_TERM_CODE_CTLG_2 SGBSTDN_TERM_CODE_CTLG_1 SGBSTDN_TERM_CODE_CTLG_2 SHRDGMR_TERM_CODE_CTLG_1 SHRDGMR_TERM_CODE_CTLG_2 SRBRECR_TERM_CODE_CTLG_1	SORLCUR_TERM_CODE_CTLG
SGBSTDN_TERM_CODE_ADMIT SGBSTDN_TERM_CODE_ADMIT_2	SORLCUR_TERM_CODE_ADMIT
SGBSTDN_ADMT_CODE SGBSTDN_ADMT_CODE_2	SORLCUR_ADMT_CODE
SGBSTDN_TERM_CODE_MATRIC	SORLCUR_TERM_CODE_MATRIC
SGBSTDN_PRIM_ROLL_IND SGBSTDN_SECD_ROLL_IND	SORLCUR_ROLL_IND

Pre-Banner 7.0 Columns	Banner 7.X Columns
SARADAP_DEPT_CODE SARADAP_MAJR_CODE_1	SORLFOS_MAJR_CODE SORLFOS_DEPT_CODE WHERE SORLFOS_LFST_CODE = SB_FIELDOFSTUDY_STR.F_MAJOR
SGBSTDN_DEPT_CODE SGBSTDN_MAJR_CODE_1	
SHRDGMR_DEPT_CODE SHRDGMR_MAJR_CODE_1	
SRBRECR_DEPT_CODE SRBRECR_MAJR_CODE	
SARADAP_DEPT_CODE_1_2 SARADAP_MAJR_CODE_1_2	
SGBSTDN_DEPT_CODE_1_2 SGBSTDN_MAJR_CODE_1_2	
SHRDGMR_DEPT_CODE_1_2 SHRDGMR_MAJR_CODE_1_2	
SRBRECR_DEPT_CODE_2 SRBRECR_MAJR_CODE_2	
SARADAP_DEPT_CODE_2 SARADAP_MAJR_CODE_2	
SGBSTDN_DEPT_CODE_2 SGBSTDN_MAJR_CODE_2	
SHRDGMR_DEPT_CODE_2 SHRDGMR_MAJR_CODE_2	
SARADAP_DEPT_CODE_2_2 SARADAP_MAJR_CODE_2_2	
SGBSTDN_DEPT_CODE_2_2 SGBSTDN_MAJR_CODE_2_2	
SHRDGMR_DEPT_CODE_2_2 SHRDGMR_MAJR_CODE_2_2	

Pre-Banner 7.0 Columns	Banner 7.X Columns
SARADAP_MAJR_CODE_MINR_1_1 SARADAP_MAJR_CODE_MINR_1_2 SARADAP_MAJR_CODE_MINR_2_1 SARADAP_MAJR_CODE_MINR_2_2 SGBSTDN_MAJR_CODE_MINR_1 SGBSTDN_MAJR_CODE_MINR_1_2 SGBSTDN_MAJR_CODE_MINR_2 SGBSTDN_MAJR_CODE_MINR_2_2 SHRDGMR_MAJR_CODE_MINR_1 SHRDGMR_MAJR_CODE_MINR_1_2 SHRDGMR_MAJR_CODE_MINR_2 SHRDGMR_MAJR_CODE_MINR_2_2 SRBRECR_MAJR_CODE_MINR_1 SRBRECR_MAJR_CODE_MINR_1_2	SORLFOS_MAJR_CODE WHERE SORLFOS_LFST_CODE = SB_FIELDOFSTUDY_STR.F_MINOR

Pre-Banner 7.0 Columns	Banner 7.X Columns
SARADAP_MAJR_CODE_CONC_1 SARADAP_MAJR_CODE_CONC_1_2 SARADAP_MAJR_CODE_CONC_1_3 SARADAP_MAJR_CODE_CONC_121 SARADAP_MAJR_CODE_CONC_122 SARADAP_MAJR_CODE_CONC_123 SARADAP_MAJR_CODE_CONC_2 SARADAP_MAJR_CODE_CONC_211 SARADAP_MAJR_CODE_CONC_212 SARADAP_MAJR_CODE_CONC_213 SARADAP_MAJR_CODE_CONC_221 SARADAP_MAJR_CODE_CONC_222 SARADAP_MAJR_CODE_CONC_223 SGBSTDN_MAJR_CODE_CONC_1 SGBSTDN_MAJR_CODE_CONC_1_2 SGBSTDN_MAJR_CODE_CONC_1_3 SGBSTDN_MAJR_CODE_CONC_121 SGBSTDN_MAJR_CODE_CONC_122 SGBSTDN_MAJR_CODE_CONC_123 SGBSTDN_MAJR_CODE_CONC_2 SGBSTDN_MAJR_CODE_CONC_2_2 SGBSTDN_MAJR_CODE_CONC_2_3 SGBSTDN_MAJR_CODE_CONC_221 SGBSTDN_MAJR_CODE_CONC_222 SGBSTDN_MAJR_CODE_CONC_223 SHRDGMR_MAJR_CODE_CONC_1 SHRDGMR_MAJR_CODE_CONC_1_2 SHRDGMR_MAJR_CODE_CONC_1_3 SHRDGMR_MAJR_CODE_CONC_121 SHRDGMR_MAJR_CODE_CONC_122 SHRDGMR_MAJR_CODE_CONC_123 SHRDGMR_MAJR_CODE_CONC_2 SHRDGMR_MAJR_CODE_CONC_2_2 SHRDGMR_MAJR_CODE_CONC_2_3 SHRDGMR_MAJR_CODE_CONC_221 SHRDGMR_MAJR_CODE_CONC_222 SHRDGMR_MAJR_CODE_CONC_223 SRBRECR_MAJR_CODE_CONC_111 SRBRECR_MAJR_CODE_CONC_112 SRBRECR_MAJR_CODE_CONC_113 SRBRECR_MAJR_CODE_CONC_121 SRBRECR_MAJR_CODE_CONC_122 SRBRECR_MAJR_CODE_CONC_123	SORLFOS_MAJR_CODE WHERE SORLFOS_LFST_CODE = SB_FIELDOFSTUDY_STR.F_CONCENTRATION

The Banner concurrent curriculum tables provide the additional ability to perform non-destructive updates. This adds some complexity to the selection of curriculum data. Data is not overwritten. Instead, changes are inserted into a new record, and the older record becomes non-current. That is how non-destructive updates are performed, so a history of curriculum changes can be maintained.

Two fields in the SOVLCUR and SOVLFOS views (`CURRENT_IND` and `ACTIVE_IND`) are used to keep track of the most recent version of the curriculum and field of study data. These columns indicate which curriculum rows are current and active.

- The `CURRENT_IND` field is a calculated element with a value of *Y* or *N*.
- The `ACTIVE_IND` field value is derived from the curriculum status rules defined on SORCACT and is based on the `SORLCUR_CACT_CODE` and `SORLFOS_CACT_CODE` values.

The `SORLCUR_CURRENT_CDE` and `SORLFOS_CURRENT_CDE` columns are set to *Y* or *Null* and reflect the current status. These settings are determined during the backfill process when curricula and fields of study are committed to the database. The `CURRENT_IND` reflects the value of the `CURRENT_CDE` if the switch has been made to using the `CURRENT_CDE` value. Please see the "How to Use the Current Indicator" section which follows for more information on the `CURRENT_CDE`.

The ORDER column in the SOVCFOS, SOVCCUR, SOVACUR, SAVCCUR, SAVACUR, SRVCCUR, SRVACUR, SGVCCUR, SGVACUR, SHVCCUR, and SHVACUR views represents the ordinal value of the priority. If the curriculum or field of study is not current or not active, the order is set to the value of *0* (zero). The primary curriculum and field of study has an order value of *1*.

Additional columns on SORLCUR are optional and apply only to tracking additional information for the learner curriculum. These columns are not automatically populated with data.

- `SORLCUR_SITE_CODE`
- `SORLCUR_RATE_CODE`
- `SORLCUR_STYP_CODE`
- `SORLCUR_LEAV_CODE`
- `SORLCUR_LEAV_FROM_DATE`
- `SORLCUR_LEAV_TO_DATE`
- `SORLCUR_GRAD_DATE`
- `SORLCUR_TERM_CODE_GRAD`
- `SORLCUR_ACYR_CODE`

Two additional columns appear only on the learner curriculum, and then only if the curriculum is created as a result of an admissions decision.

- SORLCUR_APPL_SEQNO
- SORLCUR_APPL_KEY_SEQNO

The SORLCUR_APPL_SEQNO column is the sequence number on the application curriculum, and the SORLCUR_APPL_KEY_SEQNO column is the application number from SARADAP. These columns will be null on all other records, including duplications of the originally accepted learner curriculum.

Another column, SORLCUR_GAPP_SEQNO, is the sequence number on the graduation application for the curriculum. This column is on the learner curriculum and the outcome curriculum.

Auditing the update of the rolled sequence number and the graduation application sequence is recorded in the following columns:

- SORLCUR_USER_ID_UPDATE
- SORLCUR_ACTIVITY_DATE_UPDATE
- SORLFOS_USER_ID_UPDATE
- SORLFOS_ACTIVITY_DATE_UPDATE

The originally inserted activity date and user ID will remain in the original columns for the user ID and activity date.

How to Use the Current Indicator

The **Current (Indicator)** checkbox is displayed in the Curriculum Windows. Only curriculum records with the indicator checked (set to *Y*) are displayed in the Curricula Summary blocks. The Current Code column on the SORLCUR and SORLFOS tables (SORLCUR_CURRENT_CDE and SORLFOS_CURRENT_CDE) indicates the current curriculum or field of study record for the module. Valid values for the column are *Y* for current and *Null* for not current.

The Current Code optionally replaces the calculated Current Indicator, found in the SOVLCUR and SOVLFOS views. Scripts are delivered to populate the Current Code column with the value from the Current Indicator column. Once the Current Code has been populated on all old curriculum records, the SOVLCUR and SOVLFOS views will use the Current Code. The other views used with curriculum processing also use the Current Code. If you do not run the scripts, you can continue to use the Current Indicator.

The following indicators on SOBCTRL are used to track the migration process:

- SOBCTRL_CURRENT_RECR_CDE
- SOBCTRL_CURRENT_ADM_CDE
- SOBCTRL_CURRENT_LEARNER_CDE
- SOBCTRL_CURRENT_OUTCOME_CDE

The end term information for learner curriculum records is displayed in the Curriculum Windows and the Curricula Summary blocks. This information is used to determine if the learner curriculum is current, based on the effective term and end term of the SGBSTDN record.

Current Indicator Conversion

Conversion scripts are used to migrate the Current Indicator to the Current Code. The backfill process also performs updates.

Conversion Scripts

Conversion scripts are delivered to migrate the setting of the **Current (Indicator)** to the Current Code. These scripts are executed from SQL*Plus. They are not run as part of the upgrade process. The scripts are:

- susoplccd_recr.sql (Recruiting)

  This script updates the SORLCUR_CURRENT_CDE and SORLFOS_CURRENT_CDE columns for recruiting records.

  Banner views that query the Current Indicator will continue to execute the API functions that calculate the Current Indicator until this script has run to a successful completion. The SOBCTRL_CURRENT_RECR_CDE column is updated to *Y* to track the execution of this script.

  Once this script has been run, the Curriculum API procedure will return the SORLCUR_CURRENT_CDE, and the Field of Study API function will return the SORLFOS_CURRENT_CDE.

- susoplccd_adm.sql (Admissions)

  This script updates the SORLCUR_CURRENT_CDE and SORLFOS_CURRENT_CDE columns for admissions records.

  Banner views that query the Current Indicator will continue to execute the API functions that calculate the Current Indicator until this script has run to a successful completion. The SOBCTRL_CURRENT_ADM_CDE column is updated to *Y* to track the execution of this script.

  Once this script has been run, the Curriculum API procedure will return the SORLCUR_CURRENT_CDE, and the Field of Study API function will return the SORLFOS_CURRENT_CDE.

- `susoplccd_learner.sql` (Learner)

  This script updates the `SORLCUR_CURRENT_CDE` and `SORLFOS_CURRENT_CDE` columns for learner records and updates the `SORLUR_TERM_CODE_END` column where appropriate.

  Banner views that query the Current Indicator will continue to execute the API functions that calculate the Current Indicator until this script has run to a successful completion. The `SOBCTRL_CURRENT_LEARNER_CDE` column is updated to *Y* to track the execution of this script.

  Once this script has been run, the Curriculum API procedure will return the `SORLCUR_CURRENT_CDE`, and the Field of Study API function will return the `SORLFOS_CURRENT_CDE`.

- `susoplccd_outcome.sql` (Outcome)

  This script updates the `SORLCUR_CURRENT_CDE` and `SORLFOS_CURRENT_CDE` columns for outcome records.

  Banner views that query the Current Indicator will continue to execute the API functions that calculate the Current Indicator until this script has run to a successful completion. The `SOBCTRL_CURRENT_OUTCOME_CDE` column is updated to *Y* to track the execution of this script.

  Once this script has been run, the Curriculum API procedure will return the `SORLCUR_CURRENT_CDE`, and the Field of Study API function will return the `SORLFOS_CURRENT_CDE`.

Current Indicator for Outcome

The **Display current outcome curriculum by term** checkbox on SOACTRL is used to change the current outcome curriculum record order. The order is changed from by sequence number within priority to by term first and then by sequence number within priority. If the `susoplccd_outcome.sql` script has been run after this indicator has been set, the `susoplccd_outcome.sql` script must be rerun. If the conversion to use the current code on SORLCUR has not been run, the indicator can be checked, and all *OUTCOME* current indicators will be based on term first, and then on sequence number within priority.

Backfill Process

The `susoplccd_backfill.sql` script is executed from SQL*Plus and will execute the backfill process for a single Banner ID or for a population selection. The prompts include Banner ID or the population selection criteria: Application Code, Selection Identifier, Creator ID, and User ID. The script will execute the backfill process (`soklcur.p_backload_curr`) for all recruiting, application, degree, and learner records. This script is not run as part of the upgrade process.

The backfill process is used to update the `SORLCUR_CURRENT_CDE` column and the `SORLFOS_CURRENT_CDE` column. The `soklcur.p_backload_curr` procedure

executes the `soklcur.p_load_current` procedure to update the current code and learner end term. The `soklcur.p_backload_curr` procedure performs the following tasks:

1. The procedure checks for changes between the old and new curriculum values before calling the host DDL, and it only calls the backfill procedure if a change is found.

2. The `soklcur.p_load_current` procedure is called after the backfill process, and it updates the **Current (Indicator)** checkbox for *LEARNER* and non-*LEARNER* types with the Current Indicator setting from the Curriculum API and Field of Study API calculations. For *LEARNER* types, the following occurs to update the **End Term** field.

   2.1. All SORLCUR *LEARNER* records are read, including old, current calculations.

   2.2. If the Current Indicator is set to *Y*, and the `SORLCUR_CURRENT_CDE` is *null*, then the `SORLCUR_CURRENT_CDE` is set to *Y*.

   2.3. If the Current Indicator is set to *N*, the `SORLCUR_CURRENT_CDE` is *Y*, and the `SORLCUR_TERM_CODE` is less than the effective term, then the `SORLCUR_CURRENT_CDE` remains set to *Y*, and the `SORLCUR_TERM_CODE_END` is set to the effective term.

   2.4. The `SORLCUR_TERM_CODE_END` may be reset to *null*, if it is now current for the end term.

   2.5. The `SORLFOS_CURRENT_CDE` can remain set to *Y*, if the `SORLCUR_CURRENT_CDE` is *Y*, and the `SORLCUR_TERM_CODE_END` is populated.

Query for the Current Curriculum Using the CURRENT_CDE

Querying for current records uses the SOVLCUR view to check SOBCTRL and determine whether or not the conversion has taken place. If it has, the Curriculum API (`sb_curriculum`) returns the `SORLCUR_CURRENT_CDE` value, rather than calculating the setting of the Current Indicator.

Here is some sample code for how to query on the Current Indicator (`SORLCUR_CURRENT_CDE`).

For non-*LEARNER* types: (*RECRUIT*, *OUTCOME*, *ADMISSIONS*) the process is as follows:

```
Select * from sorlcur
   where nvl(sorlcur_current_cde, 'N') = 'Y'
```

For *LEARNER* types, the process compares the curriculum end term to the SGBSTDN record end term:

```
Select * from sorlcur, sgbstdn
where nvl(sorlcur_current_cde, 'N') = 'Y' and
sgbstdn_pidm = sorlcur_pidm
and sorlcur_lmod_code = sb_curriculum_str.f_learner and
sgbstdn_term_code_eff <  nvl(sorlcur_term_code_end,'999999')
and
sorlcur_term_code <
sb_learner.f_query_end(sorlcur_pidm, sgbstdn_term_code_eff);
```

If the user keeps all curriculum records synchronized with the SGBSTDN term, the select is similar to the non-learner curriculum process:

```
Select * from sorlcur, sgbstdn
where nvl(sorlcur_current_cde,'N') = 'Y' and
sgbstdn_pidm = sorlcur_pidm
and sorlcur_lmod_code = sb_curriculum_str.f_learner
sgbstdn_term_code_eff = sorlcur_term_code
```

Selection Examples Using Banner Views and API Cursors

The following examples for Recruit, Admissions, Learner, and Outcome show how to select the current and active concurrent curriculum rows using the Banner views and API cursors.

Recruit (SRBRECR) - Banner Views

These steps could be used in SQL*Plus and in views.

1. Select the recruit record.

```
Select srbrecr_pidm, srbrecr_term_code, srbrecr_admin_seqno
From srbrecr
Where .....
```

2. Select the current and active base concurrent curriculum rows.

```
Select * from sovlcur
Where sovlcur_pidm = (srbrecr_pidm)
And sovlcur_key_seqno = (srbrecr_admin_seqno)
And sovlcur_term_code = (srbrecr_term_code)
And sovlcur_lmod_code = sb_curriculum_str.f_recruit
```

```
And sovlcur_current_ind = 'Y'
And sovlcur_active_ind = 'Y'
Order by sovlcur_priority_no;
```

3. Select all current and active fields of study for the base curriculum.

```
Select * from sovlfos
Where sovlfos_pidm = (srbrecr_pidm)
And sovlfos_lcur_seqno = (sovlcur_seqno)
And sovlfos_active_ind = 'Y'
And sovlfos_current_ind = 'Y'
Order by decode(sovlfos_lfst_code,
sb_fieldofstudy_str.f_major,'1',
sb_fieldofstudy_str.f_minor,'2', sb_fieldofstudy_str.
f_concentration,'3','4'),
sovlfos_priority_no;
```

4. The SRVCCUR and SRVACUR views can also be used to select a recruiting curriculum record. It is not necessary to join SRVCCUR and SRVACUR to SRBRECR, because both views include all non-curricular SRBRECR columns.

   The following example uses SRVCCUR and will return all current and active curricula for a recruiting record.

```
Select srvccur.* from srvccur
Where srvccur_order > 0
And srvccur_pidm = (input parameter for srbrecr_pidm)
And srvccur_key_seqno = ( input parameter for
srbrecr_admin_seqno)
And srvccur_term_code = ( input parameter for
srbrecr_term_code);
```

   This example using SRVCCUR shows how to select the recruiting record, the curricula, and the primary major.

```
Select srvccur.*, sovcfos.*
from sovcfos, srvccur
Where sovcfos_lcur_seqno = srvccur_seqno
And sovcfos_lfst_code = sb_fieldofstudy_str.f_major
And sovcfos_order = 1
And sovcfos_pidm = srvccur_pidm
And srvccur_order > 0
And srvccur_pidm = (input parameter for srbrecr_pidm)
And srvccur_key_seqno = ( input parameter for
srbrecr_admin_seqno)
And srvccur_term_code = ( input parameter for
srbrecr_term_code);
```

Here is an example using SRVACUR to return the current and active curricula along with its top two majors, top two minors, and top six concentrations. This view does not require a join to SOVCFOS to retrieve the fields of study.

```
Select srvacur.*
From srvacur, srbrecr
Where srvacur_pidm = (input parameter for srbrecr_pidm)
And srvacur_key_seqno = ( input parameter for
srbrecr_admin_seqno)
And srvacur_term_code = ( input parameter for
srbrecr_term_code)
And srvacur_order > 0;
```

5. Select the current and active curriculum using the `CURRENT_CDE`.

   This method is dependent on running the `susoplccd_recruit.sql` conversion script to populate the `CURRENT_CDE`. All curriculum views use the `CURRENT_CDE`, once the script has been run.

```
Select srbrecr_pidm, srbrecr_term_code, srbrecr_admin_seqno,
Sorlcur_priority_no, Sorlcur_program, sorlcur_coll_code
From sobcact, sorlcur, srbrecr
Where sorlcur_pidm = srbrecr_pidm
And sorlcur_term_code = srbrecr_term_code
And sorlcur_key_seqno = srbrecr_admin_seqno
And nvl(sorlcur_current_cde,'N') = 'Y'
And sobcact_cact_code = sorlcur_cact_code
And sobcact_active_ind = 'Y'
And sorlcur_lmod_code =
   ( Select sb_curriculum_str.f_recruit from dual)
```

Recruit (SRBRECR) - API Cursors

The steps use the API cursors and are best used in processes.

1. Select the recruit record.
2. Select the base curriculum.
3. Select all fields of study for the base curriculum.

The following is sample code for the above three steps:

```
Declare
   Lv_recruit_cur sb_recruit.recruit_ref; /* cursor for recruit*/
   Lv_recruit_rec sb_recruit.recruit_rec;
```

```
   Lv_curriculum_cur b_curriculum.curriculum_ref; /* cursor for
   curriculum */
   Lv_curriculum_rec sb_curriculum.curriculum_rec;
   Lv_fieldofstudy_cur sb_fieldofstudy.fieldofstudy_ref;
   Lv_fieldofstudy_rec sb_fieldofstudy.fieldofstudy_rec;
   /* select the recruit record using API cursor */
   /* if you do not know the term and seq, use
   sb_recruit.f_query_all */
   /* you can send that cursor the pidm and optionally the term
   and seqno */
   Lv_recruit_cur := sb_recruit.f_query_one(p_pidm => :pidm,
         P_term_code => :term_code,
         P_admin_seqno => :admin_seqno);
   Fetch lv_recruit_cur into lv_recruit_rec;
   Close lv_recruit_cur;
   /* select the base curriculum for the recruit using the API
   query current cursor */
   Lv_curriculum_cur := sb_curriculum.f_query_current(p_pidm =>
   :pidm,
         P_term_code => lv_recruit_rec.r_term_code,
         P_keyseqno => lv_recruit_rec.r_admin_seqno,
         p_lmod_code => sb_curriculum_str.f_recruit,
         P_active_ind => 'Y');

Loop
   Fetch lv_curriculum_cur into lv_curriculum_rec;
   Exit when lv_curriculum_cur%notfound;
   <process lv_curriculum_rec.r_* elements >
   /* select the field of studies for the curriculum using the API
   query current cursor*/
   lv_fieldofstudy_cur := sb_fieldofstudy.f_query_current
         (p_pidm => :pidm,
         p_lcur_seqno => lv_curriculum_rec.r_seqno,
         p_active_ind => 'Y');
   loop
   fetch lv_fieldofstudy_cur into lv_fieldofstudy_rec;
   exit when lv_fieldofstudy_cur%notfound;
         <process lv_fieldofstudy_rec.r_* elements>
         end loop;
         close lv_fieldofstudy_cur;

end loop;

close lv_curriculum_cur;
```

```
end;
```

Admissions (SARADAP) - Banner Views

These steps could be used in SQL*Plus and in views.

1. Select the application.

```
Select saradap_pidm, saradap_term_code_entry, saradap_appl_no
From saradap
Where .....
```

2. Select the current and active base curriculum for the application.

```
Select * from sovlcur
Where sovlcur_pidm = (saradap_pidm)
And sovlcur_key_seqno = (saradap_appl_no)
And sovlcur_term_code = (saradap_term_code_entry)
And sovlcur_lmod_code = sb_curriculum_str.f_admissions
And sovlcur_current_ind = 'Y'
And sovlcur_active_ind = 'Y'
Order by sovlcur_priority_no;
```

3. Select all active and current fields of study for the base curriculum.

```
Select * from sovlfos
Where sovlfos_pidm = (saradap_pidm)
And sovlfos_lcur_seqno = (sovlcur_seqno)
And sovlfos_active_ind = 'Y'
And sovlfos_current_ind = 'Y'
Order by decode(sovlfos_lfst_code,
sb_fieldofstudy_str.f_major,'1',
sb_fieldofstudy_str.f_minor,'2', sb_fieldofstudy_str.
f_concentration,'3','4'),
sovlfos_priority_no;
```

4. The SAVCCUR and SAVACUR views can also be used to select admissions curriculum records. It is not necessary to join SAVCCUR and SAVACUR to SARADAP, because both views include all non-curricular SARADAP columns.

   The following example uses SAVCCUR and will return all current and active curricula for an application record.

```
Select savccur.* from savccur
Where savccur_order > 0
And savccur_pidm = (input parameter for saradap_pidm)
And savccur_key_seqno = ( input parameter for saradap_appl_no)
```

```
And savccur_term_code = ( input parameter for
saradap_term_code_entry);
```

This example using SAVCCUR shows how to select the application record, the curricula, and the primary major.

```
Select savccur.*, sovcfos.*
from sovcfos, savccur
Where sovcfos_lcur_seqno = savccur_seqno
And sovcfos_lfst_code = sb_fieldofstudy_str.f_major
And sovcfos_order = 1
And sovcfos_pidm = savccur_pidm
And savccur_order > 0
And savccur_pidm = (input parameter for saradap_pidm)
And savccur_key_seqno = ( input parameter for saradap_appl_no)
And savccur_term_code = ( input parameter for
saradap_term_code_entry);
```

Here is an example using SAVACUR to return the current and active curricula along with its top two majors, top two minors, and top six concentrations. This view does not require a join to SOVCFOS to retrieve the fields of study.

```
Select savacur.*
From savacur
Where savacur_pidm = (input parameter for saradap_pidm)
And savacur_key_seqno = ( input parameter for saradap_appl_no)
And savacur_term_code = ( input parameter for
saradap_term_code_entry)
And savacur_order > 0;
```

5. Select the current and active curriculum using the `CURRENT_CDE`.

This method is dependent on running the `susoplccd_adm.sql` conversion script to populate the `CURRENT_CDE`. All curriculum views use the `CURRENT_CDE`, once the script has been run.

```
Select saradap_pidm, saradap_term_code_entry,
saradap_appl_seqno,
Sorlcur_priority_no, Sorlcur_program, sorlcur_coll_code
From sobcact, sorlcur, saradap
Where sorlcur_pidm = saradap_pidm
And sorlcur_term_code = saradap_term_code_entry
And sorlcur_key_seqno = saradap_appl_seqno
And nvl(sorlcur_current_cde,'N') = 'Y'
And sobcact_cact_code = sorlcur_cact_code
And sobcact_active_ind = 'Y'
And  sorlcur_lmod_code =
   ( Select sb_curriculum_str.f_admissions from dual)
```

Admissions (SARADAP) - API Cursors

The steps use the API cursors and are best used in processes.

1. Select the applicant.
2. Select the base curriculum.
3. Select all fields of study for the base curriculum.

The following is sample code for the above three steps:

```
Declare
   Lv_appl_cur sb_admissionsapplication.admissionsapplication_ref;
   /* cursor */
   Lv_appl_rec sb_admissionsapplication.admissionsapplication_rec;
   Lv_curriculum_cur sb_curriculum.curriculum_ref; /* cursor */
   Lv_curriculum_rec sb_curriculum.curriculum_rec;
   Lv_fieldofstudy_cur sb_fieldofstudy.fieldofstudy_ref;
   Lv_fieldofstudy_rec sb_fieldofstudy.fieldofstudy_rec;
   /* select the applicant record using API cursor */
   /* If you do not know the term and application number, use */
   /* sb_admissionsapplication.f_query_all(p_pidm=> :pidm) */
   Lv_appl_cur := sb_admissionsapplication.f_query_one(p_pidm =>
   :pidm,
         P_term_code_entry => :term_code_entry,
         P_appl_no => :appl_no);
   Fetch lv_appl_cur into lv_appl_rec;
   Close lv_appl_cur;
   /* select the base curriculum for the applicant using the API
   query current cursor */
   Lv_curriculum_cur := sb_curriculum.f_query_current(p_pidm =>
   :pidm,
         P_term_code => lv_appl_rec.r_term_code_entry,
         P_keyseqno => lv_appl_rec.r_appl_no,
         p_lmod_code => sb_curriculum_str.f_admissions,
         P_active_ind => 'Y');

Loop
   Fetch lv_curriculum_cur into lv_curriculum_rec;
   Exit when lv_curriculum_cur%notfound;
   <process lv_curriculum_rec.r_* elements >
   /* select the field of studies for the curriculum using the API
   query current cursor*/
```

```
   lv_fieldofstudy_cur := sb_fieldofstudy.f_query_current
         (p_pidm => :pidm,
         p_lcur_seqno => lv_curriculum_rec.r_seqno,
         p_active_ind => 'Y');
   loop
         fetch lv_fieldofstudy_cur into lv_fieldofstudy_rec;
         exit when lv_fieldofstudy_cur%notfound;
         <process lv_fieldofstudy_rec.r_* elements>
   end loop;
   close lv_fieldofstudy_cur;

end loop;

close lv_curriculum_cur;

end;
```

Learner (SGBSTDN) - Banner Views

This step could be used in SQL*Plus and in views.

1. Select the learner curriculum.

```
Select sgbstdn_pidm, sgbstdn_term_code_eff,
From sgbstdn
Where .....
```

Requirements for Selecting the Correct Learner Effective Term

Insert a record into the session global temporary table for the learner concurrent curricula current calculation.

This task is only necessary if you want to select active concurrent curriculum records based on a specific SGBSTDN effective term. The process used to determine the Current Indicator setting will use the highest SGBSTDN effective term if this step is not performed.

Note

The SOTVCUR table is a global session temporary table. The records you insert are available only to your session.

```
soklcur.p_create_sotvcur(p_pidm => (sgbstdn_pidm),
            p_term_code => (sgbstdn_term_code_eff),
```

```
p_lmod_code => sb_curriculum_str.f_learner);
```

Note

If the SOTVCUR record is not created, the Current Indicator setting is going to be based on the highest SGBSTDN effective term. ■

The Current Indicator setting is based on the key column values for the module record. In the case of the learner record, there is no base except the effective term. The calculation of the Current Indicator setting for the learner curriculum is based on the highest SORLCUR sequence number for the lowest of the SGBSTDN effective terms, or the highest curriculum term for a given priority that is less than the SGBSTDN end term. The term added to SOTVCUR defines the SGBSTDN effective term being processed, the default term being the highest SGBSTDN term.

At the conclusion of your selection process, delete the SOTVCUR record that was created.

```
Delete sotvcur where sotvcur_pidm = (sgbstdn_pidm);
```

Selecting the Learner Curriculum

1. Select the base curriculum.

```
Select * from sovlcur
Where sovlcur_pidm = (sgbstdn_pidm)
And sovlcur_lmod_code = sb_curriculum_str.f_learner
And sovlcur_current_ind = 'Y'
And sovlcur_active_ind = 'Y'
Order by sovlcur_priority_no;
```

2. Select all fields of study for the base curriculum.

```
Select * from sovlfos
Where sovlfos_pidm = (sgbstdn_pidm)
And sovlfos_lcur_seqno = (sovlcur_seqno)
And sovlfos_active_ind = 'Y'
And sovlfos_current_ind = 'Y'
Order by decode(sovlfos_lfst_code,
sb_fieldofstudy_str.f_major,'1',
sb_fieldofstudy_str.f_minor,'2', sb_fieldofstudy_str.
f_concentration,'3','4'),
sovlfos_priority_no;
```

3. The SGVCCUR and SGVACUR views can also be used to select learner curriculum records. These views do not require that record be inserted into the SOTVCUR table to identify the effective term. It is not necessary to join SGBSTDN to SGVCCUR and SGVACUR, because both views include all non-curricular SGBSTDN columns.

   The following example uses SGVCCUR and will return all current and active curricula for the learner effective term.

```
Select sgvccur.* from sgvccur
Where sgvccur_order > 0
And sgvccur_pidm = (input parameter for sgradap_pidm)
And sgvccur_stdn_term_code_eff = ( input parameter for
sgbstdn_term_code_eff);
```

This example using SGVCCUR shows how to select the general student record for a specific effective term. This will select the current and active curricula for the effective term, as well as the primary major.

```
Select sgvccur.*, sovcfos.*
from sovcfos, sgvccur
Where sovcfos_lcur_seqno = sgvccur_seqno
And sovcfos_lfst_code = sb_fieldofstudy_str.f_major
And sovcfos_order = 1
And sovcfos_pidm = sgvccur_pidm
And sgvccur_order > 0
And sgvccur_pidm = (input parameter for sgradap_pidm)
And sgvccur_stdn_term_code_eff =
      (select max(m.sgbstdn_term_code_eff)
            from sgbstdn m
            where m.sgbstdn_pidm = sgvccur.sgvccur_pidm
            and m.sgbstdn_term_code_eff <= '&&TERM');
```

Here is an example using SGVCCUR to return the current and active curricula along with its top two majors, top two minors, and top six concentrations. This view does not require a join to SOVCFOS to retrieve the fields of study.

```
Select sgvccur.*
From sgvccur
Where sgvccur_pidm = (input parameter for sgbstdn_pidm)
And sgvccur_stdn_term_code_eff = ( input parameter for
sgbstdn_term_code_eff)
And sgvccur_order > 0;
```

4. Select the current and active curriculum using the `CURRENT_CDE`.

   This method is dependent on running the `susoplccd_learner.sql` conversion script to populate the `CURRENT_CDE`. All curriculum views use the `CURRENT_CDE`, once the script has been run.

   If your site maintains a one-to-one relationship between the SGBSTDN term and SORLCUR records, the following method can be used:

```
Select sgbstdn_pidm, sgbstdn_term_code_eff, sorlcur_appl_seqno,
Sorlcur_priority_no, Sorlcur_program, sorlcur_coll_code
From sobcact, sorlcur, sgbstdn
Where sorlcur_pidm = sgbstdn_pidm
```

```
And sorlcur_term_code = sgbstdn_term_code_eff
And nvl(sorlcur_current_cde,'N') = 'Y'
And sobcact_cact_code = sorlcur_cact_code
And sobcact_active_ind = 'Y'
And sorlcur_lmod_code =
   ( Select sb_curriculum_str.f_learner from dual)
```

If your site allows SGBSTDN to exist without corresponding SORLCUR records, the following must be used, which compares the SORLCUR_END_TERM to the SGBSTDN end term code:

```
Select sgbstdn_pidm, sgbstdn_term_code_eff, sorlcur_appl_seqno,
Sorlcur_priority_no, Sorlcur_program, sorlcur_coll_code
From sobcact, sorlcur, sgbstdn
Where sorlcur_pidm = sgbstdn_pidm
And sorlcur_term_code = sgbstdn_term_code_eff
And nvl(sorlcur_current_cde,'N') = 'Y'
And sobcact_cact_code = sorlcur_cact_code
And sobcact_active_ind = 'Y'
And sgbstdn_term_code_eff <
nvl(sorlcur_term_code_end,'999999')
And sorlcur_term_code <
   sb_learner.f_query_end(sorlcur_pidm, sgbstdn_term_code_eff)
And sorlcur_lmod_code =
   ( Select sb_curriculum_str.f_learner from dual)
```

Learner (SGBSTDN) - API Cursors

The steps use the API cursors and are best used in processes.

1. Select the learner.
2. Select the base curriculum.
3. Select all fields of study for the base curriculum.

The following is sample code for the above three steps:

```
Declare
   Lv_learner_cur sb_learner.learner_ref; /* cursor */
   Lv_learner_rec sb_learner.learner_rec;
   Lv_curriculum_cur sb_curriculum.curriculum_ref; /* cursor */
   Lv_curriculum_rec sb_curriculum.curriculum_rec;
   Lv_fieldofstudy_cur sb_fieldofstudy.fieldofstudy_ref;
   Lv_fieldofstudy_rec sb_fieldofstudy.fieldofstudy_rec;
```

```
   /* select the learner record using API cursor */
   /* if you do not know the term, use the sb_learner.f_query_all
   (p_pidm => :pidm) */
   /* to select all records for the person */
   Lv_learner_cur := sb_learner.f_query_one(p_pidm => :pidm,
         P_term_code_eff => :term_code);
   Fetch lv_learner_cur into lv_learner_rec;
   Close lv_learner_cur;
   /* select the base curriculum for the learner using the API
   query current cursor */
   Lv_curriculum_cur := sb_curriculum.f_query_current(p_pidm =>
   :pidm,
         p_lmod_code => sb_curriculum_str.f_learner,
         p_keyseqno => 99, /* fake seqno for learner */
         P_active_ind => 'Y'
         P_eff_term => lv_learner_rec.r_term_code_eff);

Loop
   Fetch lv_curriculum_cur into lv_curriculum_rec;
   Exit when lv_curriculum_cur%notfound;
   <process lv_curriculum_rec.r_* elements >
   /* select the field of studies for the curriculum using the API
   query current cursor*/
   lv_fieldofstudy_cur := sb_fieldofstudy.f_query_current
         (p_pidm => :pidm,
         p_lcur_seqno => lv_curriculum_rec.r_seqno,
         p_active_ind => 'Y');
   loop
         fetch lv_fieldofstudy_cur into lv_fieldofstudy_rec;
         exit when lv_fieldofstudy_cur%notfound;
         <process lv_fieldofstudy_rec.r_* elements>
   end loop;
   close lv_fieldofstudy_cur;

end loop;

close lv_curriculum_cur;

/* clean up temp table */

/* only your session has access to the records you inserted and the
table contents are completely removed when you log out */

delete sotvcur where sotvcur_pidm = :pidm;
```

```
end;
```

Outcome (SHRDGMR) - Banner Views

These steps could be used in SQL*Plus and in views.

1. Select the outcome.

```
Select shrdgmr_pidm, shrdgmr_seq_no
From shrdgmr
Where .....
```

2. Select the base curriculum.

```
Select * from sovlcur
Where sovlcur_pidm = (shrdgmr_pidm)
And sovlcur_key_seqno = (shrdgmr_seq_no)
And sovlcur_lmod_code = sb_curriculum_str.f_outcome
And sovlcur_current_ind = 'Y'
And sovlcur_active_ind = 'Y'
Order by sovlcur_priority_no;
```

3. Select all fields of study for the base curriculum.

```
Select * from sovlfos
Where sovlfos_pidm = (shrdgmr_pidm)
And sovlfos_lcur_seqno = (sovlcur_seqno)
And sovlfos_active_ind = 'Y'
And sovlfos_current_ind = 'Y'
Order by decode(sovlfos_lfst_code,
sb_fieldofstudy_str.f_major,'1',
sb_fieldofstudy_str.f_minor,'2', sb_fieldofstudy_str.
f_concentration,'3','4'),
sovlfos_priority_no;
```

4. The SHVCCUR and SHVACUR views can also be used to select outcome curriculum records. It is not necessary to join SHRDGMR to SHVCCUR and SHVACUR, because both views include all non-curricular SHRDGMR columns.

   The following example uses SHVCCUR and will return all current and active curricula for an outcome record.

```
Select shvccur.* from shvccur, shrdgmr
Where shvccur_order > 0
And shvccur_pidm = (input for SHRDGMR_PIDM)
And shvccur_key_seqno = (input for SHRDGMR_SEQ_NO);
```

This example using SHVCCUR shows how to select the outcome record, the curricula, and the primary major.

```
Select shvccur.*, sovcfos.*
from sovcfos, shvccur
Where sovcfos_lcur_seqno = shvccur_seqno
And sovcfos_lfst_code = sb_fieldofstudy_str.f_major
And sovcfos_order = 1
And sovcfos_pidm = shvccur_pidm
And shvccur_order > 0
And shvccur_pidm = (input for SHRDGMR_PIDM)
And shvccur_key_seqno = (input for SHRDGMR_SEQ_NO);
```

Here is an example using SHVACUR to return the current and active curricula along with its top two majors, top two minors, and top six concentrations. This view does not require a join to SOVCFOS to retrieve the fields of study.

```
Select shvacur.*
From shvacur
Where shvacur_pidm = (input for SHRDGMR_PIDM)
And shvacur_key_seqno = (input for SHRDGMR_SEQ_NO)
And shvacur_order > 0;
```

5. Select the current and active curriculum using the `CURRENT_CDE`.

   This method is dependent on running the `susoplccd_outcome.sql` conversion script to populate the `CURRENT_CDE`. All curriculum views use the `CURRENT_CDE`, once the script has been run.

```
Select shrdgmr_pidm, shrdgmr_seq_no,
Sorlcur_priority_no, Sorlcur_program, sorlcur_coll_code
From sobcact, sorlcur, shrdgmr
Where sorlcur_pidm = shrdgmr_pidm
And sorlcur_key_seqno = shrdgmr_seq_no
And nvl(sorlcur_current_cde,'N') = 'Y'
And sobcact_cact_code = sorlcur_cact_code
And sobcact_active_ind = 'Y'
And sorlcur_lmod_code =
   ( Select sb_curriculum_str.f_outcome from dual)
```

Outcome (SHRDGMR) - API Cursors

The steps use the API cursors and are best used in processes.

1. Select the outcome.

2. Select the base curriculum.

3. Select all fields of study for the base curriculum.

The following is sample code for the above three steps:

```
Declare
   Lv_outcome_cur sb_learneroutcome.learneroutcome_ref; /* cursor
   */
   Lv_outcome_rec sb_learneroutcome.learneroutcome_rec;
   Lv_curriculum_cur sb_curriculum.curriculum_ref; /* cursor */
   Lv_curriculum_rec sb_curriculum.curriculum_rec;
   Lv_fieldofstudy_cur sb_fieldofstudy.fieldofstudy_ref;
   Lv_fieldofstudy_rec sb_fieldofstudy.fieldofstudy_rec;
   /* select the outcome record using API cursor */
   /* the input parms pidm and seq_no are required.  If the seq_no
   is not known, */
   /* use f_query_all(p_pidm => :pidm) */
   Lv_outcome_cur := sb_outcome.f_query_one(p_pidm => :pidm,
         P_seq_no => :seq_no);
   Fetch lv_outcome_cur into lv_outcome_rec;
   Close lv_outcome_cur;
   /* select the base curriculum for the outcome using the API
   query current cursor */
   Lv_curriculum_cur := sb_curriculum.f_query_current(p_pidm =>
   :pidm,
         P_keyseqno => lv_outcome_rec.r_seq_no,
         p_lmod_code => sb_curriculum_str.f_outcome,
         P_active_ind => 'Y');

Loop
   Fetch lv_curriculum_cur into lv_curriculum_rec;
   Exit when lv_curriculum_cur%notfound;
   <process lv_curriculum_rec.r_* elements >
   /* select the field of studies for the curriculum using the API
   query current cursor*/
   lv_fieldofstudy_cur := sb_fieldofstudy.f_query_current
         (p_pidm => :pidm,
         p_lcur_seqno => lv_curriculum_rec.r_seqno,
         p_active_ind => 'Y');
   loop
         fetch lv_fieldofstudy_cur into lv_fieldofstudy_rec;
         exit when lv_fieldofstudy_cur%notfound;
         <process lv_fieldofstudy_rec.r_* elements>

end loop;
```

```
close lv_fieldofstudy_cur;

end loop;

close lv_curriculum_cur;

end;
```

Return the Row ID for Current and Active Records

The SOKLCUR package uses two functions to return the row IDs of the current and active curriculum records from SORLCUR (`soklcur.f_lcur_rowid`) and the field of study records from SORLFOS (`soklcur.f_lfos_rowid`).

The following example shows how to select the primary curriculum and the associated primary major for an application:

```
SELECT
   sorlcur_coll_code, sorlfos_majr_code,
   saradap_term_code_entry
   FROM   SORLFOS, SORLCUR, SARADAP
   WHERE
   sorlfos.rowid = soklcur.f_lfos_rowid
         ( sb_curriculum_str.f_admissions,
         sorlcur_seqno,
         sb_fieldofstudy_str.f_major,
         sorlcur_pidm,
         sorlcur_term_code,
         null,
         1)
   and sorlcur.rowid =
         soklcur.f_lcur_rowid(
         sb_curriculum_str.f_admissions,
         saradap_pidm,
         saradap_term_code_entry,
         saradap_appl_no,
         null,
         1)
   and  SARADAP_TERM_CODE_ENTRY = :term
   ORDER  BY  sorlcur_coll_code, sorlfos_majr_code,
         SARADAP_TERM_CODE_ENTRY;
```

The parameter arguments for the `soklcur.f_lcur_rowid` function are:

- Module
- PIDM
- Term code
- Key sequence number
- Order of the priority

The parameter arguments for the `sokclur.f_lfos_rowid` function are:

- Module
- SORLCUR sequence number
- Field of Study Type
- PIDM
- Term code
- Date (not used)
- Order of the priority

Return Select Column Values for Current and Active Records

The SOKCCUR package uses two functions to return select column values for the current and active curriculum records from SORLCUR (`sokccur.f_curriculum_value`) and the field of study records from SORLFOS (`sokccur.f_fieldofstudy_value`).

The following example shows how to select the primary curriculum level and degree for a learner:

```
SELECT  sgbstdn.sgbstdn_term_code_eff "Eff",
   SOKCCUR.f_curriculum_value ( sgbstdn.sgbstdn_pidm,
         sb_curriculum_str.f_learner,
         sgbstdn.sgbstdn_term_code_eff,
         99,
         sgbstdn.sgbstdn_term_code_eff,
         1,  /* designates the order */
         'LEVL' )  "Level" ,
   SOKCCUR.f_curriculum_value ( sgbstdn.sgbstdn_pidm,
         sb_curriculum_str.f_learner,
         sgbstdn.sgbstdn_term_code_eff,
```

```
      99,
      sgbstdn.sgbstdn_term_code_eff,
      1,  /* designates the order */
      'DEGC' )  "Degree"   ,
FROM sgbstdn
WHERE sgbstdn.sgbstdn_pidm = &pidm
and sgbstdn.sgbstdn_term_code_eff = '&term';
```

The following example shows how to select the primary major value for an application:

```
SELECT ---,
   substr(SOKCCUR.f_fieldofstudy_value(
      savccur_pidm, sb_curriculum_str.f_admissions,
      savccur_term_code, savccur_key_seqno, savccur_seqno,
      sb_fieldofstudy_str.f_major, 1, 'MAJR'),1,4) "MAJOR",
   sovccur_levl_code "LEVEL"
FROM SAVCCUR
WHERE SAVCCUR_PIDM = :pidm
```

The parameter arguments for the `sokccur.f_curriculum_value` function are:

- PIDM
- Module
- Term code
- Key sequence number
- Order of the priority
- Keyword for column

PRIOR	SORLCUR_PRIORITY_NO
ROLL	SORLCUR_ROLL_IND
CACT	SORLCUR_CACT_CODE
LEVL	SORLCUR_LEVL_CODE
DEGC	SORLCUR_DEGC_CODE
CTLG	SORLCUR_TERM_CODE_CTLG
MATRIC	SORLCUR_TERM_CODE_MATRIC
TADMIT	SORLCUR_TERM_CODE_ADMIT
ADMT	SORLCUR_ADMT_CODE
CAMP	SORLCUR_CAMP_CODE
PROGRAM	SORLCUR_PROGRAM
START	SORLCUR_START_DATE
END	SORLCUR_END_DATE
STYP	SORLCUR_STYP_CODE

GRADDATE	SORLCUR_EXP_GRAD_DATE
LEAV	SORLCUR_LEAV_CODE
FLEAV	SORLCUR_LEAV_FROM_DATE
TLEAV	SORLCUR_LEAV_TO_DATE
RATE	SORLCUR_RATE_CODE
GRADTERM	SORLCUR_TERM_CODE_GRAD
SITE	SORLCUR_SITE_CODE
ACYR	SORLCUR_ACYR_CODE
GAPP	SORLCUR_GAPP_SEQNO
ENDTERM	SORLCUR_TERM_CODE_END
COLL	SORLCUR_COLL_CODE

The parameter arguments for the `sokccur.f_fieldofstudy_value` function are:

- PIDM
- Module
- Term code
- Key sequence number
- SORLCUR sequence number
- Field of study type
- Order of the priority
- Keyword for column

MAJR	SORLFOS_MAJR_CODE
CRUL	SORLCUR_LFOS_RULE
DEPT	SORLFOS_DEPT_CODE
ACON	SORLFOS_MAJR_CODE_ATTACH

Using Custom Learner Module Codes

It is a recommended best practice to use the string functions found in the `sb_curriculum_str` and the `sb_fieldofstudy_str` APIs to return the value of the constants for the learner module codes and field of study types.

The functions and their returned values are as follows.

Function	Returned Value
sb_fieldofstudy_str.f_major	*MAJOR*
sb_fieldofstudy_str.f_minor	*MINOR*
sb_fieldofstudy_str.f_concentration	*CONCENTRATION*
sb_curriculum_str.f_recruit	*RECRUIT*
sb_curriculum_str.f_admissions	*ADMISSIONS*
sb_curriculum_str.f_learner	*LEARNER*
sb_curriculum_str.f_outcome	*OUTCOME*

Your institution may have a requirement to separate one of the learner modules by site, college, or campus and wish to track those records separately. An alternative is to implement VPD (Virtual Private Database) on all four modules, including the curriculum and field of study tables. If that is not possible, the string function provides a global that can store customized rules used to differentiate between your institution's records.

The baseline function to return the learner module value for Admissions is:

```
FUNCTION f_admissions RETURN VARCHAR2 IS
   BEGIN
         return sb_curriculum_str.ADMISSIONS;
   END;
```

You may need to change it to be:

```
FUNCTION f_admissions RETURN VARCHAR2 IS
   BEGIN
   IF <site code global variable> = 'X'
         return 'ADMISSIONSX';
   ELSE
         return 'ADMISSIONSY';
   END;
```

Be sure to update the maximum allowed counts on SOACTRL if you add additional learner module or field of study type codes.

Determining Primary and Secondary Curriculum

The priorities for a curriculum and field of study are stored as values in the SORLCUR_PRIORITY_NO and SORLFOS_PRIORITY_NO data columns. The values entered into the priority number fields can be for any number values, not just the typical values of *1*, *2*, *3*, or *4*. The primary curriculum is always the current and active curriculum with the lowest priority number. Using the following SQL statements, here is how you would select the primary curriculum.

```
Select * from sovlcur
Where sovlcur_pidm = (shrdgmr_pidm)
And sovlcur_key_seqno = (shrdgmr_seq_no)
And sovlcur_lmod_code = sb_curriculum_str.f_outcome
And sovlcur_current_ind = 'Y'
And sovlcur_active_ind = 'Y'
And sovlcur_priority_no =
      (select min(prim.sovlcur_priority_no)
      from sovlcur prim
      where prim.sovlcur_pidm = sovlcur.sovlcur_pidm
      and prim.sovlcur_key_seqno = sovlcur.sovlcur_key_seqno
      and prim.sovlcur_lmod_code = sovlcur.sovlcur_lmod_code
      and prim.sovlcur_active_ind = 'Y'
      and prim.sovlcur_current_ind = 'Y')
```

You can also use the views discussed previously to select primary and secondary curriculum information.

The above select becomes:

```
Select * from shvccur
Where shvccur_order = 1
And shvccur_pidm = <shrdgmr_pidm>
And shvccur_key_seqno = <shrdgmr_seq_no>
```

For recruits and applicants, include the term code in the select statement.

```
Select * from sovlcur
Where sovlcur_pidm = (srbrecr_pidm)
And sovlcur_term_code = (srbrecr_term_code)
And sovlcur_key_seqno = (srbrecr_admin_seqno)
And sovlcur_lmod_code = sb_curriculum_str.f_recruit
And sovlcur_current_ind = 'Y'
And sovlcur_active_ind = 'Y'
And sovlcur_priorityno =
      (select min(prim.sovlcur_priority_no)
```

```
from sovlcur prim
where prim.sovlcur_pidm = sovlcur.sovlcur_pidm
and prim.sovlcur_key_seqno = sovlcur.sovlcur_key_seqno
and prim.sovlcur_term_code = sovlcur.sovlcur_term_code
and prim.sovlcur_lmod_code = sovlcur.sovlcur_lmod_code
and prim.sovlcur_active_ind = 'Y'
and prim.sovlcur_current_ind = 'Y'
```

You can also use the views discussed previously to select primary and secondary curriculum information.

The above select becomes:

```
Select * from srvccur
Where srvccur_order = 1
And srvccur_pidm = <srbrecr_pidm>
And srvccur_term_code = <srbrecr_term_code>
And srvccur_key_seqno = <srbrecr_admin_seqno>;
```

For recruits and applicants, include the term code in the select statement.

```
Select * from sovlcur
Where sovlcur_pidm = (saradap_pidm)
And sovlcur_term_code = (saradap_term_code_entry)
And sovlcur_key_seqno = (saradap_appl_no)
And sovlcur_lmod_code = sb_curriculum_str.f_admissions
And sovlcur_current_ind = 'Y'
And sovlcur_active_ind = 'Y'
And sovlcur_priority_no =
      (select min(prim.sovlcur_priority_no)
      from sovlcur prim
      where prim.sovlcur_pidm = sovlcur.sovlcur_pidm
      and prim.sovlcur_key_seqno = sovlcur.sovlcur_key_seqno
      and prim.sovlcur_term_code = sovlcur.sovlcur_term_code
      and prim.sovlcur_lmod_code = sovlcur.sovlcur_lmod_code
      and prim.sovlcur_active_ind = 'Y'
      and prim.sovlcur_current_ind = 'Y'
```

You can also use the views discussed previously to select primary and secondary curriculum information.

The above select becomes:

```
Select * from savccur
Where savccur_order = 1
And savccur_pidm = <saradap_pidm>
```

```
And savccur_term_code = <sardap_term_code_entry>
And sarvccur_key_seqno = <sardap_appl_no>
```

To select the primary curriculum for the most recent learner record, you do not have to include the term and sequence number.

```
Select * from sovlcur
Where sovlcur_pidm = (sgbstdn_pidm)
And sovlcur_lmod_code = sb_curriculum_str.f_learner
And sovlcur_current_ind = 'Y'
And sovlcur_active_ind = 'Y'
And sovlcur_priority_no =
      (select min(prim.sovlcur_priorityy_no)
      from sovlcur prim
      where prim.sovlcur_pidm = sovlcur.sovlcur_pidm
      and prim.sovlcur_active_ind = 'Y'
      and prim.sovlcur_current_ind = 'Y'
      and prim.sovlcur_lmod_code = sovlcur.sovlcur_lmod_code);
```

If you are looking for the primary curriculum for a learner record that does not have the highest SGBSTDN effective term, you need to first create the SOTVCUR record.

```
      soklcur.p_create_sotvcur(p_pidm => (sgbstdn_pidm),
         p_term_code => (sgbstdn_term_code),
         p_lmod_code => sb_curriculum_str.f_learner);
Select * from sovlcur
Where sovlcur_pidm = (sgbstdn_pidm)
And sovlcur_lmod_code = sb_curriculum_str.f_learner
And sovlcur_current_ind = 'Y'
And sovlcur_active_ind = 'Y'
And sovlcur_priority_no =
      (select min(prim.sovlcur_priority_no)
      from sovlcur prim
      where prim.sovlcur_pidm = sovlcur.sovlcur_pidm
      and prim.sovlcur_lmod_code = sovlcur.sovlcur_lmod_code
      and prim.sovlcur_active_ind = 'Y'
      and prim.sovlcur_current_ind = 'Y'
      /* clean up temp table */
      delete sotvcur where sotvcur_pidm = :pidm;
```

You can also use the views discussed previously to select primary and secondary curriculum information. The requirement to insert the term into SOTVCUR is not required when using the SGVCCUR and SGVACUR views.

The above select for the primary curriculum for a general student record becomes:

```
Select * from sgvccur
Where sgvccur_order = 1
And sgvccur_pidm = <sgbstdn_pidm>
And sgvccur_stdn_term_code_eff = <sgbstdn_term_code_eff>
```

You can use also use the `soklcur.p_lcur_priority` procedure to determine the primary and secondary curriculum. This procedure returns the SORLCUR row information for a priority. The priority argument indicates which sequential priority you are querying. The number *1* indicates that you want the first (primary) priority. The number *2* indicates that you want the secondary priority. The actual numbering method for the priority numbers may not follow *1*, *2*, *3*, *4*, but a *1* in this argument returns the lowest priority, which is the primary priority.

The following is an example of how to use `p_lcur_priority`:

```
set serveroutput on
prompt Enter learner module code:
accept in_lmod char;
prompt Enter ID of person:
accept in_ID char;
prompt Enter term code on the module record:
accept in_term char;
prompt Enter key seqno for the module record (enter 99 for
learner):
accept in_keyseqno char;
declare
      lv_sorlcur_row sb_curriculum.curriculum_rec;
      lv_primary_rowid varchar2(18) := NULL;
      lv_secondary_rowid varchar2(18) := NULL;
      lv_pidm spriden.spriden_pidm%TYPE;
begin
      lv_pidm := gb_common.f_get_pidm('&in_ID');
      soklcur.p_create_sotvcur(p_pidm => lv_pidm,
            p_term_code => '&in_term',
            p_lmod_code => '&in_lmod');
      soklcur.p_lcur_priority(
            p_lmod => '&in_lmod',
            p_pidm => lv_pidm,
            p_term_code => '&in_term',
            p_keyseqno => &in_keyseqno,
            p_act_date => null,
            p_priority => 1,
            p_sorlcur_row => lv_sorlcur_row,
            p_lcur_ROWID => lv_primary_rowid) ;
```

```
        dbms_output.put_line('Primary curriculum seqno ' ||
              to_char(lv_sorlcur_row.r_seqno));
        soklcur.p_lcur_priority(
              p_lmod => '&in_lmod',
              p_pidm => lv_pidm,
              p_term_code => '&in_term',
              p_keyseqno => &in_keyseqno,
              p_act_date => null,
              p_priority => 2,
              p_sorlcur_row => lv_sorlcur_row,
              p_lcur_ROWID => lv_secondary_rowid) ;
        dbms_output.put_line('Secondary curriculum seqno ' ||
              to_char(lv_sorlcur_row.r_seqno));
              /* clean up temp table */
        delete sotvcur where sotvcur_pidm = :pidm;
end;
```

Inserting New Concurrent Curriculum Records

Conversion Procedure

The Banner 7.0 foundation release of Concurrent Curricula introduced an optional batch conversion process for moving the curriculum from the base tables to the concurrent curricula tables. The conversion is executed in all places in Banner 7.X where curriculum is maintained or concurrent curricula displayed, and prior to the display or maintenance activities. The same methodology for executing the conversion should be added in all enhancements or modifications to ensure data integrity.

The `soklcur.p_convert_curr` procedure converts curricula data from the SARADAP, SRBRECR, SGBSTDN, and SHRDGMR base tables to the SORLCUR and SORLFOS concurrent curricula tables. The arguments to the procedure are the PIDM, learner module code, and the ROWID for the module record.

Conversion Logic

Conversion occurs as follows.

1. The input module record (SARADAP, SRBRECR, SGBSTDN and SHRDGMR,) is read, and checks occur for the existence of any concurrent curricula rows for that

person and module. The conversion continues only if no concurrent curricula records already exist.

For example, when the procedure converts an application, it reads the SARADAP record and then checks for the existence of SORLCUR rows for the *ADMISSIONS* learner module. The process continues only if no *ADMISSIONS* rows exist on SORLCUR.

2. The primary and secondary curriculums and their majors, minors, and concentrations are pulled from SARADAP, SRBRECR, SGBSTDN, or SHRDGMR and inserted into the SORLCUR and SORLFOS tables.

Error Reporting

Three SQL*PLUS scripts are used to report data errors on SRBRECR, SARADAP, SGBSTDN, and SHRDGMR.

The scripts are as follows:

- `srsoplccv2.sql`

  This script reports majors, minors, and concentrations that are no longer valid on STVMAJR. In addition, it shows program codes that do not exist in SMRPRLE, level codes that are no longer valid on STVLEVL, college codes that are no longer valid on STVCOLL, degree codes that are no longer valid on STVDEGC, and campus codes that are no longer valid on STVCAMP.

- `srsoplccv.sql`

  This script reports missing key data such as:

  - the PIDM on SPRIDEN,
  - the level, college, degree, and major 1 on the primary curriculum on SARADAP, SHRDGMR, and SGBSTDN,
  - the level, degree, and major 1 on SRBRECR,
  - the level, college, degree, and major 1 on the secondary curriculum on SARADAP and SGBSTDN, and
  - the college and major 1 on the secondary curriculum on SHRDGMR.

- `srsorchlg.sql`

  This script reports discrepancies between the key backfilled data on SRBRECR, SARADAP, SGBSTDN, and SHRDGMR. Key backfill data includes the level, college, degree, program, campus, and the primary major. The college is not included in the comparison for SRBRECR.

Functions, Procedure, and Batch Process Used in Conversion

Four functions, a procedure, and a batch process are also used to assist in the concurrent curricula conversion. They are described below.

soklcur.f_convert_recruit

This function reads all SRBRECR records for a PIDM and calls the `soklcur.p_convert_curr` procedure for each recruit record that is selected. Dependencies exist between the admissions and recruit curriculum status codes. For this reason, the applicant records are always converted prior to converting the recruit records. This function first calls `soklcur.f_convert_applicant` so the dependencies are satisfied. The recruit conversion does not convert recruit curriculum records if any SORLCUR *RECRUIT* learner module rows exist for the person.

soklcur.f_convert_applicant

This function reads all SARADAP records for a PIDM and calls the `soklcur.p_convert_curr` procedure for each record that is selected. The admissions conversion does not convert application curriculum records if any SORLCUR *ADMISSIONS* learner module rows exist for the person.

soklcur.f_convert_learner

This function reads all SGBSTDN records for a PIDM and calls the `soklcur.p_convert_curr` procedure for each record that is selected. The learner conversion does not convert learner curriculum if any SORLCUR *LEARNER* learner module rows exist for the person.

soklcur.f_convert_outcome

This function reads all SHRDGMR records for a PIDM and calls the `soklcur.p_convert_curr` procedure for each record that is selected. Dependencies exist between the learner and outcome curriculum status codes. For this reason, the learner records are always converted prior to converting the outcome records. The outcome conversion function first calls `soklcur.f_convert_learner` to satisfy the dependency. The outcome conversion does not convert outcome curriculum if any SORLCUR *OUTCOME* learner module rows exist for the person.

soklcur.p_convert_curr

This procedures requires parameters for the PIDM, learner module code (STVLMOD), and the ROWID. The ROWID belongs to the SARADAP, SRBRECR, SGBSTDN, or SHRDGMR record. This procedure reads the module record for the PIDM and ROWID

and converts the curriculum to the new SORLCUR and SORLFOS tables. The conversion will not convert any records if SORLCUR rows exist for the person and the learner module code.

soplccv.pc

This is a batch process that executes `soklcur.p_convert_curr`. This process allows you to rerun a conversion. When run in rerun mode, the process deletes all concurrent curricula data for the PIDM and learner module, and starts the conversion from scratch. You can run SOPLCCV for all IDs by entering *%* in the Learner ID parameter.

Conversion Example

The following example will convert all curriculum records for a person. The `f_convert_recruit` function calls the `f_convert_admissions` function, and the `f_convert_outcome` function calls the `f_convert_learner` function.

```
declare
      outcome_conversion varchar2(1) := '';
      recruit_conversion varchar2(1) := '';
begin
      recruit_conversion := soklcur.f_convert_recruit(:pidm);
      outcome_conversion := soklcur.f_convert_outcome(:pidm);
      if recruit_conversion = 'Y' or outcome_conversion = 'Y'
then
            gb_common.p_commit();
      end if;
end;
```

The two other functions, `f_convert_admissions` and `f_convert_learner`, can be called independently if it is not necessary to convert all modules.

Banner Backfill Procedure

The Banner 7.0 foundation release of Concurrent Curricula introduced a process that is executed after all curriculums and fields of study are inserted, during the commit process. The `soklcur.p_backload_curr` process copies the active and current primary and secondary curriculum records from SORLCUR and SORLFOS back to the original curriculum data fields on SRBRECR, SARADAP, SGBSTDN, and SHRDGMR.

Please refer to the section that follows on “Special Requirements for Selecting the Learner Effective Term”, and follow the instructions for calling `soklcur.p_insert_sotvcur`.

The call to `p_backload_curr` is as follows:

```
Soklcur.p_backload_curr(p_lmod => (sb_curriculum_str.f_recruit
or
```

```
            f_admissions or f_learner or f_outcome),
      p_term_code => (term code value like
sgbstdn_term_code_eff,
            srbrecr_term_code, saradap_term_code and
            use '000000' for shrdgmr),
      p_keyseqno => (seqno like shrdgmr_seqno, saradap_appl_no,
            srbrecr_admin_seqno and '99' for sgbstdn),
      p_pidm => pidm value);
```

Special Requirements for Selecting the Learner Effective Term

The following is repeated information from the “Requirements for Selecting the Correct Learner Effective Term” section. This task may be required to determine the effective term of the curriculum. The step to insert the effective term into SOTVCUR is not required if you are using the SGVCCUR or SGVACUR views or the `p_eff_term` parameter for the `sb_curricuum.f_query_current` function.

Insert a record into the session global temporary table for the learner concurrent curricula current calculation. This task is only necessary if you want to select active concurrent curriculum records based on a specific SGBSTDN effective term. The process used to determine the Current Indicator setting will use the highest SGBSTDN effective term if this step is not performed.

Note

The SOTVCUR table is a global session temporary table. The records you insert are available only to your session.

```
soklcur.p_create_sotvcur(p_pidm => (sgbstdn_pidm),
      p_term_code => (sgbstdn_term_code_eff),
      p_lmod_code => sb_curriculum_str.f_learner);
```

Note

If the SOTVCUR record is not created, the Current Indicator setting is going to be based on the highest SGBSTDN effective term.

The Current Indicator setting is based on the key column values for the module record. In the case of the learner record, there is no base except the effective term. The calculation of the Current Indicator setting for the learner curriculum is based on the highest SORLCUR sequence number for the lowest of the SGBSTDN effective terms, or the highest curriculum term for a given priority that is less than the SGBSTDN end term. The term added to SOTVCUR defines the SGBSTDN effective term being processed, the default term being the highest SGBSTDN term.

At the conclusion of your selection process, delete the SOTVCUR record that was created.

```
Delete sotvcur where sotvcur_pidm = (sgbstdn_pidm);
```

Using APIs with Concurrent Curricula

Banner 7.0 introduced APIs for purposes of implementing the Unified Digital Campus. The advantage of using APIs is that Banner business logic is included in the validation and processing rules. Please refer to Chapter 3, “APIs”, for more information on “Curricula Checking and APIs” and “Curriculum Conversion Using Functions and APIs”.

The SORLCUR and SORLFOS concurrent curriculum tables use APIs. The following are examples of how to insert data using the APIs.

1. Create the recruit, application, learner, or degree record.
2. Create a primary curriculum.
3. Create a primary major.
4. Backfill the curriculum to the recruit, application, learner, or degree record.

The following example uses the learner record.

Note

The input variable definition is assumed to be part of this process, and all have a prefix of a colon (:). An example is “:pidm”. Most input parameters to `sb_learner.p_create` accept a null value.

The following is an example of inserting a record for a leaner.

```
Declare
   Out_rowid varchar2(18) := null;
   Lcur_out sorlcur.sorlcur_seqno%TYPE := null;
   Curr_error number(1) := null;
   Severity_out varchar2(1) := null;
   Lfos_seqno_out sorlfos.sorlfos_seqno%TYPE := null;
   Convert_outcome varchar2(1) := null;
   Convert_recruit varchar2(1) := null;

Begin
   Convert_recruit := soklcur.f_convert_recruit(:pidm);
   Convert_outcome := soklcur.f_convert_outcome(:pidm);
   Sb_learner.p_create (
         p_pidm => :pidm,
         p_term_code_eff => :term_code_new,
         p_stst_code => :stst_code,
         p_styp_code => :styp_code,
         p_exp_grad_date => :exp_grad_date,
         p_full_part_ind => :full_part_ind,
```

```
      p_sess_code => :sess_code,
      p_resd_code => :resd_code,
      p_orsn_code => :orsn_code,
      p_prac_code => :prac_code,
      p_advr_pidm => :advr_pidm,
      p_grad_credit_appr_ind=> :grad_credit_appr_ind,
      p_capl_code => :capl_code,
      p_leav_code => :leav_code,
      p_leav_from_date => lv_learner_rec.r_leav_from_date,
      p_leav_to_date => lv_learner_rec.r_leav_to_date,
      p_astd_code => :astd_code,
      p_term_code_astd => :term_code_astd,
      p_rate_code => :rate_cod,
      p_edlv_code => :edlv_cod,
      p_emex_code => :emex_code,
      p_aprn_code => :aprn_code,
      p_trcn_code => :trcn_code,
      p_gain_code => :gain_code,
      p_voed_code => :voed_code,
      p_blck_code => :blck_code,
      p_term_code_grad => :term_code_grad,
      p_acyr_code => :acyr_code,
      p_site_code => :site_code,
      p_egol_code => :egol_code,
      p_degc_code_dual => :degc_code_dual,
      p_levl_code_dual => :levl_code_dual,
      p_dept_code_dual => :dept_code_dual,
      p_coll_code_dual => :coll_code_dual,
      p_majr_code_dual => :majr_code_dual,
      p_bskl_code => :bskl_code,
      p_prev_code => '',
      p_term_code_prev => '',
      p_cast_code => '',
      p_term_code_cast => '',
      p_user_id => gb_common.f_sct_user,
      p_data_origin => gb_common.data_origin,
      p_rowid_out => out_rowid);
/* insert the effective term into the collector table */
soklcur.p_create_sotvcur(p_pidm => :term_code_new,
      p_term_code => :term_code_new,
      p_lmod_code => sb_curriculum_str.f_learner);
/* now create the primary curriculum */
```

```
/* override severity level will prevent fatal
curriculum errors from occurring */
sb_curriculum.p_create(p_pidm  => :pidm,
      p_seqno => '',
      p_lmod_code => sb_curriculum_str.f_LEARNER,
      p_term_code => :term_code_new,
      p_key_seqno => 99,
      p_priority_no => 1,
      p_roll_ind => 'D',
      p_cact_code => null,
      p_user_id => gb_common.f_sct_user,
      p_data_origin => data_common.data_origin,
      p_levl_code => :levl_code,
      p_coll_code => :coll_code,
      p_degc_code => :degc_code,
      p_term_code_ctlg=> :term_code_new,
      p_admt_code => :admt_code,
      p_term_code_admit => :term_code_new,
      p_term_code_matric => :term_code_new,
      p_camp_code => :camp_code,
      p_program => :program_1,
      p_curr_rule => null,
      p_rolled_seqno => null,
      p_rowid_out => out_rowid,
      p_curr_error_out => curr_error,
      p_seqno_out => lcur_seqno,
      p_override_severity => 'N',
      p_severity_out => severity_out
      p_styp_code => :styp_code,
      p_exp_grad_date => :exp_grad_date,
      p_leav_code => :leav_code,
      p_leav_from_date => :leav_from_date,
      p_leav_to_date => :leav_to_date,
      p_rate_code => :rate_code,
      p_term_code_grad => :term_code_grad,
      p_acyr_code => :acyr_code,
      p_site_code => :site_code
      );
/* now create the primary major */
SB_FIELDOFSTUDY.P_CREATE (P_PIDM => :pidm,
      P_LCUR_SEQNO => lcur_seqno,
      P_SEQNO => NULL,
```

```
      P_LFST_CODE => sb_fieldofstudy_str.f_major,
      P_TERM_CODE => :term_code_new,
      P_PRIORITY_NO => 1,
      P_CSTS_CODE => null,
      P_CACT_CODE => null,
      P_DATA_ORIGIN => data_common.data_origin,
      P_USER_ID => gb_common.f_sct_user,
      P_MAJR_CODE => :majr_code,
      P_TERM_CODE_CTLG => :term_code_new,
      P_TERM_CODE_END => null,
      P_DEPT_CODE => :dept_code,
      P_LFOS_RULE => null,
      P_CONC_ATTACH_RULE => null,
      P_START_DATE => :start_date,
      P_END_DATE => :end_date,
      P_TMST_CODE => :tmst_code,
      p_MAJR_CODE_ATTACH => null,
      p_rolled_seqno => NULL,
      P_ROWID_OUT => out_rowid,
      P_CURR_ERROR_OUT => curr_error,
      p_override_severity=> 'N',
      P_SEVERITY_OUT => severity_out,
      P_LFOS_SEQNO_OUT => lfos_seqno);
/* example of how to create a minor */
SB_FIELDOFSTUDY.P_CREATE (P_PIDM => :pidm,
      P_LCUR_SEQNO => lcur_seqno,
      P_SEQNO => NULL,
      P_LFST_CODE => sb_fieldofstudy_str.f_minor,
      P_TERM_CODE => :term_code_new,
      P_PRIORITY_NO => 1,
      P_CSTS_CODE => null,
      P_CACT_CODE => null,
      P_DATA_ORIGIN => data_common.data_origin,
      P_USER_ID => gb_common.f_sct_user,
      P_MAJR_CODE => :minor_code,
      P_TERM_CODE_CTLG => :term_code_new,
      P_TERM_CODE_END => null,
      P_DEPT_CODE => null,
      P_LFOS_RULE => null,
      P_CONC_ATTACH_RULE => null,
      P_START_DATE => :start_date,
      P_END_DATE => :end_date,
```

```
      P_TMST_CODE => :tmst_code,
      p_MAJR_CODE_ATTACH => null,
      p_rolled_seqno => NULL,
      P_ROWID_OUT => out_rowid,
      P_CURR_ERROR_OUT => curr_error,
      p_override_severity=> 'N',
      P_SEVERITY_OUT => severity_out,
      P_LFOS_SEQNO_OUT=> lfos_seqno);
/* example of how to create a concentration */
SB_FIELDOFSTUDY.P_CREATE (P_PIDM => :pidm,
      P_LCUR_SEQNO => lcur_seqno,
      P_SEQNO => NULL,
      P_LFST_CODE =>
      sb_fieldofstudy_str.f_concentration,
      P_TERM_CODE => :term_code_new,
      P_PRIORITY_NO => 1,
      P_CSTS_CODE => null,
      P_CACT_CODE => null,
      P_DATA_ORIGIN => data_common.data_origin,
      P_USER_ID => gb_common.f_sct_user,
      P_MAJR_CODE => :concentration_code,
      P_TERM_CODE_CTLG => :term_code_new,
      P_TERM_CODE_END => null,
      P_DEPT_CODE => null,
      P_LFOS_RULE => null,
      P_CONC_ATTACH_RULE => null,
      P_START_DATE => :start_date,
      P_END_DATE => :end_date,
      P_TMST_CODE => :tmst_code,
      /* insert the major that already exists
      that the conc is attached to, can be
      blank */
      p_MAJR_CODE_ATTACH => :majr_code,
      p_rolled_seqno => NULL,
      P_ROWID_OUT => out_rowid,
      P_CURR_ERROR_OUT => curr_error,
      p_override_severity=> 'N',
      P_SEVERITY_OUT => severity_out,
      P_LFOS_SEQNO_OUT => lfos_seqno);
/* you only need to create this record for the learner lmod */
soklcur.p_create_sotvcur(p_pidm => :pidm,
      p_term_code => :term_code_new,
```

```
         p_lmod_code => sb_curriculum_str.f_learner ;
         /* now run the backfill process - use appropriate lmod
   code,
         this example is specifically about the learner */
   Soklcur.p_backload_curr(p_lmod => sb_curriculum_str.f_learner,
         p_term_code => :term_code_new),
         p_keyseqno => 99,
         p_pidm => :pidm);
   /* clean up temp table */

delete sotvcur where sotvcur_pidm = :pidm;

end;

/
```

The following is an example of replacing the primary major for a learner in a new term.

```
EXEC SQL EXECUTE DECLARE
   Lv_curriculum_rec       sb_curriculum.curriculum_rec;
   Lv_curric_ref           sb_curriculum.curriculum_ref;
   Lv_fieldofstudy_rec     sb_fieldofstudy.fieldofstudy_rec;
   Lv_fieldofstudy_cur     sb_fieldofstudy.fieldofstudy_ref;
   Lv_Lcur_seqno           sorlcur.sorlcur_seqno%type;
   Lfos_rowid              VARCHAR2(18);
   Lfos_seqno              sorlfos.sorlfos_seqno%type;
   severity_level          VARCHAR2(1) := NULL;
   curr_err                NUMBER  := NULL;
   current_term            stvterm.stvterm_code%TYPE;
   cnter                   PLS_INTEGER := 0;
   cnter2                  PLS_INTEGER := 0;

BEGIN
   -- find max term for the new term
   current_term := sb_learner.f_query_current(p_pidm => :pidm,
   p_term_code_eff => :update_term);
   --- create new student record for the new term
   if current_term <> :update_term then
   sb_learner.p_copy_learner
      (p_pidm => :pidm,
      p_term_code_old => current_term,
      p_term_code_new => :update_term);
   end if;
   -- read primary curriculum record
```

```
lv_curric_ref := sb_curriculum.f_query_current
   (p_pidm => :pidm,
   p_lmod_code => sb_curriculum_str.f_learner,
   p_term_code => :update_term,
   p_active_ind => 'Y',
   p_eff_term => :update_term,
   p_keyseqno => 99);
fetch lv_curric_newref into lv_curriculum_rec;
close lv_curric_ref;
SB_CURRICULUM.P_CREATE(P_PIDM  => :pidm,
      P_SEQNO            => NULL,
      P_LMOD_CODE        => sb_curriculum_str.f_learner,
      P_TERM_CODE        => :update_term,
      P_KEY_SEQNO        => 99,
      P_PRIORITY_NO      => lv_curriculum_rec.r_priority_no,
      P_ROLL_IND         => lv_curriculum_rec.r_roll_ind,
      P_CACT_CODE        => '',
      P_USER_ID          => gb_common.f_sct_user,
      P_DATA_ORIGIN      => gb_common.data_origin,
      P_LEVL_CODE        => lv_curriculum_rec.r_levl_code,
      P_COLL_CODE        => lv_curriculum_rec.r_coll_code,
      P_DEGC_CODE        => lv_curriculum_rec.r_degc_code,
      P_TERM_CODE_CTLG   => lv_curriculum_rec.r_term_code_ctlg,
      P_CAMP_CODE        => lv_curriculum_rec.r_camp_code,
      P_PROGRAM          => lv_curriculum_rec.r_program,
      P_TERM_CODE_MATRIC => lv_curriculum_rec.r_term_code_matric,
      P_TERM_CODE_ADMIT  => lv_curriculum_rec.r_term_code_admit,
      P_ADMT_CODE        => lv_curriculum_rec.r_admt_code,
      P_CURR_RULE        => lv_curriculum_rec.r_curr_rule,
      P_site_code        => lv_curriculum_rec.r_site_code,
      P_rate_code        => lv_curriculum_rec.r_rate_code,
      P_leav_code        => lv_curriculum_rec.r_leav_code,
      P_leav_from_date   => lv_curriculum_rec.r_leav_from_date,
      P_leav_to_date     => lv_curriculum_rec.r_leav_to_date,
      P_acyr_code        => lv_curriculum_rec.r_acyr_code,
      P_term_code_grad   =>  lv_curriculum_rec.r_term_code_grad,
      P_exp_grad_date    =>  lv_curriculum_rec.r_exp_grad_date,
      P_styp_code        => :updated_styp,
      P_appl_key_seqno   => null,
      P_appl_seqno       => null,
      p_rolled_seqno     => null,
      p_gapp_seqno => lv_curriculum_rec.r_gapp_seqno,
```

```
      P_ROWID_OUT        => lfos_rowid,
      P_CURR_ERROR_OUT   => curr_err,
      p_seqno_out       => lv_lcur_seqno,
      P_override_severity => 'N',
      p_severity_out => severity_level);lv_fieldofstudy_cur :
      = sb_fieldofstudy.f_query_current
   (p_pidm => :pidm,
   p_lcur_seqno => :lcur_seqno,
   p_active_ind => 'Y');
LOOP
   FETCH lv_fieldofstudy_cur INTO lv_fieldofstudy_rec;
   EXIT WHEN lv_fieldofstudy_cur%NOTFOUND  ;
   SB_FIELDOFSTUDY.P_CREATE(P_PIDM   => :pidm,
      P_LCUR_SEQNO     => lv_lcur_seqno,
      P_SEQNO          => NULL,
      P_LFST_CODE      => lv_fieldofstudy_rec.r_lfst_code,
      P_TERM_CODE      => :update_term,
      P_PRIORITY_NO    => lv_fieldofstudy_rec.r_priority_no,
      P_CSTS_CODE      => '',
      P_CACT_CODE      => '',
      P_DATA_ORIGIN    => gb_common.data_origin,
      P_USER_ID        => gb_common.f_sct_user,
      P_MAJR_CODE      => lv_fieldofstudy_rec.r_majr_code,
      P_DEPT_CODE      => lv_fieldofstudy_rec.r_dept_code,
      P_TERM_CODE_CTLG => lv_fieldofstudy_rec.r_term_code_ctlg,
      P_LFOS_RULE      => lv_fieldofstudy_rec.r_lfos_rule,
      P_CONC_ATTACH_RULE  =>lv_fieldofstudy_rec.r_conc_attach_rule,
      p_rolled_seqno   => null,
      p_term_code_end   => lv_fieldofstudy_rec.r_term_code_end,
      p_start_date     => lv_fieldofstudy_rec.r_start_date,
      p_end_date       => lv_fieldofstudy_rec.r_end_date,
      p_tmst_code      => lv_fieldofstudy_rec.r_tmst_code,
      p_majr_code_attach => lv_fieldofstudy_rec.r_majr_code_attach,
      P_ROWID_OUT      => lfos_rowid,
      P_CURR_ERROR_OUT     => curr_err,
      P_override_severity => 'N',
      p_severity_out => severity_level,
      p_lfos_seqno_out => lfos_seqno);
END LOOP;
CLOSE lv_fieldofstudy_cur;
-- Now insert the new major
SB_FIELDOFSTUDY.P_CREATE(P_PIDM   => :pidm,
```

```
            P_LCUR_SEQNO     => lv_lcur_seqno,
            P_SEQNO          => NULL,
            P_LFST_CODE      => sb_fieldofstudy_str.f_major,
            P_TERM_CODE      => :update_term,
            P_PRIORITY_NO    => 1,
            P_CSTS_CODE      => '',
            P_CACT_CODE      => '',
            P_DATA_ORIGIN    => gb_common.data_origin,
            P_USER_ID        => gb_common.f_sct_user,
            P_MAJR_CODE      => :major_value,
            P_DEPT_CODE      => :dept_code,
            P_TERM_CODE_CTLG => lv_curriculum_rec.r_term_code_ctlg,
            P_LFOS_RULE      => null,
            P_CONC_ATTACH_RULE  => null,
            p_rolled_seqno   => null,
            p_term_code_end   => null,
            p_start_date     => null,
            p_end_date       => null,
            p_tmst_code      => null,
            p_majr_code_attach => null,
            P_ROWID_OUT      => lfos_rowid,
            P_CURR_ERROR_OUT     => curr_err,
            P_override_severity => 'N',
            p_severity_out => severity_level,
            p_lfos_seqno_out => lfos_seqno);
         soklcur.p_create_sotvcur(p_pidm => :pidm,
            p_term_code =>   :update_term,
            p_lmod_code => sb_curriculum_str.f_learner);
         soklcur.p_backload_curr(sb_curriculum_str.f_learner,
            :update_term, 99,  :pidm);

END;
```

The following is an example of inserting a record for an applicant.

```
Declare
   Out_rowid varchar2(18) := null;
   Lcur_out sorlcur.sorlcur_seqno%TYPE := null;
   Curr_error number(1) := null;
   Severity_out varchar2(1) := null;
   Lfos_seqno_out sorlfos.sorlfos_seqno%TYPE := null;
   Convert_outcome varchar2(1) := null;
   Convert_recruit varchar2(1) := null;
```

```
   adap_applno saradap.saradap_appl_no%TYPE := '';

Begin
   sb_admissionsapplication.p_create(p_pidm=> :pidm,
         p_term_code_entry => :term_code_entry,
         p_appl_no_inout => adap_applno,
         p_appl_date => trunk(systdate),
         p_apst_code => :apst_code,
         p_apst_date => :apst_date,
         p_maint_ind => :maint_ind,
         p_admt_code => :admt_code,
         p_styp_code => :styp_code,
         p_site_code => :site_code,
         p_resd_code => :resd_code,
         p_full_part_ind => :full_part_ind_in,
         p_sess_code => :sess_ code,
         p_wrsn_code => :wrsn_ code,
         p_intv_code => :intv_ code,
         p_fee_ind => :fee_ind_ind,
         p_fee_date => :fee_date,
         p_rate_code => :rate_ code,
         p_egol_code => :egol_code,
         p_edlv_code => :edlv_code,
         p_sbgi_code => :sbgi_code,
         p_recr_code => :recr_code,
         p_rtyp_code => :rtyp_ code,
         p_web_acct_misc_ind => data:web_acct_misc_ind,
         p_web_cashier_user => :web_cashier_user,
         p_web_trans_no => :web_trans_no,
         p_web_amount => :web_amount,
         p_web_receipt_number => :web_receipt_number,
         p_waiv_code => :waiv_code,
         p_data_origin => data_common.data_origin,
         p_user_id => gb_common.f_sct_user,
         p_rowid_out => out_rowid,
         p_appl_preference => :appl_preference
   );
   /* now create the primary curriculum */
   /* override severity level will prevent fatal
   curriculum errors from occurring */
   sb_curriculum.p_create(p_pidm => :pidm,
         p_seqno => '',
```

```
        p_lmod_code => sb_curriculum_str.f_admissions,
        p_term_code => :term_code_entry,
        p_key_seqno => adap_applno,
        p_priority_no => 1,
        p_roll_ind => 'D',
        p_cact_code => null,
        p_user_id => gb_common.f_sct_user,
        p_data_origin => data_common.data_origin,
        p_levl_code => :levl_code,
        p_coll_code => :coll_code,
        p_degc_code => :degc_code,
        p_term_code_ctlg=> :term_code_entry,
        p_camp_code => :camp_code,
        p_program => :program,
        p_curr_rule => null,
        p_rolled_seqno => null,
        p_rowid_out => out_rowid,
        p_curr_error_out => curr_error,
        p_seqno_out => lcur_seqno,
        p_override_severity => 'N',
        p_severity_out => severity_out,
);
/* now create the primary major */
SB_FIELDOFSTUDY.P_CREATE (P_PIDM => :pidm,
      P_LCUR_SEQNO => lcur_seqno,
      P_SEQNO => NULL,
      P_LFST_CODE => sb_fieldofstudy_str.f_major,
      P_TERM_CODE => :term_code_entry,
      P_PRIORITY_NO => 1,
      P_CSTS_CODE => null,
      P_CACT_CODE => null,
      P_DATA_ORIGIN => data_common.data_origin,
      P_USER_ID => gb_common.f_sct_user,
      P_MAJR_CODE => :majr_code,
      P_TERM_CODE_CTLG => :term_code_new,
      P_TERM_CODE_END => null,
      P_DEPT_CODE => :dept_code,
      P_LFOS_RULE => null,
      P_CONC_ATTACH_RULE => null,
      P_START_DATE => :start_date,
      P_END_DATE => :end_date,
      P_TMST_CODE => :tmst_code,
```

```
         p_MAJR_CODE_ATTACH => null,
         p_rolled_seqno => NULL,
         P_ROWID_OUT => out_rowid,
         P_CURR_ERROR_OUT => curr_error,
         p_override_severity=> 'N',
         P_SEVERITY_OUT => severity_out,
         P_LFOS_SEQNO_OUT => lfos_seqno);
   /* backfill the primary and secondary curriculum into saradap
   */

Soklcur.p_backload_curr(p_lmod => sb_curriculum_str.f_admissionsr,
         p_term_code => :term_code_entry),
         p_keyseqno => adap_applno,
         p_pidm => :pidm);

end;

/
```

Using SOKCCUR.p_match_curriculum to Find Curriculum or Field of Study Values

The `sokccur.f_match_curriculum` procedure can be used from within PL/SQL logic to determine whether a person has a curriculum record with a particular value. Only current and active curriculum records are examined. The value in the `p_prim_sec_cde` parameter indicates which curriculum record is to be examined (*P* for Primary, *A* for Any, *S* for Secondary). When a null value is submitted, all current and active curriculum records are examined.

Note

This procedure should not be used within a `WHERE` clause, but rather in a PL/SQL code block. ■

Here is an example in which the student's level is matched to *UG* for any curriculum occurrence.

```
   sokccur.f_match_curriculum(
         p_pidm             => :pidm,
         p_lmod_code        => :lmod_code,
         p_key_seqno        => :key_seqno,
         p_term_code        => :term_code,
         p_levl_code        =>  'UG',
         p_coll_code        =>  '',
         p_degc_code        =>  '',
         p_camp_code        =>   '',
```

```
        p_program          =>  '',
        p_prim_sec_cde     =>  '', -- A, P or S
        p_lfst_code        =>  '',
        p_majr_code        =>   '',
        p_dept_code        =>   '',
        p_admt_code        =>  '',
        p_term_code_admt   =>  '',
        p_term_code_matric => '' )  = 'Y' then
     return 'TRUE'
  else
     return 'FALSE'
  end if;
```

Streamlining Data Selection Through SOTCPRT

The SOTCPRT temporary table is used with Banner Student 8.X to help control the selection of curriculum data. This table is a session table, which means the contents are deleted when your Banner session is terminated.

The table description is as follows:

Column	Null?	Type
SOTCRPT_PIDM		NUMBER(8)
SOTCRPT_SEQNO		NUMBER(3)
SOTCRPT_TERM_CODE		VARCHAR2(6 CHAR)
SOTCRPT_KEY_SEQNO		NUMBER(2)
SOTCRPT_LMOD_CODE		VARCHAR2(30 CHAR)
SOTCRPT_COLL_CODE		VARCHAR2(2 CHAR)
SOTCRPT_DEGC_CODE		VARCHAR2(6 CHAR)
SOTCRPT_PROGRAM		VARCHAR2(12 CHAR)
SOTCRPT_LEVL_CODE		VARCHAR2(2 CHAR)
SOTCRPT_MAJR_CODE		VARCHAR2(4 CHAR)
SOTCRPT_DEPT_CODE		VARCHAR2(4 CHAR)
SOTCRPT_CAMP_CODE		VARCHAR2(3 CHAR)

The example using this table performs the following:

1. selects and inserts applications into SOTCRPT,
2. selects curriculum records in a separate select and updates SOTCRPT, and
3. queries SOTCRPT.

Here is the sample code for a simple histogram of the admissions curriculum.

```
delete sotcrpt;
variable term_code char(6)
begin
   :term_code :=  '&&term_code' ;
end;
/
insert into sotcrpt
(sotcrpt_term_code, sotcrpt_pidm, sotcrpt_key_seqno,
sotcrpt_seqno)
select
   saradap_term_code_entry, saradap_pidm, saradap_appl_no,
   rownum
   from saradap
   where saradap_term_code_entry = :term_code
   and saradap_admt_code = 'ST';
update sotcrpt
set sotcrpt_levl_code =
sokccur.f_curriculum_value(sotcrpt_pidm,
      sb_curriculum_str.f_admissions, sotcrpt_term_code,
      sotcrpt_key_seqno, '', 1, 'LEVL');
select sotcrpt_levl_code "LV", nvl(count(*),0)
   from sotcrpt
   group by sotcrpt_levl_code
   order by sotcrpt_levl_code ;
```

Using SAKMODS to Insert Recruiting, Admissions, and General Student Records

Another option exists that can be used to insert data into SRBRECR, SARADAP, SGBSTDN, and the curriculum records. Instead of using APIs, you can use the three procedures in the Application Maintenance Package (SAKMODS). They are:

- `sakmods.p_create_recruit`
- `sakmods.p_create_application`

- `sakmods.p_create_student`

These procedures contain long parameter lists that include the curriculum and field of study items individually. An additional parameter is used to override the curriculum severity check when the severity is set to *Fatal* on SOACTRL.

- Enter the value *F* for the override severity parameter to cause the process to abort if the curriculum is invalid.
- Enter the value *N* for the override severity parameter to allow the process to run to completion, even if the curriculum is not valid.

The procedures do not report warnings when the severity is set to *Warning* on SOACTRL.

The procedures use the field of study API (`sb_fieldofstudy`) to insert the first two majors, two minors, and six concentrations. The procedures do not insert the major the concentration is attached to on the concentration field of study row. The procedure parameters include just the concentration code value and not the attached to major value.

Procedure	Tables	Data Inserted
`sakmods.p_create_recruit`	SRBRECR	one recruiting record
	SORLCUR	one curriculum record
	SORLFOS	two majors, two minors, six concentrations
	SRRRSRC	recruiting source
	SORCCOL	communication plan collector records
	curriculum backfill	curriculum backfill

Procedure	Tables	Data Inserted
sakmods.p_create_application	SARADAP SORLCUR SORLFOS SARRSRC SARCHKL SORCCOL curriculum backfill	one application record two curriculum records two majors, two minors, six concentrations application source checklist items communication plan collector records curriculum backfill Applications are not created when an application hold exists for the person.
sakmods.p_create_student (Contains option to copy curricula and field of study from an application) **Note:** This procedure is not used in baseline. It was replaced with sakdscn.p_process_decision.	SGBSTDN SORLCUR SORLFOS SGRSATT SGRCHRT SORCCOL curriculum backfill	one general student record two curriculum records two majors, two minors, six concentrations copies data from SARSATT copies data from SARCHRT communication plan collector records curriculum backfill Curriculum and status information is copied from the application when the procedure is submitted with the application number.

Using SAKDCSN to Insert General Student Records from Admissions Applications

Another option exists that can be used to insert data from SARADAP into SGBSTDN. The sakdcsn.p_process_decision procedure in the Admissions Decision Package (SAKDCSN), is used in all Banner Student decision processing, except for the Quick Admit Form (SAAQUIK). The process requires that the application exist but then handles all processing required for decisions, including copying data from Admissions tables to General Student tables.

The parameters for the process are as follows:

- PIDM (Input value)
- Application Term Code (Input value)
- Application Number (Input value)
- Decision Code (STVAPDC) (Input value)
- Self Service Indicator (Y or N) (Input value)
- Fatal Curriculum Errors Allowed Indicator (Y or N) (Input value)
- Create the Communication Plan Indicator (Y or N) (Input value)

- Error Message Type (Returned value)
- Error Message (Returned value)
- Batch Error Message (Returned value, Upper case version of message)
- Maintenance Indicator (Input value, User Entered, System Generated, Batch Generated)
- Decision Date (Input value)

Procedure	Tables	Data Inserted
`sakdcsn.p_process_decision`	SARDCRV SGBSTDN SORLCUR SORLFOS SGRSATT SGRCHRT SORCCOL	Admissions decision General Student All curricula and fields of study are copied from the application attributes and cohorts attributes and cohorts attributes and cohorts

Set up Masking for the Curriculum Window

You can use the Data Display Mask Columns Form (GORDMCL) and the Data Display Mask Rules Form (GORDMSK) on select fields in the Curriculum window to mask fields (make fields invisible), specifically the non-required fields.

In the SORLCUR block of the Curriculum window, the following fields are required and should not be masked on the SRARECR, SRAQUIK, SAAADMS, SAAQUIK, SGASTDN, SFAREGS, or SHADEGR forms.

- `SORLCUR_CACT_CODE`
- `SORLCUR_PRIORITY_NO`
- `SORLCUR_DEGC_CODE`
- `SORLCUR_LEVL_CODE`
- `SORLCUR_COLL_CODE`

The `SORLCUR_TERM_CODE` field is required and enterable on SHADEGR and should not be masked on that form.

In the SORLFOS block in the Curriculum window, the following fields are required and should not be masked on the SRARECR, SRAQUIK, SAAADMS, SAAQUIK, SGASTDN, SFAREGS, or SHADEGR forms.

- SORLFOS_CACT_CODE
- SORLFOS_CSTS_CODE
- SORLFOS_LFST_CODE
- SORLFOS_PRIORITY_NO
- SORLFOS_LFOS_CODE

You can set up masking for specific fields on a form on GORDCML. You can set up masking rules for a form on GORDMSK. For example, to mask the SORLCUR_STYP_CODE field on SGASTDN, use GORDMCL to identify the object (*SGASTDN*), block (*SORLCUR*), and item (SORLCUR_STYP_CODE) to be masked. Use GORDMSK to set up the rule for the same object, block, and item and then define the rest of the masking characteristics and formatting for that rule.

5 System-Required Data

Overview

Banner® is a complex system with many parts that work together to manage your institution's data and to interact with users. When any one of the components of the system is missing, some of the system's functions may fail or may not work as intended.

In some cases, data itself can be considered an essential component of the system. The complete contents of certain tables, and specific rows in other tables, must be present for the system to work correctly. This special data is called *system-required data*. System-required data is a subset of the seed data delivered with a new Banner installation. Banner software releases often include seed data scripts that deliver additional system-required data.

Generally, Banner forms and processes will prevent you from deleting system-required data. But when you are using database tools or scripts to delete rows from the database—for example, during database cleanup to remove sample data before migrating into production—there is nothing to prevent system-required data from being accidentally deleted. In those situations, you should take care not to delete any system-required data.

This chapter lists system-required data for Banner Student. For more information on system-required data, please see the *Banner General Technical Reference Manual*.

System-Required Rows

System-required data rows in Banner Student are listed below. The list is organized alphabetically by table name. Some Banner General tables are listed first, as the values shown are used in Banner Student processing.

GTVLFST **Learner Field of Study Type Validation Form**

MAJOR	Major
MINOR	Minor
CONCENTRATION	Concentration

GTVMTYP **Meeting Type Validation Form**

CLAS	Classroom

GTVPRNT — Printer Validation Form

SHRPESE_PRNT	Printer setup for SHRPESE	Command - Not used but required for sleep/ wake
SHRPESI_PRNT	Printer setup for SHRPESI	Command - Not used but required for sleep/ wake

GTVPROC — Process Name Validation Form

WEBCCAPPDEP	Web Credit Card Application Deposit Process
WEBCCAPPFEES	Web Credit Card Application Fees Process
WEBCCENROLLDEP	Web Credit Card Enrollment Deposit
WEBCCEPRTREQ	Web Credit Card Enrollment Verification Charge
WEBCCGRADAPP	Web Credit Card Graduation Application Process
WEBCCREGFEES	Web Credit Card Registration Fees Process
WEBCCTRANSREQ	Web Credit Card Request Process

GTVSCHS — Scheduling Status Code Validation Form

NSM	Class needs a room assignment.
1SM	Class needs a room assignment and has a preferred first choice room indicated in the **Room Name** field. This code limits the initial pool of candidate rooms in the assignment algorithm.
WSM	Class needs a room assignment and must be assigned with the preceding NSM or 1SM record to the same room at the same time (cross-listed).
RSM	Class is related to the preceding NSM or 1SM record and must be assigned to the same room but not at the same days/time.
NXM	Class needs a room assignment and can share a room with another class whose times overlap with it (can be doubled-booked).
1XM	Class needs a room assignment, has a preferred first choice room indicated in the **Room Name** field, and can share a room with another class whose times overlap with it (can be double-booked).
RXM	Class is related to the previous NXM or 1XM record and must be assigned to the same room at the same or overlapping times.

GTVSCHS Scheduling Status Code Validation Form

ASM	Class has a room assignment that was made manually or in another system, such as the student information system.
AXM	Class has a room assignment that was made manually or in another system, and the class time span overlaps part of all of the time span of another class assigned to the same room (double-booking or intentional conflict).
HSM	This is a set of home cross-listed classes pre-assigned to the same room at identical days and times.
VSM	This is a set of visitor cross-listed classes pre-assigned to the same room at identical days and times.
5SM	Schedule25 assigned the class a room during a previous run.
5XM	Schedule25 assigned the class a room, and it is double-booked with another class.

STVACAT Degree Award Category Code Validation Form

11	Elementary School Program
12	Junior High School Program
13	High School Program
21	Post Secondary Certificate/Diploma < one year
22	Post Secondary Certificate/Diploma > one year and < two years
23	Associate's Degree
24	Baccalaureate Degree (Bachelor's)
25	Post Secondary Certificate/Diploma > two years and < four years
41	Post-Baccalaureate Certificate (Graduate)
42	Master's Degree
43, 32	Post-Masters's Certificate (Intermediate Graduate)
44	Doctoral Degree - Research/Scholarship
45, 31	Doctoral Degree - Professional Practice
46	Doctoral Degree - Other

STVACTP Activity Type Validation Form

SPRTS	Sports

STVACYR Academic Year Validation Form

0000	Beginning of Time
9999	End of Time

STVADDA Administrator Assignment Data Element Validation Form

Base Table	Data Element	Description	Validation Table
GORPGEO	GORPGEO_GEOD_CODE	Person Geod Code	STVGEOD
GORPGEO	GORPGEO_GEOR_CODE	Person Geor Code	STVGEOR
SORHSCH	GORSGEO_HSCH_GEOD_CODE	HSCH SBGI Geod Code	STVGEOD
SORHSCH	GORSGEO_HSCH_GEOR_CODE	HSCH SBGI Geor Code	STVGEOR
SORPCOL	GORSGEO_PCOL_GEOD_CODE	PCOL SBGI Geod Code	STVGEOD
SORPCOL	GORSGEO_PCOL_GEOR_CODE	PCOL SBGI Geor Code	STVGEOR
GORVISA	GORVISA_VTYP_CODE	Visa - Current	STVVTYP
SARAATT	SARAATT_ATTS_CODE	Applicant Attribute Code	STVATTS
SARADAP	SARADAP_ADMT_CODE	App Admit Code	STVADMT
SARADAP	SARADAP_CAMP_CODE	App Camp Code	STVCAMP
SARADAP	SARADAP_COLL_CODE_1	App Coll Code	STVCOLL
SARADAP	SARADAP_DEGC_CODE_1	App Degree Code	STVDEGC
SARADAP	SARADAP_DEPT_CODE	App Dept Code	STVDEPT
SARADAP	SARADAP_FULL_PART_IND	App Full/Part Time Ind	N/A
SARADAP	SARADAP_LEVL_CODE	App Level Code	STVLEVL
SARADAP	SARADAP_MAJR_CODE_1	App Major Code	STVMAJR
SARADAP	SARADAP_PROGRAM_1	App Program	SMRPRLE
SARADAP	SARADAP_RESD_CODE	App Residence Code	STVRESD
SARADAP	SARADAP_RTYP_CODE	App Recruit Type Code	STVRTYP
SARADAP	SARADAP_STYP_CODE	App Student Type Code	STVSTYP
SARADAP	SARADAP_TERM_CODE_ENTRY	Applicant Term code	STVTERM
SARADAP	SARADAP_LFST_CODE_1	App 1st Curric LFST Code	GTVLFST
SARADAP2	SARADAP_LFST_CODE_2	App 2nd Curric LFST Code	GTVLFST
SARADAP2	SARADAP_CAMP_CODE_2	App 2nd Curric Camp Code	STVCAMP
SARADAP2	SARADAP_COLL_CODE_2	App 2nd Curric Coll Code	STVCOLL
SARADAP2	SARADAP_DEGC_CODE_2	App 2nd Curr Degree Code	STVDEGC
SARADAP2	SARADAP_DEPT_CODE_2	App 2nd Curr Dept Code	STVDEPT
SARADAP2	SARADAP_LEVL_CODE_2	App 2nd Curr Level Code	STVLEVL
SARADAP2	SARADAP_MAJR_CODE_2	App 2nd Curr Major Code	STVMAJR
SARADAP2	SARADAP_PROGRAM_2	App 2nd Curr Program	SMRPRLE
SARCHRT	SARCHRT_CHRT_CODE	Applicant Cohort Code	STVCHRT
SORHSCH	SOBSBGI_HSCH_CITY	HSCH SBGI City	N/A
SORHSCH	SOBSBGI_HSCH_CNTY_CODE	HSCH SBGI County Code	STVCNTY
SORHSCH	SOBSBGI_HSCH_EPSC_CODE	HSCH SBGI EPS Code	STVEPSC
SORHSCH	SOBSBGI_HSCH_STAT_CODE	HSCH SBGI State Code	STVSTAT

STVADDA Administrator Assignment Data Element Validation Form

Base Table	Data Element	Description	Validation Table
SORHSCH	SOBSBGI_HSCH_ZIP	HSCH SBGI ZIP Code	GTVZIPC
SORPCOL	SOBSBGI_PCOL_CITY	PCOL SBGI City	N/A
SORPCOL	SOBSBGI_PCOL_CNTY_CODE	PCOL SBGI County Code	STVCNTY
SORPCOL	SOBSBGI_PCOL_EPSC_CODE	PCOL SBGI EPS Code	STVEPSC
SORPCOL	SOBSBGI_PCOL_STAT_CODE	PCOL SBGI State Code	STVSTAT
SORPCOL	SOBSBGI_PCOL_ZIP	PCOL SBGI ZIP Code	GTVZIPC
SORCONT	SORCONT_CTYP_CODE	Contact Code	STVCTYP
SORHSCH	SORHSCH_SBGI_CODE	HSCH SBGI Code	STVSBGI
SORINTS	SORINTS_INTS_CODE	Interest Code	STVINTS
SORPCOL	SORPCOL_PCOL_CODE	PCOL SBGI Code	STVSBGI
SORTEST	SORTEST_TEAC_CODE	Test Accommodation Code	STVTEAC
SORTEST	SORTEST_TEIN_CODE	Test Instrument Code	STVTEIN
SORTEST	SORTEST_TESC_CODE	Test Score Code	STVTESC
SPBPERS	SPBPERS_CITZ_CODE	Citizenship Code	STVCITZ
SPBPERS	SPBPERS_ETHN_CODE	Ethnic Code	STVETHN
SPBPERS	SPBPERS_LGCY_CODE	Legacy Code	STVLGCY
SPBPERS	SPBPERS_SEX	Gender Code	N/A
SPRIDEN	SPRIDEN_LAST_NAME	Last Name	N/A
SRBRECR	SRBRECR_ADMT_CODE	Recruit Admit Code	STVADMT
SRBRECR	SRBRECR_CAMP_CODE	Recruit Camp Code	STVCAMP
SRBRECR	SRBRECR_COLL_CODE	Recruit Coll Code	STVCOLL
SRBRECR	SRBRECR_DEGC_CODE	Recruit Degree Code	STVDEGC
SRBRECR	SRBRECR_DEPT_CODE	Recruit Dept Code	STVDEPT
SRBRECR	SRBRECR_LEVL_CODE	Recruit Level Code	STVLEVL
SRBRECR	SRBRECR_MAJR_CODE	Recruit Major Code	STVMAJR
SRBRECR	SRBRECR_PROGRAM_1	Recruit Program	SMRPRLE
SRBRECR	SRBRECR_RTYP_CODE	Recruit Type Code	STVRTYP
SRBRECR	SRBRECR_STYP_CODE	Recruit Student Type Code	STVSTYP
SRBRECR	SRBRECR_TERM_CODE	Recruit Term Code	STVTERM
SRBRECR	SRBRECR_LFST_CODE_1	Recruit 1st Curr LFST Code	GTVLFST
SRBRECR2	SRBRECR_LFST_CODE_2	Recruit 2nd Curr LFST Code	GTVLFST
SRBRECR2	SRBRECR_CAMP_CODE_2	Recruit 2nd Curr Camp Code	STVCAMP
SRBRECR2	SRBRECR_COLL_CODE_2	Recruit 2nd Curr Coll Code	STVCOLL

STVADDA Administrator Assignment Data Element Validation Form

Base Table	Data Element	Description	Validation Table
SRBRECR2	SRBRECR_DEGC_CODE_2	Recruit 2nd Curr Degree Code	STVDEGC
SRBRECR2	SRBRECR_DEPT_CODE_2	Recruit 2nd Curr Dept Code	STVDEPT
SRBRECR2	SRBRECR_LEVL_CODE_2	Recruit 2nd Curr Level Code	STVLEVL
SRBRECR2	SRBRECR_MAJR_CODE_2	Recruit 2nd Curr Major Code	STVMAJR
SRBRECR2	SRBRECR_PROGRAM_2	Recruit 2nd Curr Program	SMRPRLE
SRRCHRT	SRRCHRT_CHRT_CODE	Recruit Cohort Code	STVCHRT
SRRRATT	SRRRATT_ATTS_CODE	Recruit Attribute Code	STVATTS

STVAPLS EDI Application Source Code Validation Form

EDI	Electronic Data Interchange
WEB	World Wide Web

STVAPST Admissions Application Status Code Validation Form

C	Complete/Ready for Review
D	Decision Made
I	Incomplete/Items Outstanding

STVASTD Academic Standing Code Validation Form

00	Good Standing

STVATYP Address Type Code Validation Form

MA	Mailing MA is required for data load processing and is used by the Banner Advancement System for Banner Student loading processes.
PA	Parents PA is from the original system-required values used on SOAFOLK when entering a parent address.
BI	Billing
BU	Business

STVATYP Address Type Code Validation Form

BI and BU are required for sample data purposes if the Banner Finance System is installed.

XX	Accounts Payable

XX is reserved for use by TGRFEED.

STVCACT Curriculum Activity Status Validation Form

ACTIVE	Active Curriculum
INACTIVE	Inactive Curriculum
REMOVED	Removed

Note

Please note that STVCACT does not contain a system-required indicator. The indicator resides in the Curriculum Activity Status Rules Table (SOBCACT). The specified values must exist in the table when using curriculum processing.

STVCAST Combined Academic Standing Code Validation Form

00	Good Standing

STVCKSR Application Checklist Source Validation Form

BASELINE	Originates from Banner

STVCOLL College Code Validation Form

00	No College Designated
99	College Used in Statistical Calculation

STVCSTS — Curriculum Status Validation Form

ADMITREPLACE	Application replaces learner
APPACCEPT	Applicant Acceptance
APPLIED	Application Exists
AWARDED	Degree Awarded
CHANGED	Changed
COMPLETED	Completed
DENIED	Application has been rejected
INPROGRESS	In Progress
INSTACCEPT	Institution Acceptance
NOPUSH	Self-Service No Push to Learner
OVERLOAD	Overload
PENDING	Pending
REMOVED	Inactivate field of study
SOUGHT	Degree has sought status
STUDYPATH	Study Path Created

Note

Please note that STVCSTS shares these required values with the Curriculum Status Events Table (SORCSTS). The specified values must exist in the SORCSTS table when using curriculum status events and user-preferred translations for the value of the curriculum status code on STVCSTS.

STVDAYS — Days of the Week Validation Form

U	Sunday
M	Monday
T	Tuesday
W	Wednesday
R	Thursday
F	Friday
S	Saturday

STVDEGC — Degree Code Validation Form

000000	Degree Not Declared

STVDEGS — Degree Status Code Validation Form

SO	Sought

STVDEPT — Department Code Validation Form

0000	Department Not Declared

STVDFLT — Compliance Defaults Option Validation Form

ONLINE	Online processing default
BATCH	Batch processing default
WEB	Web processing default

STVEAPL — Electronic Application Status Validation Form

H	Admissions Hold
I	Can’t Insert Decision Code
R	Match Error
N	Not Processed
Y	Process Complete
P	Push Error
U	Suspended Record
V	Verification Error

STVEDIS — EDI Status Code Validation Form

P0	XML Request Received
P1	XML Transcript Exported
P2	XML Export had errors
P4	XML Export held for final grades
P5	XML Export held for awarded degree

STVEGRP — EDI Rule Group Validation Form

ACHR	AMCAS Course Summaries
ADDR	Address Source Rules
ADMS	Admission Rules
AGPA	AMCAS GPA Types
ATST	AMCAS Test Codes
ATYP	Address Type Rules
CHKL	Checklist Rules
CQLF	Code Qualifiers
CURR	Curriculum Rules
DCSN	Applicant Decision
DISP	Web Display Rules
DTQL	Date/Time Qualifier Rules

STVEGRP EDI Rule Group Validation Form

ENTY	Entity Rules
FLVL	Field of Study Level Rules
LGCY	Legacy Rules
PATH	System Path Rules
PCOL	Prior College Rules
PQLF	Phone Qualifier Code Rules
PREL	Electronic Prospect Group
QSTN	Question Code Rules
RESD	Residency Rules
RFQL	Reference Qualifier Rules
RLTN	Relationship Qualifier Rules
TELE	Telephone Patern Rules
VCRL	Verification Control Rules

STVELMT HTML Letter Module Validation Form

Module	Description	View
A	Admissions	AS_ADMISSIONS_APPLICANT
E	Electronic App	SAVEAPS
F	Registration	AS_STUDENT_REGISTRATION_DETAIL
P	Electronic Prospect	SRVPREL
R	Recruit	AS_RECRUITING_DATA
S	Student	AS_STUDENT_DATA
T	Transcripts	SHVTRE1

STVESTS Student Registration Status Code Validation Form

EL	Eligible to Register

STVETCT IPEDS Ethnic Code Validation Form

1	Black Non-Hispanic
2	American Indian or Alaskan Native
3	Asian or Pacific Islander
4	Hispanic
5	White Non-Hispanic
6	Other

STVEVEN — Academic History Event Code Validation Form

999	Miscellaneous Event

STVFTYP — Fee Type Validation Form

FLAT	Flat Fee
BILL	Per Bill Hour Fee
CRED	Per Credit Hour Fee
DURN	Per Duration Unit

STVGAST — Graduation Application Status Validation Form

AC	Active Application
IA	Inactivate Application

STVGCHG — Grade Change Code Validation Form

CC	Composite Calculation
CR	Capped Resit
DL	Degraded Late Mark
OE	Original Entry
RC	Re-Calculated
SG	Substitute Grade

STVINTS — Outside Interest Code Validation Form

A1	Instrumental Music
A2	Vocal Music
A3	Student Government
A4	Publications, Literary
A5	Debate
A6	Departmental Clubs
A7	Dramatics, Theater
A8	Religious Organizations
A9	Racial or Ethnic Organizations
AA	Intramural Athletics
AB	Varsity Athletics
AC	Political Organizations
AD	Radio - TV
AE	Fraternity or Sorority
AF	Special Interest Groups
AG	Campus or Community Service Groups

STVINTS **Outside Interest Code Validation Form**

AH	Art
AI	Coop or Internship Programs
AJ	Dance
AK	Environmental or Ecology Activity
AL	Foreign Study Abroad
AM	Honors or Independent Study
AN	ROTC

STVLEVL **Level Code Validation Form**

00	Level Not Declared

STVLMOD **Learner Module Validation Form**

RECRUIT	Recruiting
ADMISSIONS	Admissions
LEARNER	Student
OUTCOME	Student Outcome

Note

Please note that STVLMOD does not contain a system-required indicator. The indicator resides in the Learner Curriculum Frequency Rules Table (SOBLMOD). The specified values must exist in the table when using curriculum processing.

STVMAJR **Major Code Validation Form**

0000	Major Not Declared

STVMECL Mass Entry Column Validation Form

Mass Entry Code	Description	Mass Entry Forms Used
ACTC_CODE	Athletic - Sports Code	SGAMSPT
ADD_BILL_HRS	Add registration bill hours	SFAMREG
ADD_CAMP_CODE	Add registration campus code	SFAMREG
ADD_CREDIT_HRS	Add registration credit hours	SFAMREG
ADD_CRN	Add registration CRN	SFAMREG
ADD_END_DATE	Add registration end date	SFAMREG
ADD_GMOD_CODE	Add registration grade mode code	SFAMREG
ADD_INSM_CODE	Add registration instructional method code	SFAMREG
ADD_LEVL_CODE	Add registration level code	SFAMREG

STVMECL Mass Entry Column Validation Form

Mass Entry Code	Description	Mass Entry Forms Used
ADD_PTRM_CODE	Add registration part of term	SFAMREG
ADD_RSTS_CODE	Add registration status code	SFAMREG
ADD_SSTS_CODE	Add registration with registration status code	SFAMREG
ADD_START_DATE	Add registration start date	SFAMREG
ADD_TMST_HRS	Add registration with time status hours	SFAMREG
ADMR_CODE	Admissions checklist code	SAAMAPP
ADMT_CODE	Admissions Code	SAAMAPP
APDC_CODE	Decision code	SAAMAPP
APDC_CODE_MOST_RECENT	Most recent application decision	SAAMAPP
APDC_DATE	Decision date	SAAMAPP
APDC_IND	Current Decision: Accept; Reject; Confirm	SAAMAPP
APST_CODE	Application status	SAAMAPP
APST_DATE	Application status date	SAAMAPP
ATHL_AID_IND	Athletic - Aid Indicator	SGAMSPT
ATTS_CODE	Attribute	SAAMAPP, SGAMSTU
ATYP_CODE	Address type code	SHANDIP
BLCK_CODE	Block code	SFAMREG, SGAMSTU
BLOCK_PROCESS_IND	Block registration processing indicator	SFAMREG
BLOCK_RSTS_CODE	Block registration status code	SFAMREG
BYPASS_REG_ELIG	Bypass registration eligibility check	SFAMREG
CAP_ORDER_DATE	Cap-Gown-Hood tickets order date	SHAMUCA
CAP_ORDER_IND	Cap order indicator	SHAMUCA
CAP_PICKUP_IND	Cap picked up indicator	SHAMUCA
CAP_RETURN_DATE	Cap-Gown-Hood Returned date	SHAMUCA
CAP_RETURN_IND	Cap returned indicator	SHAMUCA
CAP_SIZE	Graduation Cap Size	SHAMUCA
CAP_TYPE	Cap Type	SHAMUCA
CERT_CODE	Ceremony code	SHAMCAT, SHAMUCA
CERT_ORDER_DATE	Order date	SHAMUCA
CERT_PICKUP_DATE	Pickup date	SHAMUCA
CERT_RETURN_DATE	Return date	SHAMUCA
CHKL_SOURCE	Checklist source	SAAMAPP
CHRT_CODE	Cohort code	SAAMAPP, SGAMSTU

STVMECL Mass Entry Column Validation Form

Mass Entry Code	Description	Mass Entry Forms Used
CKSR_CODE	Checklist source	SAAMAPP
CLAS_CODE	Class code	SFARMEG, SGAMSTU, SAAMAPP, SHAMDEG, SHAMCAT, SHAMDIP
CONFIRM	APDC IND, Student accept	SAAMAPP
CONFIRM_APDC_IND	Student accept	SAAMAPP
COPY_ACTC_CODE	Athletic Copy - Copy Sports Code	SGAMSPT
COPY_ATTR_COHORT	Copy attribute and cohort to new term	SGAMSTU
COPY_ATTRIBUTES	Athletic Copy - Copy All Attributes from Search Term	SGAMSPT
COPY_ELIG_CODE	Athletic Copy - Eligible	SGAMSPT
COPY_SAAT_CODE	Athletic Copy - Attribute Code	SGAMSPT
COPY_SAEL_CODE	Athletic Copy - Academic Eligibility	SGAMSPT
COPY_SPST_CODE	Athletic Copy - Status Code	SGAMSPT
CREATE_CERT_IND	Create Ceremony Attendance	SHAMIP
CREATE_DIPL_IND	Create Diploma	SHAMDIP
CRN_REG	Student is registered in CRN	SFAMREG
CRN_RSTS_CODE	Status code for currently registered CRN	SFAMREG
DEGS_CODE	Outcome status	SHAMDEG, SHAMCAT, SHAMUCA, SHAMDIP, SHAMUDI
DIPL_MAILED_DATE	Diploma mailed date	SHAMUDI
DIPL_ORDER_DATE	Diploma order date	SHAMDIP, SHAMUDI
DIPL_PICKUP_DATE	Diploma pick up date	SHAMUDI
DIPL_PICKUP_IND	Pick up indicator for diploma	SHAMUCA
DROP_ALL_CRN	Drop registration for all CRNs indicator	SFAMREG
DROP_CRN	Drop registration CRN	SFAMREG
DROP_CRSE_NUMB	Drop registration course number	SFAMREG
DROP_REMOVE_REG	Drop all registration records	SFAMREG
DROP_RSTS_CODE	Drop registration status on drop	SFAMREG
DROP_SEQ_NUMB	Drop registration sequence number	SFAMREG
DROP_SUBJ_CODE	Drop registration subject code	SFAMREG
EDLV_CODE	Education level	SGAMSTU
EGOL_CODE	Education goal	SGAMSTU
ELIG_CODE	Athletic - Eligible	SGAMSPT

STVMECL Mass Entry Column Validation Form

Mass Entry Code	Description	Mass Entry Forms Used
EXP_GRAD_DATE	Expected graduation date	SGAMSTU
FEE_AMOUNT	Fee Amount	SHAMDEG, SHAMDIP, SHAMCAT
FEE_DATE	Fee effective date	SHAMDEG, SHAMDIP, SHAMCAT
FEE_DETC_CODE	Fee detail code	SHAMDEG, SHAMDIP, SHAMCAT
FEE_IND	Y to charge fee; W to waive fee	SHAMDEG, SHAMDIP, SHAMCAT
FEE_TERM_CODE	Fee term code	SHAMDEG, SHAMDIP, SHAMCAT
GAST_CODE	Graduation Application Status	SHAMDEG
GMOD_CODE	CRN grade mode	SFAMREG
GOWN_ORDER_IND	Gown order indicator	SHAMUCA
GOWN_PICKUP_IND	Gown picked up indicator	SHAMUCA
GOWN_RETURN_IND	Gown returned indicator	SHAMUCA
GOWN_SIZE	Graduation Gown Size	SHAMUCA
GOWN_TYPE	Gown type	SHAMUCA
GRAD_ATTEND_CDE	Graduation Attend Value	SHAMCAT
GRAD_ATTEND_IND	Graduation Attend Ceremony Indicator	SHAMUCA
GRAD_DATE	Graduation date	SGAMSTU, SHAMDEG
GRAD_YEAR	Graduation year	SGAMSTU, SHAMDEG
GRST_CODE	Graduation status	SHAMDEG
HOOD_ORDER_IND	Hood order indicator	SHAMUCA
HOOD_PICKUP_IND	Hood picked up indicator	SHAMUCA
HOOD_RETURN_IND	Hood returned indicator	SHAMUCA
HOOD_TYPE	Hood type	SHAMUCA
INNM_CODE	Degree Awarded by Code	SHAMDIP
INSERT_ACTC_CODE	Athletic Insert - Sports Code	SGAMSPT
INTS_CODE	Interest code	SAAMAPP
LCUR_ADMIT_TERM	Learner admission term	SGAMSTU
LCUR_ADMT_CODE	Learner admission code	SGAMSTU
LCUR_CAMP_CODE	Campus	SAAMAPP, SFAMREG, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI

STVMECL Mass Entry Column Validation Form

Mass Entry Code	Description	Mass Entry Forms Used
LCUR_COLL_CODE	College	SAAMAPP, SFAMREG, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI
LCUR_CURRICULA	Which curricula to review: Primary, Secondary or All	SAAMAPP, SFAMREG, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI
LCUR_DEGC_CODE	Degree	SAAMAPP, SFAMREG, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI
LCUR_DEPT_CODE	Department	SAAMAPP, SFAMREG, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI
LCUR_LEVL_CODE	Level	SAAMAPP, SFAMREG, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI
LCUR_LFST_CODE	Field of study type	SAAMAPP, SFAMREG, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI
LCUR_MAJR_CODE	Field of study code	SAAMAPP, SFAMREG, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI
LCUR_MATRIC_TERM	Learner matriculation term	SGAMSTU
LCUR_PROGRAM	Program	SAAMAPP, SFAMREG, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI
LETR_CODE	Letter code	SAAMAPP, SGAMSPT, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI, SHAMUCA
LETR_DATE	Initiated date on the letter	SAAMAPP, SGAMSPT, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI, SHAMUCA

STVMECL	Mass Entry Column Validation Form

Mass Entry Code	Description	Mass Entry Forms Used
LETR_INIT_CODE	Letter initials code	SAAMAPP, SGAMSPT, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI, SHAMUCA
LETR_PRINT_DATE	Date letter was printed	SAAMAPP, SGAMSPT, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI, SHAMUCA
LETR_WAIT_DAYS	Wait days on the new letter	SAAMAPP, SGAMSPT, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI, SHAMUCA
MANDATORY_IND	Admissions checklist mandatory indicator	SAAMAPP
NUMBER_TICKETS	Number of Tickets	SHAMUCA
POPSEL_APPLICATION_ID	Population Selection Application	SAAMAPP, SGAMSPT, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI, SHAMUCA, SFAMREG
POPSEL_CREATOR_ID	Population Selection Creator ID	SAAMAPP, SGAMSPT, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI, SHAMUCA, SFAMREG
POPSEL_SELECTION_ID	Population Selection ID	SAAMAPP, SGAMSPT, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI, SHAMUCA, SFAMREG
POPSEL_USER_ID	Population Selection User ID	SAAMAPP, SGAMSPT, SGAMSTU, SHAMDEG, SHAMCAT, SHAMDIP, SHAMUDI, SHAMUCA, SFAMREG
RATE_CODE	Student fee assessment rate code	SGAMSTU, SAAMAPP, SFAMREG
REG_CHECK_APPR	Indicator to check for approvals	SFAMREG
REG_CHECK_ATTS	Indicator to check for student attributes	SFAMREG
REG_CHECK_CAMP	Indicator to check for campus	SFAMREG
REG_CHECK_CAPC	Indicator to check for capacity	SFAMREG

STVMECL Mass Entry Column Validation Form

Mass Entry Code	Description	Mass Entry Forms Used
REG_CHECK_CHRT	Indicator to check for cohort	SFAMREG
REG_CHECK_CLAS	Indicator to check for class	SFAMREG
REG_CHECK_COLL	Indicator to check for college	SFAMREG
REG_CHECK_CORQ	Indicator to check for corequisites	SFAMREG
REG_CHECK_DEGREE	Indicator to check for degree	SFAMREG
REG_CHECK_DEPT	Indicator to check for department	SFAMREG
REG_CHECK_DUPL	Indicator to check for duplicates	SFAMREG
REG_CHECK_HOLD	Indicator to check for holds	SFAMREG
REG_CHECK_LEVL	Indicator to check for level	SFAMREG
REG_CHECK_LINK	Indicator to check for linked courses	SFAMREG
REG_CHECK_MAJR	Indicator to check for fields of study	SFAMREG
REG_CHECK_MAXH	Indicator to check for maximum hours	SFAMREG
REG_CHECK_MEXC	Indicator to check for mutual exclusions	SFAMREG
REG_CHECK_MINH	Indicator to check for minimum hours	SFAMREG
REG_CHECK_PREQ	Indicator to check for prerequisites	SFAMREG
REG_CHECK_PROGRAM	Indicator to check for program	SFAMREG
REG_CHECK_REPT	Indicator to check for repeats	SFAMREG
REG_CHECK_RPTH	Indicator to check for repeat hours	SFAMREG
REG_CHECK_TIME	Indicator to check for time conflict	SFAMREG
REG_CURRENT_CRN	Currently in CRN	SFAMREG
REG_DATE	Registration date	SFAMREG
REG_FEE_ASSESSMENT	Process fees in batch or online	SFAMREG
REG_GRADE_MODE	Grade mode for currently registered CRN	SFAMREG
REJECT	ADPC IND, Student rejected	SAAMAPP
REJECT_ADPC_IND	Student rejected	SAAMAPP
RESD_CODE	Residence Code	SAAMAPP, SGAMSTU
RETURN_DATE	Tickets returned date	SHAMUCA
SAAT_CODE	Athletic - Attribute Code	SGAMSPT
SAEL_CODE	Athletic - Academic Eligibility	SGAMSPT
SBGI_INST_AWARD	Awarding institution	SHAMDIP
SEASON_USED_IND	Athletic - Season Used Indicator	SGAMSPT
SPST_CODE	Athletic - Status Code	SGAMSPT

STVMECL Mass Entry Column Validation Form

Mass Entry Code	Description	Mass Entry Forms Used
STST_CODE	Student status	SGAMSTU
STYP_CODE	Student type code	SAAMAPP, SGAMSTU
TERM_CODE_APPL	Admissions application term	SAAMAPP
TERM_CODE_ATHL_COMP	Athletic - Compliance Term Code	SGAMSPT
TERM_CODE_ATHL_COPY	Athletic - Copy Compliance Term Code	SGAMSPT
TERM_CODE_CERT	Ceremony term	SHAMCAT, SHAMDIP, SHAMUCA, SHAMUDI
TERM_CODE_DEGREE_COMPLETION	Term the degree is completed	SHAMDEG
TERM_CODE_EFF	Learner effective term	SGAMSTU, SFAMREG
TERM_CODE_EFF_IND	Term code equal indicator	SGAMSTU
TERM_CODE_GRAD	Graduation term	SHAMDEG, SGAMSTU, SHAMCAT, SHAMDIP
TERM_CODE_REG	Registration term	SFAMREG
TICKET_MAILED	Tickets mailed date	SHAMUCA
TICKET_ORDER_IND	Ticket order indicator	SHAMUCA
TICKET_PICKUP_IND	Ticket picked up indicator	SHAMUCA
WITHDRAWN_IND	Student withdrew	SAAMAPP

STVORIG Originator Code Validation Form

AUTO	Generated Automatically

Note

Please note that STVORIG does not contain a system-required indicator. However, it is recommended that the value *AUTO* exist when CAPP compliance processing is used at your institution. If this value does not exist, batch compliance will not run successfully. ■

STVPREV Progress Evaluation Code Validation Form

00	Good Standing

STVPROC Process Control Code Validation Form

COMPLIANCE	Compliance Requests
DISPLAYGRADES	Display Roster Grades
DISPLAYHOLDS	Display Student Holds
DISPLAYSCHEDULE	Display Student Schedule
DISPLAYTESTS	Display Test Scores

STVPROC — Process Control Code Validation Form

ENTERGRADES	Enter Roster Grades
TRANSCRIPT	Transcript Request

STVPTRM — Part of Term Code Validation Form

1	Full Term
C	Combined Parts of Term

STVRATP — Admissions Rating Type Validation Form

0000	Admissions Rating

STVRECR — Recruiter Code Validation Form

000	Recruiter Not Assigned (for Data Load Interface users only)

STVRESD — Residence Code Validation Form

0 †	Undeclared †*Note for clients in production prior to 1994.* If your institution is already using 0 (zero) for a value other than "Undeclared", you must assign one value which is not otherwise being used as a system-required "Undeclared" value.

STVRSTA — Recruiting Status Code Validation Form

00	Undetermined - New Recruit (for Data Load Interface users only)

STVRSTS — Course Registration Status Code Validation Form

RE	Registered
DD	Delete
DC	Drop Course
WL	Waitlisted

STVSBGI Source/Background Institution Code Validation Form

999999 Miscellaneous Institution

Note

Please note that STVSBGI does not contain a system-required indicator. However, it is recommended that the value *999999* exist when AMCAS processing and/or transfer course processing is used at your institution. If this value does not exist, processing may not run successfully. It is also recommended that the value *999999* exist on SHATRNS so you can enter a name for the institution when you are entering transfer work for a person.

STVSTSP Student Study Path Status Code Validation Form

AS	Active Study Path	Active and Allow Registration = checked (Yes) Y

STVSTST Student Status Code Validation Form

			Value in database	
AS	Active Student	Allow Registration = checked	(Yes)	Y
IS	Inactive Student	Allow Registration = unchecked	(No)	N

STVSTYP Student Type Code Validation Form

0 * Undeclared

* *Note for clients in production prior to 1994:*
If your institution is already using 0 (zero) for a value other than "Undeclared", you must assign one value which is not otherwise being used as a system-required "Undeclared" value.

STVTERM Term Code Validation Form

000000 The Beginning of Time

999999 The End of Time

Note

Term codes are numeric and are in the format YYYYTT. The codes must be constructed so that the codes maintain the appropriate sequence of terms. Term codes are displayed in descending order with the highest term first.

STVTESC **Test Code Validation Form**

Code	Test Type	Description	Number Positions	Data Type *	Min Score	Max Score
* The Data Type value of Numeric (checked) corresponds to a value of *N* in the database. The Data Type value of Alphanumeric (unchecked) corresponds to a value of *A* in the database.						
AA1	ASSET	Asset	2	checked	00	99
A01	ACT	English	2	checked	01	36
A02	ACT	Math	2	checked	01	36
A03	ACT	Reading	2	checked	01	36
A04	ACT	Science Reasoning	2	checked	01	36
A05	ACT	Composite	2	checked	01	36
A06	ACT	Sum of Standard Scores	3	unchecked	001	180
A07	ACT	ACT Combined English/Writing	2	checked	01	36
SWR	ACT	ACT Subscore Writing	2	checked	02	12
NEW	ACT	ACT Norm English/Writing	2	checked	01	99
NWR	ACT	ACT Norm Writing	2	checked	01	99
SUM	ACT	Subscore - Usage and Mech.	2	unchecked	01	18
SRH	ACT	Subscore - Rhetorical Skills	2	unchecked	01	18
SEA	ACT	Subscore - Elementary Algebra	2	unchecked	01	18
SAG	ACT	Subscore - Algebra/Geometry	2	unchecked	01	18
SGT	ACT	Subscore - Plane Geometry/Trig.	2	unchecked	01	18
SSS	ACT	Subscore - Social Studies	2	unchecked	01	18
SAL	ACT	Subscore - Arts/Literature	2	unchecked	01	18
NUM	ACT	Norm - Usage and Mech	2	unchecked	01	99
NRH	ACT	Norm - Rhetorical Skills	2	unchecked	01	99
NEA	ACT	Norm - Elementary Algebra	2	unchecked	01	99
NAG	ACT	Norm - Algebra/Geometry	2	unchecked	01	99
NGT	ACT	Norm - Plane Geometry/Trig	2	unchecked	01	99
NSS	ACT	Norm - Social Studies	2	unchecked	01	99
NAL	ACT	Norm - Arts/Literature	2	unchecked	01	99
G01	GMAT	GMAT Verbal Percent	2	checked	00	99
G02	GMAT	GMAT Quantitative Percent	2	checked	00	99
G03	GMAT	GMAT Total Converted Percent	3	checked	200	800
G04	GMAT	GMAT Writing Percent	2	checked	00	99
G05	GMAT	GMAT Total Percent	2	checked	00	99
G06	GMAT	GMAT Verbal Converted	2	checked	00	60
G07	GMAT	GMAT Quantitative Converted	2	checked	00	60
G08	GMAT	GMAT Writing Converted	2	checked	00	60
G09	GMAT	GMAT Integrated Reasoning Scor	2	checked	00	99
G10	GMAT	GMAT Integrated Reasoning Conv	1	checked	0	8

STVTESC **Test Code Validation Form**

Code	Test Type	Description	Number Positions	Data Type *	Min Score	Max Score
G03Q	GRE	Revised General Quantitative	3	checked	130	170
G03V	GRE	Revised General Verbal	3	checked	130	170
G03W	GRE	Revised General Writing	3	checked	0.0	6.0
GR01	GRE	Verbal Code	3	checked	200	800
GR02	GRE	Quantitative Code	3	checked	200	800
GR03	GRE	Analytical Code	3	checked	200	800
GR04	GRE	Writing Assessment	3	checked	0.0	6.0
GR05	GRE	Analytical Writing Test	3	checked	0.0	6.0
G22	GRE	Biochem, C & M Total Score	3	checked	200	990
G221	GRE	Biochem, C & M Subscore 1	3	checked	020	099
G222	GRE	Biochem, C & M Subscore 2	3	checked	020	099
G223	GRE	Biochem, C & M Subscore 3	3	checked	020	099
G24	GRE	Biology Total Score	3	checked	200	990
G241	GRE	Biology Subscore 1	3	checked	020	099
G242	GRE	Biology Subscore 2	3	checked	020	099
G243	GRE	Biology Subscore 3	3	checked	020	099
G27	GRE	Chemistry Total Score	3	checked	200	990
G271	GRE	Chemistry Subscore 1	3	checked	020	099
G272	GRE	Chemistry Subscore 2	3	checked	020	099
G273	GRE	Chemistry Subscore 3	3	checked	020	099
G29	GRE	Computer Science Total Score	3	checked	200	990
G291	GRE	Computer Science Subscore 1	3	checked	020	099
G292	GRE	Computer Science Subscore 2	3	checked	020	099
G293	GRE	Computer Science Subscore 3	3	checked	020	099
G2QE	GRE	GRE Quantitative Estimated Cur	3	checked	130	170
G2VE	GRE	GRE Verbal Estimated Current	3	checked	130	170
G31	GRE	Economics Total Score	3	checked	200	990
G311	GRE	Economics Subscore 1	3	checked	020	099
G312	GRE	Economics Subscore 2	3	checked	020	099
G313	GRE	Economics Subscore 3	3	checked	020	099
G34	GRE	Education Total Score *	3	checked	020	099
G341	GRE	Education Subscore 1 *	3	checked	020	099
G342	GRE	Education Subscore 2 *	3	checked	020	099
G343	GRE	Education Subscore 3 *	3	checked	020	099
G35	GRE	Rev. Education Total Score	3	checked	200	990
G351	GRE	Rev. Education Subscore 1	3	checked	020	099
G352	GRE	Rev. Education Subscore 2	3	checked	020	099
G353	GRE	Rev. Education Subscore 3	3	checked	020	099

STVTESC **Test Code Validation Form**

Code	Test Type	Description	Number Positions	Data Type *	Min Score	Max Score
G37	GRE	Engineering Total Score	3	checked	200	990
G371	GRE	Engineering Subscore 1	3	checked	020	099
G372	GRE	Engineering Subscore 2	3	checked	020	099
G373	GRE	Engineering Subscore 3	3	checked	020	099
G44	GRE	French Total Score *	3	checked	020	099
G441	GRE	French Subscore 1 *	3	checked	020	099
G442	GRE	French Subscore 2 *	3	checked	020	099
G443	GRE	French Subscore 3 *	3	checked	020	099
G46	GRE	Geography Total Score *	3	checked	020	099
G461	GRE	Geography Subscore 1 *	3	checked	020	099
G462	GRE	Geography Subscore 2 *	3	checked	020	099
G463	GRE	Geography Subscore 3 *	3	checked	020	099
G47	GRE	Geology Total Score	3	checked	200	990
G471	GRE	Geology Subscore 1	3	checked	020	099
G472	GRE	Geology Subscore 2	3	checked	020	099
G473	GRE	Geology Subscore 3	3	checked	020	099
G52	GRE	German Total Score *	3	checked	020	099
G521	GRE	German Subscore 1 *	3	checked	020	099
G522	GRE	German Subscore 2 *	3	checked	020	099
G523	GRE	German Subscore 3 *	3	checked	020	099
G57	GRE	History Total Score	3	checked	200	990
G571	GRE	History Subscore 1	3	checked	020	099
G572	GRE	History Subscore 2	3	checked	020	099
G573	GRE	History Subscore 3	3	checked	020	099
G64	GRE	Literature Total Score	3	checked	200	990
G641	GRE	Literature Subscore 1	3	checked	020	099
G642	GRE	Literature Subscore 2	3	checked	020	099
G643	GRE	Literature Subscore 3	3	checked	020	099
G67	GRE	Mathematics Total Score	3	checked	200	990
G671	GRE	Mathematics Subscore 1	3	checked	020	099
G672	GRE	Mathematics Subscore 2	3	checked	020	099
G673	GRE	Mathematics Subscore 3	3	checked	020	099
G68	GRE	Mathematics Rs Total Score	3	checked	200	990
G681	GRE	Mathematics Rs Subscore 1	3	checked	020	099
G682	GRE	Mathematics Rs Subscore 2	3	checked	020	099
G683	GRE	Mathematics Rs Subscore 3	3	checked	020	099
G71	GRE	Music Total Score *	3	checked	020	099
G711	GRE	Music Subscore 1 *	3	checked	020	099

STVTESC **Test Code Validation Form**

Code	Test Type	Description	Number Positions	Data Type *	Min Score	Max Score
G712	GRE	Music Subscore 2 *	3	checked	020	099
G713	GRE	Music Subscore 3 *	3	checked	020	099
G72	GRE	Music Total Score	3	checked	200	990
G721	GRE	Music Subscore 1	3	checked	020	099
G722	GRE	Music Subscore 2	3	checked	020	099
G723	GRE	Music Subscore 3	3	checked	020	099
G74	GRE	Philosophy Total Score *	3	checked	020	099
G741	GRE	Philosophy Subscore 1 *	3	checked	020	099
G742	GRE	Philosophy Subscore 2 *	3	checked	020	099
G743	GRE	Philosophy Subscore 3 *	3	checked	020	099
G77	GRE	Physics Total Score	3	checked	200	990
G771	GRE	Physics Subscore 1	3	checked	020	099
G772	GRE	Physics Subscore 2	3	checked	020	099
G773	GRE	Physics Subscore 3	3	checked	020	099
G78	GRE	Rev. Political Science Total Score	3	checked	200	990
G781	GRE	Rev. Political Science Subscore 1	3	checked	020	099
G782	GRE	Rev. Political Science Subscore 2	3	checked	020	099
G783	GRE	Rev. Political Science Subscore 3	3	checked	020	099
G79	GRE	Political Science Total Score *	3	checked	020	099
G791	GRE	Political Science Subscore 1 *	3	checked	020	099
G792	GRE	Political Science Subscore 2 *	3	checked	020	099
G793	GRE	Political Science Subscore 3 *	3	checked	020	099
G81	GRE	Psychology Total Score	3	checked	200	990
G811	GRE	Psychology Subscore 1	3	checked	020	099
G812	GRE	Psychology Subscore 2	3	checked	020	099
G813	GRE	Psychology Subscore 3	3	checked	020	099
G87	GRE	Sociology Total Score	3	checked	200	990
G871	GRE	Sociology Subscore 1	3	checked	020	099
G872	GRE	Sociology Subscore 2	3	checked	020	099
G873	GRE	Sociology Subscore 3	3	checked	020	099
G91	GRE	Spanish Total Score *	3	checked	020	099
G911	GRE	Spanish Subscore 1 *	3	checked	020	099
G912	GRE	Spanish Subscore 2 *	3	checked	020	099
G913	GRE	Spanish Subscore 3 *	3	checked	020	099
		** These codes are no longer used for the GRE.*				
MBS	MCAT	MCAT Biological Science Score	2	checked	00	99
MPS	MCAT	MCAT Physical Science Score	2	checked	00	99
MVR	MCAT	MCAT Verbal Score	2	checked	00	99

STVTESC **Test Code Validation Form**

Code	Test Type	Description	Number Positions	Data Type *	Min Score	Max Score
MWS	MCAT	MCAT Writing Score	2	unchecked	J	T
S01	SAT	Verbal Score	3	checked	200	800
S02	SAT	Mathematics Score	3	checked	200	800
S03	SAT	Reading Subscore	2	checked	20	80
S04	SAT	Vocabulary Subscore	2	checked	20	80
S07	SAT	SAT Writing	3	checked	200	800
S08	SAT	SAT Essay Subscore	2	checked	00	12
S09	SAT	SAT MC Subscore	2	checked	20	80
AH	SATII	American History/Social Studies	3	checked	200	800
BY	SATII	Biology	3	checked	200	800
CH	SATII	Chemistry	3	checked	200	800
CL	SATII	Chinese with Listening	3	checked	200	800
EB	SATII	Bio-Ecological Emphasis	3	checked	200	800
EH	SATII	European History/World Cultures	3	checked	200	800
EN	SATII	English Composition	3	checked	200	800
EP	SATII	English Proficiency	3	checked	200	800
ES	SATII	English Composition with Essay	3	checked	200	800
FL	SATII	French with Listening	3	checked	200	800
FR	SATII	French	3	checked	200	800
GL	SATII	German with Listening	3	checked	200	800
GM	SATII	German	3	checked	200	800
HB	SATII	Hebrew	3	checked	200	800
IT	SATII	Italian	3	checked	200	800
JL	SATII	Japanese with Listening	3	checked	200	800
KL	SATII	Korean with Listening	3	checked	200	800
LR	SATII	Literature	3	checked	200	800
LT	SATII	Latin	3	checked	200	800
MB	SATII	Bio-Molecular Emphasis	3	checked	200	800
M1	SATII	Mathematics Level I	3	checked	200	800
M2	SATII	Mathematics Level II	3	checked	200	800
1C	SATII	Mathematics Level IC	3	checked	200	800
2C	SATII	Mathematics Level IIC - Calculator	3	checked	200	800
MH	SATII	Modern Hebrew	3	checked	200	800
PH	SATII	Physics	3	checked	200	800
SL	SATII	Spanish with Listening	3	checked	200	800
SP	SATII	Spanish	3	checked	200	800
UH	SATII	U.S. History	3	checked	200	800

STVTESC **Test Code Validation Form**

Code	Test Type	Description	Number Positions	Data Type *	Min Score	Max Score
WH	SATII	World History	3	checked	200	800
WR	SATII	Writing	3	checked	200	800

Note

It is recommended that the listed values be checked periodically against the current delivered file layout values, since changes occur over time. ■

STVTMST **Time Status Code Validation Form**

99 Error Calculating Time Status

STVTPFD **Tape Field Names Validation Form**

*Field names with an * are no longer used for the ACT.*

A01_NATIONAL_NORM	ACT English National Norm
A01_TADM_CODE	ACT A01 Test Admin Code
A01_TEST_MON	ACT A01 Test Month
A01_TEST_SCORE	ACT A01 Test Score
A01_TEST_YEAR	ACT A01 Test Year
A02_NATIONAL_NORM	ACT Math National Norm
A02_TADM_CODE	ACT A02 Test Admin Code
A02_TEST_MON	ACT A02 Test Month
A02_TEST_SCORE	ACT A02 Test Score
A02_TEST_YEAR	ACT A02 Test Year
A03_NATIONAL_NORM	ACT Reading National Norm
A03_TADM_CODE	ACT A03 Test Admin Code
A03_TEST_MON	ACT A03 Test Month
A03_TEST_SCORE	ACT A03 Test Score
A03_TEST_YEAR	ACT A03 Test Year
A04_NATIONAL_NORM	ACT Science National Norm
A04_TADM_CODE	ACT A04 Test Admin Code
A04_TEST_MON	ACT A04 Test Month
A04_TEST_SCORE	ACT A04 Test Score
A04_TEST_YEAR	ACT A04 Test Year
A05_NATIONAL_NORM	ACT Composite National Norm
A05_TADM_CODE	ACT A05 Test Admin Code
A05_TEST_MON	ACT A05 Test Month
A05_TEST_SCORE	ACT A05 Test Score
A05_TEST_YEAR	ACT A05 Test Year

STVTPFD Tape Field Names Validation Form

A06_TADM_CODE	ACT A06 Test Admin Code
A06_TEST_MON	ACT A06 Test Month
A06_TEST_SCORE	ACT A06 Test Score
A06_TEST_YEAR	ACT A06 Test Year
A07_NATIONAL_NORM	ACT Combined English/Writing National Norm
A07_TEST_SCORE	ACT A07 Test Score
A1_INTS_CODE	A1 Interest Code Chosen
A2_INTS_CODE	A2 Interest Code Chosen
A3_INTS_CODE	A3 Interest Code Chosen
A4_INTS_CODE	A4 Interest Code Chosen
A5_INTS_CODE	A5 Interest Code Chosen
A6_INTS_CODE	A6 Interest Code Chosen
A7_INTS_CODE	A7 Interest Code Chosen
A8_INTS_CODE	A8 Interest Code Chosen
A9_INTS_CODE	A9 Interest Code Chosen
AA_INTS_CODE	AA Interest Code Chosen
AB_INTS_CODE	AB Interest Code Chosen
AC_INTS_CODE	AC Interest Code Chosen
ADDITIONAL_ID	Additional ID
ADDL_MCAT_INTENT_IND	MCAT Intent Indicator Placehld
ADM_ACTION_CODE	Admission Action Code Placehld
ADM_ACTION_DATE	Admission Action Date Placehld
ADVISOR_INFO_RELEASE	Advisor Info Release Placehold
AD_INTS_CODE	AD Interest Code Chosen
AE_INTS_CODE	AE Interest Code Chosen
AF_INTS_CODE	AF Interest Code Chosen
AGE	Age of Recruit/App Placeholder
AG_INTS_CODE	AG Interest Code Chosen
AH_INTS_CODE	AH Interest Code Chosen
AI_INTS_CODE	AI Interest Code Chosen
AJ_INTS_CODE	AJ Interest Code Chosen
AK_INTS_CODE	AK Interest Code Chosen
AL_INTS_CODE	AL Interest Code Chosen
AMCAS_ID	AMCAS ID
AM_INTS_CODE	AM Interest Code Chosen
AN_INTS_CODE	AN Interest Code Chosen
AO_INTS_CODE	AO Interest Code Chosen
APPL_TYPE	Application Type
APP_YEAR	Application Year

STVTPFD **Tape Field Names Validation Form**

AP_HOURS	Advanced Placement Hours
BIO_NUMBER	Number of the Applicant
BIRTH_CENT	Birth Date Century
BIRTH_CITY	City of Birth
BIRTH_COUNTY_COUNTRY_CODE	County or Country of Birth
BIRTH_COUNTY_COUNTRY_NAME	Birth Cnty/Ctry Name Placehold
BIRTH_COUNTY_RURAL_IND	Birth County is Rural
BIRTH_DATE	Full Birth Date
BIRTH_DAY	Day of Month of Birth
BIRTH_MON	Month of Birth
BIRTH_STATE	State or Province of Birth
BIRTH_YEAR	Year of Birth
CANADIAN_POSTAL_CODE	Canadian Postal Code
CANADIAN_PROVINCE	Canadian Province
CHANGE_FLAG	Change Flag Placeholder
CHANGE_NBR	Change Number Placeholder
CITIZENSHIP	Citizenship Code
CITIZEN_NATN_CODE	Nation Code of Citizenship
CITIZEN_NATN_NAME	Nation Name of Citize Placehld
CITY	Current Address City
CLEP_HOURS	College Level Exam Prog Hours
CNTY_CODE	Current Address County Code
CUM_AO_GPA	Cumulative All Other GPA
CUM_AO_HOURS	Cumulative All Other Hours
CUM_BCPM_GPA	Cumulative BioChemPhysMath GPA
CUM_BCPM_HOURS	Cumulative BioChemPhysMath Hrs
CUM_TOTAL_GPA	Cumulative Total GPA
CYCLE_ADD	Cycle Add Placeholder
CYCLE_CHANGE	Cycle Change Placeholder
DATE_APPLIED	Application Date
DEGC_CODE	Degree Code
DEPT_CODE	Department Code
DISADVANTAGED_IND	Disadvantaged Indicator
EARLY_DECISION	Early Decision Indicator
EDUC_GOAL	Education Goal
EFFECTIVE_DATE	Effective Date of Test
EMAIL_ADDR	Email Address
ENROLLMENT_DATE	Intended Enrollment Date
ETHN_CODE	Ethnic Code 1
ETHN_CODE10	Ethnic Code 10

ETHN_CODE2	Ethnic Code 2
ETHN_CODE3	Ethnic Code 3
ETHN_CODE4	Ethnic Code 4
ETHN_CODE5	Ethnic Code 5
ETHN_CODE6	Ethnic Code 6
ETHN_CODE7	Ethnic Code 7
ETHN_CODE8	Ethnic Code 8
ETHN_CODE9	Ethnic Code 9
EVALUATION	Evaluation Placeholder
EXP_GRAD_DATE	Expect Graduation Date Holder
FEE_WAIVER	Fee Waiver
FELONY	Felony Indicator
FELONY_PLACE_HOLDER	Felony Placeholder
FOREIGN_ADDRESS	Foreign Address Indicator
FOREIGN_NATION	Foreign Nation
FOREIGN_SSN	Foreign SSN
FR_AO_GPA	Freshman All Other GPA
FR_AO_HOURS	Freshman All Other Hours
FR_BCPM_GPA	Freshman BioChemPhysMath GPA
FR_BCPM_HOURS	Freshman BioChemPhysMath Hrs
FR_TOTAL_GPA	Freshman Total GPA
G01_TADM_CODE	GMAT G01 Test Admin Code
G01_TEST_MON	GMAT G01 Test Month
G01_TEST_SCORE	GMAT G01 Test Score
G01_TEST_YEAR	GMAT G01 Test Year
G02_TEST_SCORE	GMAT G02 Test Score
G03_TEST_SCORE	GMAT G03 Test Score
G04_TEST_SCORE	GMAT G04 Test Score
G05_TEST_SCORE	GMAT G05 Test Score
G06_TEST_SCORE	GMAT G06 Test Score
G07_TEST_SCORE	GMAT G07 Test Score
G08_TEST_SCORE	GMAT G08 Test Score
G09_TEST_SCORE	GMAT G09 Test Score
G10_TEST_SCORE	GMAT G10 Test Score
GENDER	Gender
GRAD_AO_GPA	Graduate All Other GPA
GRAD_AO_HOURS	Graduate All Other Hours
GRAD_BCPM_GPA	Graduate BioChemPhysMath GPA
GRAD_BCPM_HOURS	Graduate BioChemPhysMath Hrs
GRAD_TOTAL_GPA	Graduate Total GPA

GRE_EST_CURR0	GRE Estimated Current 0
GFRE_EST_CURR1	GRE Estimated Current 1
GRE_PERCENTILE0	GRE Percentile 0
GRE_PERCENTILE1	GRE Percentile 1
GRE_PERCENTILE2	GRE Percentile 2
GRE_PERCENTILE3	GRE Percentile 3
GRE_SCORE0	GRE Score 0
GRE_SCORE1	GRE Score 1
GRE_SCORE2	GRE Score 2
GRE_SCORE3	GRE Score 3
GRE_TEST_CENT	GRE Test Century
GRE_TEST_CODE	GRE Test Code
GRE_TEST_MON	GRE Test Month
GRE_TEST_SCORE	GRE Test Score
GRE_TEST_YEAR	GRE Test Year
GRE_TYPE0	GRE Type 0
GRE_TYPE1	GRE Type 1
GRE_TYPE2	GRE Type 2
GRE_TYPE3	GRE Type 3
HISP_IND	Hispanic Indicator
HSCH_AO_GPA	High School All Other GPA
HSCH_AO_HOURS	High School All Other Hours
HSCH_BCPM_GPA	High School ChemBioPhyMath GPA
HSCH_BCPM_HOURS	High School ChemBioPhyMath Hrs
HSCH_CITY	High School City
HSCH_CLASS_RANK	High School Class Rank
HSCH_CLASS_SIZE	High School Class Size
HSCH_GPA	High School Total GPA
HSCH_GRADE_LEVEL	High School Grade Level
HSCH_GRAD_CENT	High School Graduation Century
HSCH_GRAD_DAY	High School Graduation Day
HSCH_GRAD_MON	High School Graduation Month
HSCH_GRAD_YEAR	High School Graduation Year
HSCH_NAME	High School Name
HSCH_NATION	High School Nation
HSCH_SBGI_CODE	High School SBGI Code
HSCH_STAT_CODE	High School State Code
HSCH_STREET1	High School Street Address 1
HSCH_TOTAL_HOURS	High School Total Hours
HSCH_ZIP	High School ZIP Code

INSTITUTIONAL_ACTION	Institutional Action
INTEREST	Interest
INTS_CODE1	Interest Code 1
INTS_CODE2	Interest Code 2
INTS_CODE3	Interest Code 3
INTS_CODE4	Interest Code 4
INTS_CODE5	Interest Code 5
INTS_CODE6	Interest Code 6
INTS_CODE7	Interest Code 7
INTS_CODE8	Interest Code 8
INTS_CODE9	Interest Code 9
INTS_CODE10	Interest Code 10
INTS_CODE11	Interest Code 11
INTS_CODE12	Interest Code 12
INTS_CODE13	Interest Code 13
INTS_CODE14	Interest Code 14
JR_AO_GPA	Junior All Other GPA
JR_AO_HOURS	Junior All Other Hours
JR_BCPM_GPA	Junior BioChemPhysMath GPA
JR_BCPM_HOURS	Junior BioChemPhysMath Hours
JR_TOTAL_GPA	Junior Total GPA
LEGAL_COUNTY_CODE	Legal County Code
LEGAL_COUNTY_NAME	Legal County Name Placeholder
LEGAL_COUNTY_RURAL_IND	Legal County Rural Indicator
LEGAL_STATE	Legal State Code
MAJR_CODE	Major Code 1
MAJR_CODE2	Major Code 2
MAJR_CODE3	Major Code 3
MAJR_CODE4	Major Code 4
MAJR_CODE5	Major Code 5
MAJR_CODE6	Major Code 6
MAJR_PSAT	PSAT Major Code
MATRIC_DATE	Matriculation Date Placeholder
MBS_LITHOCODE	MCAT BioSci Lcode Placehold
MBS_MED_MAR_IND	MCAT BioSci Med Mar Placehold
MBS_TEAC_CODE	MCAT BioSci Test Accommodation
MBS_TEFR_CODE	MCAT BioSci Test Form Code
MBS_TEST_DATE	MCAT MBS Test Date
MBS_TEST_SCORE	MCAT MBS Test Score
MBS_UNUSED1	MCAT BioSci Unused Placeholder

MBS_UNUSED2	MCAT BioSci Unused Placeholder
MBS_UNUSED3	MCAT BioSci Unused Placeholder
MBS_WS_FORM_NUM	MCAT Writing Sample Placehold
MCAT_INTENT_DATE	MCAT Intent Test Date
MILITARY_DISCHARGE_IND	Military Discharge Indicator
MILITARY_HON_DISCHARGE_IND	Military Honorable Dischg Ind
MILITARY_SERVICE_IND	Military Service Indicator
MISDEMEANOR_IND	Misdemeanor Indicator
MPS_TEST_SCORE	MCAT MPS Test Score
MVR_TEST_SCORE	MCAT MVR Test Score
MWS_TEST_SCORE	MCAT MWS Test Score
NAG_NATIONAL_NORM	ACT Alg/Coord Geom National Norm
NAG_TADM_CODE	ACT NAG Test Admin Code
NAG_TEST_MON	ACT NAG Test Month
NAG_TEST_SCORE *	ACT NAG Test Score
NAG_TEST_YEAR	ACT NAG Test Year
NAL_NATIONAL_NORM	ACT Arts/Lit National Norm
NAL_TADM_CODE	ACT NAL Test Admin Code
NAL_TEST_MON	ACT NAL Test Month
NAL_TEST_SCORE *	ACT NAL Test Score
NAL_TEST_YEAR	ACT NAL Test Year
NAME_FIRST	First Name
NAME_LAST	Last Name
NAME_LAST_OLD	Former Last Name
NAME_MI	Middle Initial
NATION_CODE	Nation Code
NATION_NAME	Nation Name
NEA_NATIONAL_NORM	ACT Elem Algebra National Norm
NEA_TADM_CODE	ACT NEA Test Admin Code
NEA_TEST_MON	ACT NEA Test Month
NEA_TEST_SCORE *	ACT NEA Test Score
NEA_TEST_YEAR	ACT NEA Test Year
NEW_TADM_CODE	ACT NEW Test Admin Code
NEW_TEST_MON	ACT NEW Test Month
NEW_TEST_SCORE *	ACT NEW Test Score
NEW_TEST_YEAR	ACT NEW Test Year
NEXT_MCAT_DATE	Next MCAT Date (pre-2008)
NGT_NATIONAL_NORM	ACT Plane Geom/Trig National Norm
NGT_TADM_CODE	ACT NGT Test Admin Code
NGT_TEST_MON	ACT NGT Test Month

NGT_TEST_SCORE *	ACT NGT Test Score
NGT_TEST_YEAR	ACT NGT Test Year
NRH_NATIONAL_NORM	ACT Rhetorical Skills National Norm
NRH_TADM_CODE	ACT NRH Test Admin Code
NRH_TEST_MON	ACT NRH Test Month
NRH_TEST_SCORE *	ACT NRH Test Score
NRH_TEST_YEAR	ACT NRH Test Year
NSS_NATIONAL_NORM	ACT Soc Stud/Sci National Norm
NSS_TADM_CODE	ACT NSS Test Admin Code
NSS_TEST_MON	ACT NSS Test Month
NSS_TEST_SCORE *‘	ACT NSS Test Score
NSS_TEST_YEAR	ACT NSS Test Year
NUMBER_OF_DEPENDENTS	Number of Dependents
NUM_MCATS	Number of MCATS Placeholder
NUM_NATIONAL_NORM	ACT Usage/Mech National Norm
NUM_TADM_CODE	ACT NUM Test Admin Code
NUM_TEST_MON	ACT NUM Test Month
NUM_TEST_SCORE *	ACT NUM Test Score
NUM_TEST_YEAR	ACT NUM Test Year
NWR_NATIONAL_NORM	ACT Writing National Norm
NWR_TADM_CODE	ACT NWR Test Admin Code
NWR_TEST_MON	ACT NWR Test Month
NWR_TEST_SCORE *	ACT NWR Test Score
NWR_TEST_YEAR	ACT NWR Test Year
OLDSAT1_PERCENTILE	SAT II National Percentile
OLDSAT1_RCRV_IND	Old SAT RCRV
OLDSAT1_TADM_CODE	Old SAT Test Admin Code
OLDSAT1_TEST_CODE	Old SAT Test Code
OLDSAT1_TEST_MON	Old SAT Test Month
OLDSAT1_TEST_SCORE	Old SAT Test Score
OLDSAT1_TEST_YEAR	Old SAT Test Year
OLDSAT2_PERCENTILE	SAT II National Percentile
OLDSAT2_TEST_CODE	Old SAT Test Code
OLDSAT2_TEST_SCORE	Old SAT Test Score
OLDSAT3_PERCENTILE	SAT II National Percentile
OLDSAT3_TEST_CODE	Old SAT Test Code
OLDSAT3_TEST_SCORE	Old SAT Test Score
PBUG_AO_GPA	Post Bac UG All Other GPA
PBUG_AO_HOURS	Post Bac UG All Other Hours
PBUG_BCPM_GPA	Post Bac UG BioChemPhyMath GPA

PBUG_BCPM_HOURS	Post Bac UG BioChemPhyMath Hrs
PBUG_TOTAL_GPA	Post Bac UG Total GPA
PCOL_CITY	Prior College City
PCOL_DEGR_CODE	Prior College Degree Code
PCOL_DEGR_CODE2	Previous College Degree 2
PCOL_DEGR_CODE3	Previous College Degree 3
PCOL_DEGR_CODE4	Previous College Degree 4
PCOL_DEGR_CODE5	Previous College Degree 5
PCOL_GPA	Previous College GPA
PCOL_GRAD_CENT	Previous College Grad Century
PCOL_GRAD_DATE	Previous College Grad Date
PCOL_GRAD_DATE2	Previous College Grad Date 2
PCOL_GRAD_DATE3	Previous College Grad Date 3
PCOL_GRAD_DATE4	Previous College Grad Date 4
PCOL_GRAD_DATE5	Previous College Grad Date 5
PCOL_GRAD_DAY	Previous College Grad Day
PCOL_GRAD_MON	Previous College Grad Month
PCOL_GRAD_YEAR	Previous College Grad Year
PCOL_LONG_DESC	Previous College Long Placehld
PCOL_MAJR_CODE	Previous College Major 1
PCOL_MAJR_CODE2	Previous College Major 2
PCOL_MAJR_CODE3	Previous College Major 3
PCOL_MAJR_CODE4	Previous College Major 4
PCOL_MAJR_CODE5	Previous College Major 5
PCOL_MAJR_NAME	Prev Coll Majr Desc 1 Placehld
PCOL_MAJR_NAME2	Prev Coll Majr Desc 2 Placehld
PCOL_MAJR_NAME3	Prev Coll Majr Desc 3 Placehld
PCOL_MAJR_NAME4	Prev Coll Majr Desc 4 Placehld
PCOL_MAJR_NAME5	Prev Coll Majr Desc 5 Placehld
PCOL_NAME	Previous College Name Placehld
PCOL_NATION	Previous College Nation
PCOL_PROGRAM	Previous College Program
PCOL_SBGI_CODE	Previous College SBGI Code
PCOL_SHORT_DESC	Previous College Shrt Placehld
PCOL_STAT_CODE	Previous College State Code
PCOL_STREET1	Previous College Street Line 1
PCOL_YEARS_ATTEND	Previous College To From YYYY
PCOL_ZIP	Previous College ZIP Code
PF_FAIL_HOURS	Pass Fail - Failed Hours
PF_PASS_HOURS	Pass Fail - Passed Hours

PHONE_AREA	Phone Area Code
PHONE_COMMENT	Phone Comment for GRE
PHONE_NUMBER	Phone Number
PHONE_PREF	Preferred Phone Number Ind
PIDM	PIDM
PLUS_MINUS_GRADES	Grades Use Plus or Minus
PREF_ADDRESS	Preferred Address Placeholder
PREF_CITY	Preferred City Placeholder
PREV_APP1	YYYY Previously Applied 1
PREV_APP2	YYYY Previously Applied 2
PREV_APP3	YYYY Previously Applied 3
PREV_APP4	YYYY Previously Applied 4
PREV_MATRIC	Previous Matriculation
RACE1	Race Code 1
RACE10	Race Code 10
RACE2	Race Code 2
RACE3	Race Code 3
RACE4	Race Code 4
RACE5	Race Code 5
RACE6	Race Code 6
RACE7	Race Code 7
RACE8	Race Code 8
RACE9	Race Code 9
REC_NUMBER	Record Number
RELG_CODE	Religion Code
S01_NATIONAL_PERCENTILE	SAT Verbal National Percentile
S01_RCRV_IND	SAT S01 RCRV Indicator
S01_STATE_PERCENTILE	SAT Verbal State Percentile
S01_TADM_CODE	SAT S01 Test Admin Code
S01_TEST_MON	SAT S01 Test Month
S01_TEST_SCORE	SAT S01 Test Score
S01_TEST_YEAR	SAT S01 Test Year
S02_NATIONAL_PERCENTILE	SAT Math National Percentile
S02_RCRV_IND	SAT S02 RCRV Indicator
S02_STATE_PERCENTILE	SAT Math State Percentile
S02_TADM_CODE	SAT S02 Test Admin Code
S02_TEST_MON	SAT S02 Test Month
S02_TEST_SCORE	SAT S02 Test Score
S02_TEST_YEAR	SAT S02 Test Year
S03_RCRV_IND	SAT S03 RCRV Indicator

S03_TADM_CODE	SAT S03 Test Admin Code
S03_TEST_MON	SAT S03 Test Month
S03_TEST_SCORE	SAT S03 Test Score
S03_TEST_YEAR	SAT S03 Test Year
S04_RCRV_IND	SAT S04 RCRV Indicator
S04_TADM_CODE	SAT S04 Test Admin Code
S04_TEST_MON	SAT S04 Test Month
S04_TEST_SCORE	SAT S04 Test Score
S04_TEST_YEAR	SAT S04 Test Year
S05_RCRV_IND	SAT S05 RCRV Indicator
S05_TADM_CODE	SAT S05 Test Admin Code
S05_TEST_MON	SAT S05 Test Month
S05_TEST_SCORE	SAT S05 Test Score
S05_TEST_YEAR	SAT S05 Test Year
S06_RCRV_IND	SAT S06 RCRV Indicator
S06_TADM_CODE	SAT S06 Test Admin Code
S06_TEST_MON	SAT S06 Test Month
S06_TEST_SCORE	SAT S06 Test Score
S06_TEST_YEAR	SAT S06 Test Year
S07_NATIONAL_PERCENTILE	SAT Writing Natl Percentile
S07_RCRV_IND	SAT S07 RCRV Indicator
S07_STATE_PERCENTILE	SAT Writing State Percentile
S07_TADM_CODE	SAT S07 Test Admin Code
S07_TEST_MON	SAT S07 Test Month
S07_TEST_SCORE	SAT S07 Test Score
S07_TEST_YEAR	SAT S07 Test Year
S08_RCRV_IND	SAT S08 RCRV Indicator
S08_TADM_CODE	SAT S08 Test Admin Code
S08_TEST_MON	SAT S08 Test Month
S08_TEST_SCORE	SAT S08 Test Score
S08_TEST_YEAR	SAT S08 Test Year
S09_RCRV_IND	SAT S09 RCRV Indicator
S09_TADM_CODE	SAT S09 Test Admin Code
S09_TEST_MON	SAT S09 Test Month
S09_TEST_SCORE	SAT S09 Test Score
S09_TEST_YEAR	SAT S09 Test Year
SAG_TADM_CODE	ACT SAG Test Admin Code
SAG_TEST_MON	ACT SAG Test Month
SAG_TEST_SCORE	ACT SAG Test Score
SAG_TEST_YEAR	ACT SAG Test Year

SAL_TADM_CODE	ACT SAL Test Admin Code
SAL_TEST_MON	ACT SAL Test Month
SAL_TEST_SCORE	ACT SAL Test Score
SAL_TEST_YEAR	ACT SAL Test Year
SAT_ESSAY_ID	SAT Essay ID
SCHOOL_NUM	School Number
SCHOOL_STATE	School State
SCHOOL_UNUSED	MCAT School Unused Placeholder
SEA_TADM_CODE	ACT SEA Test Admin Code
SEA_TEST_MON	ACT SEA Test Month
SEA_TEST_SCORE	ACT SEA Test Score
SEA_TEST_YEAR	ACT SEA Test Year
SGT_TADM_CODE	ACT SGT Test Admin Code
SGT_TEST_MON	ACT SGT Test Month
SGT_TEST_SCORE	ACT SGT Test Score
SGT_TEST_YEAR	ACT SGT Test Year
SO_AO_GPA	Sophomore All Other GPA
SO_AO_HOURS	Sophomore All Other Hours
SO_BCPM_GPA	Sophomore BioChemPhysMath GPA
SO_BCPM_HOURS	Sophomore BioChemPhysMath Hrs
SO_TOTAL_GPA	Sophomore Total GPA
SRH_TADM_CODE	ACT SRH Test Admin Code
SRH_TEST_MON	ACT SRH Test Month
SRH_TEST_SCORE	ACT SRH Test Score
SRH_TEST_YEAR	ACT SRH Test Year
SR_AO_GPA	Senior All Other GPA
SR_AO_HOURS	Senior All Other Hours
SR_BCPM_GPA	Senior BioChemPhysMath GPA
SR_BCPM_HOURS	Senior BioChemPhysMath Hours
SR_TOTAL_GPA	Senior Total GPA
SSN	Social Security Number
SSN_OLD	Old SSN Placeholder
SSS_TADM_CODE	ACT SSS Test Admin Code
SSS_TEST_MON	ACT SSS Test Month
SSS_TEST_SCORE	ACT SSS Test Score
SSS_TEST_YEAR	ACT SSS Test Year
STAT_CODE	State Code
STREET_LINE1	Street Address Line 1
STREET_LINE2	Street Address Line 2
STREET_LINE3	Street Address Line 3

STVTPFD Tape Field Names Validation Form

STUDENT_TYPE	Student Type
SUFFIX	Name Suffix
SUM_TADM_CODE	ACT SUM Test Admin Code
SUM_TEST_MON	ACT SUM Test Month
SUM_TEST_SCORE	ACT SUM Test Score
SUM_TEST_YEAR	ACT SUM Test Year
SWR_TEST_SCORE	ACT SWR Test Score
TELE_TYPE_CODE	Telephone Type Code
URM_IND	URM Indicator Placeholder
VERIFY_IND	Verified Indicator Placeholder
VISA	Visa Type
ZIP	ZIP Code

STVTRNS Transcript Name Source Code Validation Form

IDEN	General Person Identification
LEGAL	General Person Legal Name
DIPLOMA	Diploma Name

STVTSPT Test Score Percentile Type Validation Form

ANN	ACT National Norm Cumulative Percent
GRP	GRE Percentile
SMN	SAT Math National College-Bound Percentile
SMS	SAT Math State College-Bound Percentile
SVN	SAT Verbal National College-Bound Percentile
SVS	SAT Verbal State College-Bound Percentile
SWN	SAT Writing National College-Bound Percentile
SWS	SAT Writing State College-Bound Percentile
S2N	SAT II National College-Bound Percentile

STVVTAB Web Display Tables Validation Form

GTVINSM	Instructional Method Validation
STVATTR	Attribute Validation
STVBLDG	Building Code Validation
STVCAMP	Campus Code Validation
STVCOLL	College Code Validation
STVDEPT	Department Code Validation
STVDIVS	Division Code Validation
STVLEVL	Level Code Validation

STVVTAB Web Display Tables Validation Form

STVPTRM	Part of Term Code Validation
STVSCHD	Schedule Type Code Validation
STVSUBJ	Subject Code Validation
STVSESS	Session Code Validation
STVTESC	Test Code Validation

STVWACK Web Prospect Acknowledgement Letter Codes Validation Form

ADDR1_ATYP_CODE	Address 1 Address Code
ADDR1_CITY	Address 1 City
ADDR1_FROM_DATE	Address 1 From Date
ADDR1_HNUM	Address 1 House Number
ADDR1_NATN_CODE	Address 1 Nation Code
ADDR1_STAT_CODE	Address 1 State Code
ADDR1_STR1	Address 1 Street 1
ADDR1_STR2	Address 1 Street 2
ADDR1_STR3	Address 1 Street 3
ADDR1_STR4	Address 1 Street 4
ADDR1_TO_DATE	Address 1 To Date
ADDR1_ZIP	Address 1 ZIP
ADDR2_ATYP_CODE	Address 2 Address Code
ADDR2_CITY	Address 2 City
ADDR2_FROM_DATE	Address 2 From Date
ADDR2_HNUM	Address 2 House Number
ADDR2_NATN_CODE	Address 2 Nation Code
ADDR2_STAT_CODE	Address 2 State Code
ADDR2_STR1	Address 2 Street 1
ADDR2_STR2	Address 2 Street 2
ADDR2_STR3	Address 2 Street 3
ADDR2_STR4	Address 2 Street 4
ADDR2_TO_DATE	Address 2 To Date
ADDR2_ZIP	Address 2 ZIP
BIRTHDATE	Birth Date
CITZ_CODE	Citizenship
EMAIL_ADDRESS	Email Address
EMAL_CODE	Email Address Code
ETHN_CODE	Ethnicity
FIRST_NAME	First Name
FOREIGN_SSN	Foreign SSN
GENDER	Gender
HS_CITY	High School City

HS_CLASS_RANK	High School Class Rank
HS_CLASS_SIZE	High School Class Size
HS_GPA	High School GPA
HS_GRAD_DATE	High School Grad Date
HS_HNUM	High School House Number
HS_NAME	High School Name
HS_NATN_CODE	High School Nation
HS_SBGI_CODE	High School SBGI Code
HS_STAT_CODE	High School State
HS_STR1	High School Street 1
HS_STR2	High School Street 2
HS_STR3	High School Street 3
HS_STR4	High School Street 4
HS_ZIP	High School ZIP Code
INTS_CODE_1	Interest 1
INTS_CODE_2	Interest 2
INTS_CODE_3	Interest 3
INTS_CODE_4	Interest 4
INTS_CODE_5	Interest 5
LAST_NAME	Last Name
LAST_NAME_PREFIX	Surname Prefix
LEND_CODE_1	How I Learned About
LEND_CODE_2	How I Learned About
LEND_CODE_3	How I Learned About
MAJR_CODE	Major
MATL_CODE_1	Requested Material
MATL_CODE_2	Requested Material
MATL_CODE_3	Requested Material
MIDDLE_NAME	Middle Name
NAME_PREFIX	Name Prefix
NAME_SUFFIX	Name Suffix
NATN_CODE_LEGAL	Country of Citizenship
NICKNAME	Nickname
NTYP_CODE	Name Type Code
PCOL_CITY	Prior College City
PCOL_GPA	Prior College GPA
PCOL_GRAD_DATE	Prior College Grad Date
PCOL_HNUM	Prior College House Number
PCOL_NAME	Prior College Name
PCOL_NATN_CODE	Prior College Nation

PCOL_SBGI_CODE	Prior College SBGI Code
PCOL_STAT_CODE	Prior College State
PCOL_STR1	Prior College Street 1
PCOL_STR2	Prior College Street 2
PCOL_STR3	Prior College Street 3
PCOL_STR4	Prior College Street 4
PCOL_ZIP	Prior College ZIP Code
SSNTINTFN	SSN/TIN/TFN
STYP_CODE	Student Type
TELE1_ATYP_CODE	Telephone 1 Address Code
TELE1_CTRY_PHONE	Telephone 1 Country Code
TELE1_INTL_ACCESS	Telephone 1 Inter Access
TELE1_PHONE_AREA	Telephone 1 Phone Area
TELE1_PHONE_EXT	Telephone 1 Phone Extension
TELE1_PHONE_NUMBER	Telephone 1 Phone Number
TELE2_ATYP_CODE	Telephone 2 Address Code
TELE2_CTRY_PHONE	Telephone 2 Country Code
TELE2_INTL_ACCESS	Telephone 2 Inter Access
TELE2_PHONE_AREA	Telephone 2 Phone Area
TELE2_PHONE_EXT	Telephone 2 Phone Extension
TELE2_PHONE_NUMBER	Telephone 2 Phone Number
TELE3_CTRY_PHONE	Telephone 3 Country Code
TELE3_INTL_ACCESS	Telephone 3 Inter Access
TELE3_PHONE_AREA	Telephone 3 Phone Area
TELE3_PHONE_EXT	Telephone 3 Phone Extension
TELE3_PHONE_NUMBER	Telephone 3 Phone Number
TELE4_CTRY_PHONE	Telephone 4 Country Code
TELE4_INTL_ACCESS	Telephone 4 Inter Access
TELE4_PHONE_AREA	Telephone 4 Phone Area
TELE4_PHONE_EXT	Telephone 4 Phone Extension
TELE4_PHONE_NUMBER	Telephone 4 Phone Number
TELE5_CTRY_PHONE	Telephone 5 Country Code
TELE5_INTL_ACCESS	Telephone 5 Inter Access
TELE5_PHONE_AREA	Telephone 5 Phone Area
TELE5_PHONE_EXT	Telephone 5 Phone Extension
TELE5_PHONE_NUMBER	Telephone 5 Phone Number
TERM_CODE	Term Code
TEST1_DATE	Test 1 Date
TEST1_SCORE	Test 1 Score
TEST1_TESC_CODE	Test 1 TESC Code

STVWACK **Web Prospect Acknowledgement Letter Codes Validation Form**

TEST2_DATE	Test 2 Date
TEST2_SCORE	Test 2 Score
TEST2_TESC_CODE	Test 2 TESC Code
TEST3_DATE	Test 3 Date
TEST3_SCORE	Test 3 Score
TEST3_TESC_CODE	Test 3 TESC Code
TEST4_DATE	Test 4 Date
TEST4_SCORE	Test 4 Score
TEST4_TESC_CODE	Test 4 TESC Code
TEST5_DATE	Test 5 Date
TEST5_SCORE	Test 5 Score
TEST5_TESC_CODE	Test 5 TESC Code
VTYP_CODE	Visa

STVWAPP **Application Type Code Validation Form**

00	Default Example - All Sections
W1	Undergraduate Freshman
W2	Undergraduate Transfer
W3	Internatnl Undergrad Freshman
W4	Internatnl Undergrad Transfer
W5	Graduate Studies
W6	International Graduate Studies
W7	Continuing Ed, Non-Degree

STVWLTT **Web Application Customized Letter Type Validation Form**

Letter Type Code	Module	Description
APPLHOLD	E	Application Holds Exist Letter
DECNERR	E	Error on Decision Record Letter
DEFAULT	E	Default Letter If Not Specified
MATCHERR	E	Matching Processing Returns Error Letter
NOSTUREC	S	Cannot Insert Student Record Letter
PUSHERR	E	Push Processing Returns Error Letter
QUIKADMT	S	Standard Quick Start Signature Letter
STANDARD	E	Standard Web Application Signature Letter
SUSPENSE	E	Suspended Match Status Signature Letter
VERERR	E	Verify Processing Returns Error Letter

STVWPIC **Web Prospect Information Selection Validation Form**

ADDRESS1	Primary Address
ADDRESS2	Secondary Address
BIRTHDATE	Prospect Birthdate
CITIZENSHP	Prospect Citizenship
EMAIL	E-Mail Address
ENTRYTERM	Prospect Entry Term
ETHNICITY	Prospect Ethnicity/Race
GENDER	Prospect Gender
HIGHSCHOOL	Prospect High School
HOWILEARND	How Prospect Learned About Us
INTERESTS	Prospect Interests
INTERNATNL	Prospect International Info
MAJOR	Prospect Major
MATERIAL	Requested Material
NAME	Prospect Name
NTYPE	Prospect Name Type
PRIORCOLL	Prospect Prior College
SSNTINTFN	Prospect SSN/TIN/TFN
STUDENTTYP	Prospect Student Type
TELE3	Additional Telephone Numbers
TESTSCORES	Prospect Test Scores
VISA	Visa Information

STVWSCF **Admissions Web Page Element Validation Form**

Element Code	Description	Web Section
ACTIVITY	Activity	ACTIVITIES
ADDR2_CITY	City	ADDR2
ADDR2_COUNTY	County	ADDR2
ADDR2_HOUSE_NUMBER	House Number	ADDR2
ADDR2_INTL_ACCESS	International Access	ADDR2
ADDR2_NATION	Country	ADDR2
ADDR2_PHONE_CTRY_CODE	Phone Country Code	ADDR2
ADDR2_PHONE_NUMBER	Phone Number	ADDR2
ADDR2_STATE	State/Province	ADDR2
ADDR2_STREET1	Street Line 1	ADDR2
ADDR2_STREET2	Street Line 2	ADDR2
ADDR2_STREET3	Street Line 3	ADDR2
ADDR2_STREET4	Street Line 4	ADDR2

STVWSCF Admissions Web Page Element Validation Form

Element Code	Description	Web Section
ADDR2_ZIP	ZIP Code	ADDR2
ATYP	Address Type	ADDR1
BIRTHDATE	Birth Date	PERSONAL
CELL_EXTENSION	Cellular Phone Ext	PERSONAL
CELL_INTL_ACCESS	Cellular Phone Int Access Code	PERSONAL
CELL_PHONE	Cellular Phone Number	PERSONAL
CELL_PHONE_CTRY_CODE	Cellular Phone Country Code	PERSONAL
CITIZEN	Citizenship	PERSONAL
CITY	City	ADDR1
CONCENTRATION	Concentration	PLAN
CONFIDENT	Confidentiality	PERSONAL
COUNTY	County	ADDR1
EMAIL	Email	PERSONAL
ETHNIC	Ethnicity	PERSONAL
ETHNIC_CATEGORY	Ethnicity	PERSONAL
FIRST_NAME	First Name	NAME
GENDER	Gender	PERSONAL
HOUSE_NUMBER	House Number	ADDR1
HSCH_CITY	High School City	HIGHSCH
HSCH_CLASS_SIZE	High School Class Size	HIGHSCH
HSCH_CODE	High School Code	HIGHSCH
HSCH_COUNTY	High School County	HIGHSCH
HSCH_GPA	GPA	HIGHSCH
HSCH_GRAD_DATE	Graduation Date	HIGHSCH
HSCH_HOME_SCHOOL	Home School	HIGHSCH
HSCH_HOUSE_NUMBER	House Number	HIGHSCH
HSCH_NAME	High School Name	HIGHSCH
HSCH_NATION	High School Nation	HIGHSCH
HSCH_RANK	High School Rank	HIGHSCH
HSCH_STATE	High School State	HIGHSCH
HSCH_STREET1	High School Street1	HIGHSCH
HSCH_STREET2	High School Street2	HIGHSCH
HSCH_STREET3	High School Street3	HIGHSCH
HSCH_STREET4	High School Street4	HIGHSCH
HSCH_ZIP	High School ZIP Code	HIGHSCH
INTL_ACCESS	International Access	ADDR1
LAST_NAME	Last Name	NAME

STVWSCF Admissions Web Page Element Validation Form

Element Code	Description	Web Section
LAST_NAME_PREFIX	Last Name Prefix	NAME
LEGACY	Legacy	PERSONAL
MAJOR	Major	PLAN
MARITAL	Marital Status	PERSONAL
MATERIAL	Requested Material	MATERIALS
MEDICAL	Medical Information	PERSONAL
MID_NAME	Middle Name	NAME
MINOR	Minor	PLAN
NATION	Nation	ADDR1
NICKNAME	Nickname	NAME
OTHER_ACTIVITY	Other Activity	ACTIVITIES
PAR_CITY	City	PARENTS
PAR_CODE	Relationship	PARENTS
PAR_COUNTY	County	PARENTS
PAR_DECEASED	Deceased	PARENTS
PAR_FIRST_NAME	First Name	PARENTS
PAR_HOUSE_NUMBER	House Number	PARENTS
PAR_LAST_NAME	Last Name	PARENTS
PAR_MIDDLE_NAME	Middle Name	PARENTS
PAR_NATION	Nation	PARENTS
PAR_PHONE_CTRY_CODE	Phone Country Code	PARENTS
PAR_PREFIX	Prefix	PARENTS
PAR_STATE	State/Province	PARENTS
PAR_STREET1	Street Line 1	PARENTS
PAR_STREET2	Street Line 2	PARENTS
PAR_STREET3	Street Line 3	PARENTS
PAR_STREET4	Street Line 4	PARENTS
PAR_SUFFIX	Suffix	PARENTS
PAR_SURNAME_PREFIX	Last Name Prefix	PARENTS
PAR_ZIP	ZIP Code	PARENTS
PARENT_EMPLOYER	Employer	PARENTS
PARENT_INTL_ACCESS	International Access	PARENTS
PARENT_PHONE_NUMBER	Phone Number	PARENTS
PCOL_ATTEND_FROM	College Attend From Date	PRVCOLLEGE
PCOL_ATTEND_TO	College Attend To Date	PRVCOLLEGE
PCOL_CITY	City	PRVCOLLEGE
PCOL_CLASS_SIZE	Class Size	PRVCOLLEGE
PCOL_CODE	College School Code	PRVCOLLEGE
PCOL_COUNTY	College County	PRVCOLLEGE

STVWSCF Admissions Web Page Element Validation Form

Element Code	Description	Web Section
PCOL_DEGREE	College Degree	PRVCOLLEGE
PCOL_DEGREE_DATE	College Degree Date	PRVCOLLEGE
PCOL_GPA	GPA	PRVCOLLEGE
PCOL_HOUSE_NUMBER	House Number	PRVCOLLEGE
PCOL_MAJOR	College Major	PRVCOLLEGE
PCOL_MINOR	College Minor	PRVCOLLEGE
PCOL_NAME	College Name	PRVCOLLEGE
PCOL_NATION	College Nation	PRVCOLLEGE
PCOL_RANK	Rank	PRVCOLLEGE
PCOL_STATE	College State	PRVCOLLEGE
PCOL_STREET1	Street1	PRVCOLLEGE
PCOL_STREET2	Street2	PRVCOLLEGE
PCOL_STREET3	Street3	PRVCOLLEGE
PCOL_STREET4	Street4	PRVCOLLEGE
PCOL_ZIP	ZIP Code	PRVCOLLEGE
PHONE_CTRY_CODE	Phone Country Code	ADDR1
PHONE_NUMBER	Phone Number	ADDR1
PREFIX	Prefix	NAME
PREV_APPL	Previously Applied?	NAME
PREV_ATTEND	Previously Attended?	NAME
PREV_LAST	Previous Last Name	NAME
QUESTION	Question	*

*You can select the Web section value that best suits the *QUESTION* element code for your site.

RACE	Race	PERSONAL
RELIGION	Religion	PERSONAL
RESID	Resident	PERSONAL
SSN	SSN	PERSONAL
STATE	State	ADDR1
STREET1	Street Line 1	ADDR1
STREET2	Street Line 2	ADDR1
STREET3	Street Line 3	ADDR1
STREET4	Street Line 4	ADDR1
SUFFIX	Suffix	NAME
TEST_INFO	Test	TESTS
VET_CATEGORY	Veteran Category	PERSONAL
VET_FILE_NO	Veteran ID	PERSONAL
VISA_BIRTH_NATION	Birth Country	INTERNATL
VISA_CITZN_NATION	Citizenship Country	INTERNATL

STVWSCF Admissions Web Page Element Validation Form

Element Code	Description	Web Section
VISA_EXPIRE_DATE	Visa Expiration Date	INTERNATL
VISA_ISSUE_DATE	Visa Issue Date	INTERNATL
VISA_LANGUAGE	Native Language	INTERNATL
VISA_NUMBER	Visa Number	INTERNATL
VISA_TYPE	Visa	INTERNATL
ZIP	ZIP Code	ADDR1

STVWSCT Web Prospect Information Selection Validation Form

ACTIVITIES	Activities
ADDITIONAL	Additional Information
ADDR1	First Address and Phone
ADDR2	Second Address and Phone
ESSAY	Essay Questions
HIGHSCH	High School
INTERNATL	International Information
MATERIALS	Requested Materials
NAME	Name
PARENTS	Parental Information
PERSONAL	Personal Information
PLAN	Planned Course of Study
PRVCOLLEGE	Previous College
TESTS	Test Scores

Note

You must also enter the appropriate Web procedure name in the **Procedure** field for each section code.

STVXLBL EDI Verification Label Validation Form

AGEVERFY	Age Verification Codes
ASESSLVL	XML Import StudentLevelCode
CNTYRURL	County Rural Indicators
CRDBASIS	XML CourseCreditBasis to EDI
CRDUNIT	XML CourseCreditUnits to EDI
CRDLEVEL	XML CourseCreditLevel
CRSLEVEL	XML CourseLevel to EDI
CRSOVERR	XML CourseOverideSchool
CRSRPEAT	XML CourseRepeatCode to EDI
DEGRLEVL	Degree Level Codes

STVXLBL **EDI Verification Label Validation Form**

DFMTCODE	Date Format Codes
EIDNCODE	Entity Identifier Codes
FSTYIDQL	Field of Study Qualifier Codes
FSTYLEVL	Field of Study Level Codes
GENDER	Gender Codes
GRCRLEVL	Grade or Course Level Codes
GTVLFST	Field of Study Types
HEADXPRP	XML DocumentType to XPRP
HEADXRSN	XML TransmissionType to XRSN
HONRLEVL	Honor Level Codes
HSCHRSON	High School Graduation Codes
IMMZRTYP	Immunization Report Type
IMMZSTAT	Immunization Status Codes
IMMZTYPE	Immunization Type Codes
ITCCLEVL	Individual Level Codes
LANGPROF	Language Proficiency Codes
LANGUSE	Language Use Codes
NTYPCODE	Name Type Codes
QSTNCODE	EDI Question Codes
RESDCRIT	Residency Criteria Codes
RFNOQLFR	Reference Qualifier Code
SBGIQLFR	Educational Inst Qualifier
STVASTDD	XML ASTD Delinquency codes
STVASTDH	XML ASTD Honor codes
STVATTR	Course Attributes
STVATYP	Address Type Codes
STVATYPR	Relatives Address Type Codes
STVCITZ	Citizenship Type Codes
STVCLAS	Class Codes
STVCNTY	County Codes
STVDEGC	Degree Level-Degree Code
STVEGOL	College Program Type Codes
STVESEL	Eligibility Factor Codes
STVETHN	Ethnic Type Codes
STVHONR	Honor codes
STVINTS	Award and Activity Codes
STVLANGN	Language Name Codes
STVLANGW	Written Language Codes
STVLEVL	Course and Student Level codes
STVLGCY	Legacy Codes

STVXLBL **EDI Verification Label Validation Form**

STVMAJR	EDI Major Codes
STVMATL	Requested Materials
STVMEDI	Medical Condition Codes
STVMRTL	Marital Status Codes
STVNATN	Nation Codes
STVRELG	Religion Codes
STVRELT	Relationship Codes
STVSBGIC	EDI College Codes
STVSBGIH	EDI High School Codes
STVSTAT	State Codes
STVTEAC	MCAT Non-Standard Indicator
STVTEFR	XML Test form codes
STVTELE	Telephone Qualifier Codes
STVTESC	Sub-Test Codes
STVTRMT	Term Type codes
STVTRMTC	Term Type for Credit Units
STVVETC	Veteran Type Code
STVVTYP	VISA Type Codes
STVWAIV	Appl Waiver Code
SUMACTYP	XML AcademicSummaryLevel=>CTYP
SUMASLVL	XML AcademicSummaryType =>SLVL
TESTCODE	EDI Test Qualifier Codes

Index

A

B

C

D

H

I

O

R

S

U