SICAS Center

SIRIS IR Submissions

CDS
SDS
TSDS
FADS
SRDS
DADS

Student 8.6S1

August 2014

Prepared For: SICAS Members
Prepared By: SICAS Center

SUNY Oneonta
Oneonta, NY   13820

Original Release July 2008
Revised and Reissued, 6 January 2009
Revised and Reissued March 2010
Revised and Reissued January 2011
Revised and Reissued May 2011
Revised and Reissued August 2011
Revised and Reissued November 2012
Revised and Reissued May 2014
Revised and Reissued August 2014

TABLE OF CONTENTS

INTRODUCTION

SUNY System Administration's objective is to simplify and streamline the reporting of campus information to the Office of Institutional Research (IR). The implementation of web-enabled methods to govern the submission and correction of data facilitates the reporting of campus information. The development of a data warehouse environment allows System Administration to store data from all functional areas. The standardization of data element definitions and coding schemes cross-referenced and conformed across all University-wide systems improves the querying, viewing, and downloading of data. Data are stored in an online media that makes it readily available for reference, reporting, and analysis by campuses and the Office of Institutional Research.

The information required for IR submissions is collected in Banner and transmitted to System Administration using a Proxy Client (see Appendix C for information on the proxy client). The SICAS IR Submission applications support web-based reporting of campus information to the SUNY Institutional Research Information System (SIRIS). These applications collect the data from Banner and use Web Services to transmit it in an XML document formatted specifically for SUNY's Data Warehouse. The XML document is placed in the Data Warehouse staging area. Edit checks are performed and an XML document is returned with the data that was received and message entries describing any errors encountered. Errors are corrected in Banner and the altered data are re-transmitted to System Administration.

Data transmitted to the Data Warehouse do not actually update the warehouse. Data are held in the staging area in a 'batch', which can be reviewed and locked (finalized) by the campus using a web-based application on System Administration's website. Once locked, a batch cannot be edited and can only be unlocked by IR staff at System Administration. IR staff approves the locked batch and incorporates the data into the Data Warehouse.

SUNY's Office of Institutional Research developed SIRIS Submission Data Dictionaries to standardize element names and definitions. The SUNY SIRIS Submission Data Dictionaries that define the data elements for each submission are available on SUNY's Confluence website at http://confluence.itec.suny.edu.

This manual provides information for the SIRIS Course Data Submissions (CDS), Student Data Submissions (SDS), Term/Section Data Submissions (TSDS), Financial Aid Data Submission (FADS), Student Revenue Data Submission (SRDS) and Degree Awarded Data Submission (DADS). In the future, it will be expanded to include other Student applications that require data submissions to the Office of Institutional Research.

OVERVIEW

The data extracted from Banner for SUNY Institutional Research (IR) submissions for student applications are viewed and queried on the SICAS Data Transfer Interface Control Form SOAYDTI. The Financial Aid Data Submission (FADS) is viewed and queried on ROAYDTI. The Student Revenue Data Submission (SRDS) is viewed and queried on TOAYDTI. GORYDTI is the support form underlying the SOAYDTI, ROAYDTI and TOAYDTI Data Transfer forms; GORYDTI cannot be accessed directly. Users access SOAYDTI/ROAYDTI/TOAYDTI as the main form for all SIRIS submissions. This is a tabbed form that provides a definition for each data element collected and the Select Statement executed to retrieve the data from Banner. The form shows Validation Codes (System Administration values) and Code Crosswalks. Data Review, Web Service information, and Processing Errors are also available on this form.

Use the SICAS Data Transfer Interface Control Form SOAYDTI to access the Student SIRIS IR Submission interfaces. These include: Course Data Submission (CDS), Student Data Submission (SDS), Term/Section Data Submission (TSDS), Degree Awarded Data Submission (DADS) and the SIRIS IR Submission Support interface (SIRS). Use ROAYDTI to access the Financial Aid IR Submission interface (FADS). Use TOAYDTI to access the Student Revenue Data Submission (SRDS) interface.

The SIRIS IR Submission Support interface (SIRS) contains information common to the Data Submission Interface Types. The Data Elements that appear on the SIRS Column Definitions tab only need to be cross walked once using the SIRS Code Crosswalk tab. Common data elements cross walked on the SIRS interface are not available on the individual data submission interfaces (i.e. SDS or TSDS); therefore the SIRS interface must be reviewed to ensure that these data elements are cross walked. Refer to the SIRS SIRIS IR Submission Support section under Set Up for more information.

Since Course Data Submission (CDS), Student Data Submission (SDS), Term/Section Data Submission (TSDS), Financial Aid Data Submission (FADS), Student Revenue Data Submission (SRDS) and Degree Awarded Data Submission (DADS) are transmitted using Web Services only (no flat files are created), the Proxy Client and certificate must be installed and working before these applications can be used. Refer to the Web Services/Proxy Client/Certificate section under the SOAYDTI SICAS Data Transfer Interface Control Form for more information.

Web Services transmits the data collected from Banner to the staging area of the Data Warehouse at System Administration. Any errors encountered are returned immediately and can be viewed on the Processing Errors Tab of the SOAYDTI/ROAYDTI/TOAYDTI forms.

To maintain synchronization, errors must be corrected in Banner and not with the file correction tools on the System Administration website. Once all errors are corrected and re-transmitted to System Administration, 'lock' the file using the System Administration website application (refer to the System Administration website for more information on locking a file). The file cannot be locked if the Threshold Errors are too high. Threshold Errors cannot be viewed in Banner; they must be reviewed on the System Administration website. Only IR staff at System Administration can unlock a file. After the file is 'locked' and 'approved' by IR staff, the data are moved from the staging area to the Data Warehouse. Once a data submission is 'approved', it is final. System Administration does not have a way to 'unapprove' a submission. However, the Course Data Submission (CDS) can be Unposted (UU or UR mode) locally, corrections can be made, re-transmitted, and re-locked for Approval on the System Administration Website. CDS can only be Unposted before the Term/Section Data Submission (TSDS) is Approved; once TSDS is Approved, all submissions are final.

All Set Up requirements must be completed before running any process. Refer to the manual for additional information on any of the items below.

IR Submission Process Flow Overview

- Run CDS for the term. This must be locked, approved and posted before TSDS can be run.
- Run SDS for the Early Student Submission (ESS). Lock, approve and post.
- Run SDS for the End-of-term (EOT) submission. This should be "mostly clean" to limit any unusual errors in the TSDS run.
- Run TSDS until accurate.
- SDS EOT and TSDS are locked when accurate. They are then approved, and posted.
- Once TSDS is Approved, CDS can no longer be Unposted and all submissions are final.
- FADS is run during the fall for the previous Aid Year. Correct any errors, lock when accurate have System Administration approve, then post the submission (PO mode).
- SRDS has two submission periods – one encompassing charges from January 1, XXXX through June 30, XXXX and the second encompassing charges from July 1, XXXX through December 31, XXXX.
- DADS has two submission periods for the academic year – one period encompasses degrees awarded from October 1, 20XX through June 30, 20YY. The second submission period encompasses degrees awarded from July 1, 20YY through September 30, 20YY.

The campus is responsible for running the reports, transmitting the data, cleaning up the errors while reviewing the data and then locking the file. System Administration will review the data and approve the file. The campus must then post the file. The final step for each transmission is posting after SA approves the file. Posting is done by running the process in PO mode. Terms MUST be processed in sequential order for each submission. For example, if you POST your winter EOT, you will be prevented from submitting an EOT file for the previous fall semester. This is true for all of the submissions except FADS and SRDS which are not term related.

SIRS Overview

The SIRIS IR Submission Support interface (SIRS) contains information common to the Data Submission interfaces. Data elements that appear on the SIRS interface Column Definitions tab must be cross walked. Use the SIRS Code Crosswalk tab to complete the crosswalks—data elements cross walked on the SIRS interface only need to be cross walked once. Data elements on the SIRS interface are not available on the individual data submission interfaces (i.e. SDS or TSDS); therefore, the SIRS Code Crosswalk tab must be reviewed to ensure that these data elements are cross walked. Refer to the SIRS SIRIS IR Submission Support section under Set Up for more information.

CDS Overview

The SICAS Course Data Submission (CDS) application extracts course data from the Basic Course Information Form SCACRSE using data from its baseline table, related Banner baseline tables, and the SICAS Course Shadow Table SCBYCRS. The SICAS Course Data Submission Report SCRYCDS collects the CDS extract data, optionally transmits the data to the warehouse staging area, and produces a listing of data and transmission status. Any errors returned when SCRYCDS is run in Transmit Mode are inserted into the SICAS Data Selection and Conversion Error Table SFRYERR and can be viewed on the Processing Errors tab of the SICAS Data Transfer Interface Control Form SOAYDTI. CDS must be configured before use. The Set Up section provides the information to configure the CDS application.

Before running the SICAS Course Data Submission Process Report SCRYCDS for the first time, review the course data for accuracy and completeness. After the data is verified and the application set up is complete, run the SCRYCDS process in Audit Mode. Review the results to determine that the Select Code is working properly. To view the Select Code, navigate to the Select Code tab, which is located on the lower portion of the Column Definitions tab on the SICAS Data Transfer Interface Control Form SOAYDTI.

The Course Data Submission (CDS) must be locked, approved, and posted before running the SICAS Term/Section Data Submission Report SSRYTSD. TSDS depends on the CDS being posted.

Refer to the manual for additional information on any of the items below.

CDS Set Up Overview

1. Review Column Definitions on the Column Definitions tab of the SOAYDTI form

2. Enter the SUNY Campus ID on the Validation Codes tab for the SIRS interface on the SOAYDTI form – See appendix A for the correct SUNY Campus ID code.

3. Crosswalk the Classification Codes for Teacher Certification courses on the Code Crosswalk tab of the SOAYDTI form

4. Enter Primary and Secondary Contact Information on the Contact Information tab of the SOAYDTI form - Primary Contact information is required for CDS

5. Review the General Ed Approved course indicator and enter information as necessary on the Basic Course Information Form SCACRSE

6. Enter the required codes on the Crosswalk Validation Form GTVSDAX

Refer to the manual for additional information on any of the items below.

CDS Process Flow Overview

CDS must be run each term regardless of whether or not you have made any changes to your course catalog.

1. Run the SICAS Course Data Submission Report SCRYCDS in Audit Mode
    a. Use the Audit Replace Mode (AR) the first term. This mode builds/replaces all course data
    b. Use the Audit Update Mode (AU) after the first term. This mode only builds changes.
    c. Occasionally, it may be desirable to build a full replacement using the Audit Replace Mode

        •   The Audit Replace Mode collects all course data with an effective term less than or equal to the term being processed

2. Review and confirm the resulting Course Data using the Data Review tab on the SOAYDTI form. Make corrections as needed

3. Run SCRYCDS in Transmit Mode to send data to the SUNY Data Warehouse staging area. For the initial transmission of data, use the same action (Replace or Update) that was used in Step 1

4. Review Errors on the Processing Errors tab of the SOAYDTI form - make corrections as needed

        •   To maintain synchronization, do not correct errors with the file correction tools on the System Administration website. All errors must be corrected in Banner.

5. Use Transmit Update Mode (TU) to send corrected data - repeat until all errors are resolved.
    *   Using Update Mode speeds the processing since only corrections are transmitted and successful courses are not resent. However, please be aware that new records will also be sent.

6. Use the SUNY System Administration Web page to review and lock the batch for approval

7. If necessary, to resolve any problems identified by SUNY IR staff - have the IR staff unlock your batch and then re-transmit and re-lock the batch

8. After SUNY IR staff Approves the batch, Run SCRYCDS in Post Batch Mode to update the Data Posted flag on the SICAS Course Data Submission extract tables

CDS Transmission Status:

Ready-	The course has been collected in the CDS process and is ready to transmit.
Current-	The course has been collected in the CDS process and will not be sent because there have been no changes. This is done so that a campus can extract a full version of their catalog. The posted status will remain No until the entire file is posted.
Non-CDS	This course was marked as "Do Not Send" on SCACRSE
Warning-	A warning has been received from System Administration during the transmission.
Error-	A fatal error has been received from System Administration during the transmission.

SDS Overview

The SICAS Student Data Submission (SDS) application extracts student data from Banner. The enrollment information, demographic features, and other characteristics collected are used to develop a large number of annual descriptive statistical summaries used by the Office of Institutional Research for planning, academic and fiscal administration, budget process, legal obligations, and reporting requirements imposed by State and Federal agencies.

The SICAS Student Data Submission Report SGRYSDS collects the SDS extract data, optionally transmits the data to the data warehouse staging area, and produces a listing of data and transmission status. Any errors returned when SGRYSDS is run in Transmit Mode are inserted into the SICAS Data Selection and Conversion Error Table SFRYERR and can be viewed on the Processing Errors tab of the SICAS Data Transfer Interface Control Form SOAYDTI. SDS must be configured before use. The Set Up section provides the information to configure the SDS application.

After set up is complete, run the SGRYSDS process in Audit Mode. Review the results to determine that the Select Code is working properly. To view the Select Code, navigate to the Select Code tab, which is located on the lower portion of the Column Definitions tab on the SICAS Data Transfer Interface Control Form SOAYDTI.

The SGRYSDS process must be run for an Early Student Submission (ESS) in the fall and spring Terms and an End of Term (EOT) submission for all terms with student registration.

The EOT Student Data Submission should be run prior to the TSDS submission. EOT should be "mostly clean" before running TSDS. This will limit the number of errors received during the TSDS run. EOT and TSDS should be run and cleaned up simultaneously. If students are

present in the TSDS run that were not previously sent in the EOT run, you will need to re-transmit EOT with those students in the file. Once TSDS is approved all submissions are final.

SDS Set Up Overview

1. Enter the required codes on the Crosswalk Validation Form GTVSDAX

2. Mark courses included in the Course Data Submission (CDS) with an Action status of 'Ready' (Ready to Submit) and enter a Course Level for those courses on the Basic Course Information Form SCACRSE

3. Mark the sections included in the Term/Section Data Submission (TSDS) with an Action status of 'Ready' (Ready to Submit) on the SUNY IR Submissions Detail block of the Schedule Form SSASECT. Only students enrolled in Sections that meet SIRIS criteria are included in the Student Data Submission (SDS)

4. Enter an individual's Ethnicity, Hispanic Origin, and Race information on the General Person Form SPAPERS

5. Cross walk Race Codes to IPEDS Codes on the Race Code Rules Form GORRACE

6. Enter APES Codes and APES Award Levels in the SUNY SDS/ASC Data block on the Majors and Departments tab of the Curriculum Rules Form SOACURR

7. Enter a Study Abroad Code for any student enrolled in a study abroad program on the Registration tab of the Student Course Registration Form SFAREGS

8. Enter the student's Educational Goal on the SUNY Student Educational Goal Form SOAYEDG or collect the information using the SICAS Student Educational Goal Survey

9. Enter an institution code in the Prior College field and indicate if it is the student's Home Institution or a Prior Institution on the Prior College and Degree tab of the Prior College Form SOAPCOL. This is for transfer students or students whose Home Institution is not this campus. SICAS has shipped the default value of 10 (unknown) for this element

10. Enter a Visa code and a Country of Citizenship on the International Information Form GOAINTL for all foreign students. This information is required in the submission and is also used to determine what information should be transmitted in the address fields.

11. Crosswalk the Banner Codes for the Hispanic Origin data element to System Administration values on the Code Crosswalk tab of the SOAYDTI form

12. Use the Code Crosswalk tab of the SOAYDTI Form to crosswalk all data elements that have a Crosswalk Source of GTVYVAL. Data elements are listed on the SOAYDTI Column Definitions tab. Verify that the data elements on the SOAYDTI form SIRS interface Column Definitions tab are cross walked

13. Enter Primary and Secondary Contact Information on the Contact Information tab of the SOAYDTI form. Primary Contact information is required for SDS

SDS Process Flow Overview

The SDS process is broken into two submissions, Early Student Submission (ESS) and End of Term (EOT) submission. The ESS is required for the Fall and Spring terms while the EOT is required for all terms with student registration.

1. Run the SICAS Student Data Submission Report SGRYSDS in Audit Mode
    a. Use the Audit Replace Mode (AR) the first time run for any given term
    b. Use the Audit Update Mode (AU) thereafter. This mode only builds changes.
        • For the Fall and Spring terms, the SGRYSDS process must be run in ESS mode before the EOT submission.

2.   Review and Confirm the resulting Student Data using the Data Review tab on the SOAYDTI Form. Make corrections as needed

3.   Run SGRYSDS in Transmit Mode to send data to the SUNY Data Warehouse staging area. For the initial transmission of data, use the same action (Replace or Update) that was used in Step 1

4.   Review Errors on the Processing Errors tab of the SOAYDTI form. Make corrections as needed
   - To maintain synchronization, errors must be corrected in Banner and not with the file correction tools on the System Administration website

5.   Use Transmit Update Mode (TU) to send corrected data. Repeat until all errors are resolved
   - Using Update Mode speeds the processing because only updates are sent; however, please be aware that new records will be sent. This includes new registrations.

6.   Use the SUNY System Administration Web page to review and lock the batch for approval

7.   If necessary, resolve any problems identified by SUNY IR staff - have the IR Staff un-lock your batch, then re-transmit and re-lock the batch

8.   After SUNY IR staff Approves the batch, Run SGRYSDS in Post Batch Mode (PO) to update the Data Posted flag on the SICAS Student Data Submission extract tables

TSDS Overview

The Term/Section Data Submission (TSDS) application extracts section data for the term being processed from the Schedule Form SSASECT, its baseline table, and other related Banner baseline tables. The SICAS Term Section Data Submission Report SSRYTSD collects the TSDS data and optionally transmits the data to the SUNY data warehouse staging area. Any errors returned when SSRYTSD is run in Transmit Mode are inserted into the SICAS Data Selection and Conversion Error Table SFRYERR and can be viewed on the Processing Errors tab of the SICAS Data Transfer Interface Control Form SOAYDTI. TSDS must be configured before use. The Set Up section provides the information to configure the TSDS application.

The Course Data Submission (CDS) must be Locked, Approved, and Posted before running the SICAS Term/Section Data Submission Report SSRYTSD. The SDS End of Term submission must be run and "mostly clean" before TSDS can be run. TSDS depends on CDS and SDS data.

If errors in the Course Catalog or Student Data are discovered while running TSDS, the CDS batch must be unposted (UU or UR mode). This new CDS submission must be locked, approved and posted in order to clear the TSDS errors. If there are TSDS errors that require retransmitting data from the SDS submission, re-run the process. It is best to leave your EOT unlocked until TSDS has been completed but, if your End of Term SDS has been locked, have the IR Staff at System Administration unlock it. The IR staff at System Administration should not approve your End of Term SDS submission until your TSDS submission is ready to be locked. Once TSDS is Approved, CDS can no longer be unposted - all submissions are final.

Refer to the manual for additional information on any of the items below.

TSDS Set Up Overview

1.   Review Column Definitions on the Column Definitions tab of the SOAYDTI form

2. Enter the SUNY Campus ID on the Validation Codes tab of the SOAYDTI form – see Appendix A for the correct SUNY Campus ID code.

3. Enter Primary and Secondary Contact Information on the Contact Information tab of the SOAYDTI form – Primary Contact information is required for TSDS

4. Perform all required crosswalks on the Code Crosswalk tab of the SOAYDTI form including all data elements that have a Crosswalk Source of GTVYVAL. Data elements are listed on the SOAYDTI Column Definitions tab. Verify that the data elements on the SOAYDTI form SIRS interface Column Definitions tab are cross walked

5. Enter Information as necessary on the Basic Course Information Form SCACRSE

6. Enter section information as necessary on the Schedule Form SSASECT

7. Enter the required codes on the Crosswalk Validation Form GTVSDAX

Refer to the manual for additional information on any of the items below.

TSDS Process Flow Overview

1. Complete the CDS data submission - CDS data must locked, approved, and posted prior to running SSRYTSD because Term/Section Data depends on the existence of Course Catalog and Student data. The SDS End of Term submission should not be locked, but the submissions should be "mostly clean."

2. Run the SICAS Term/Section Data Submission Report SSRYTSD in Audit Mode
    a. Use the Audit Replace Mode (AR) the first time run for any given term
    b. Use the Audit Update Mode (AU) thereafter. This mode only builds changes

3. Review and Confirm the resulting data using the Data Review tab of the SOAYDTI Form. Make corrections as needed

4. If errors in the Course Catalog or Student Data are discovered while running TSDS, the CDS batch must be unposted (UU or UR) and the data can be retransmitted. This new CDS submission must be locked, approved and posted in order to clear the TSDS errors. If there are TSDS errors that require retransmitting data from the SDS submission re-run the process. It is best to leave your SDS EOT unlocked but if your End of Term SDS has been locked, have the IR Staff at System Administration unlock it. The IR staff at System Administration should not approve your End of Term SDS submission until your TSDS submission is ready to be locked. Once TSDS is approved, CDS can no longer be unposted. All submissions are final.

5. Run SSRYTSD in Transmit Mode to send data to the SUNY Data Warehouse staging area. For the initial transmission of data, use the same action (Replace or Update) that was used in Step 2

6. Review Errors on the Processing Errors tab of the SOAYDTI form - make corrections as needed
    • To maintain synchronization, errors must be corrected in Banner and not with the file correction tools on the System Administration website

7. Use Transmit Update Mode (TU) to send corrected data - repeat until all errors are resolved
    • Using Update Mode speeds the processing because only updates are sent; however, please be aware that new records will also be sent. This includes new registrations.

8. Use the SUNY System Administration Web page to review and lock the batch for approval

9. If necessary, resolve any problems identified by SUNY IR staff. Re-transmit and re-lock the batch

10. After SUNY IR staff Approves the batch, run SSRYTSD in Post Batch Mode (PO) to update the Data Posted flag on the SICAS Term/Section Data Submission extract tables

FADS Overview

The SICAS Financial Aid Data Submission (FADS) application extracts student data from Banner. The financial aid information, demographic features and other characteristics collected are used to develop a large number of annual descriptive statistical summaries used by the Office of Institutional Research for planning, academic and fiscal administration, budget process, legal obligations, and reporting requirements imposed by State and Federal agencies. The FADS data will be combined with the data from the Student Revenue Data Submission (SRDS) to produce the Cost Calculator required by the Higher Education Opportunities Act (HEOA).

The SICAS Financial Aid Data Submission Report RORYFAD collects the FADS extract data, transmits the data to the data warehouse staging area, and produces a listing of data and transmission status. Any errors returned when RORYFAD is run in Transmit Mode are inserted into the SICAS Data Selection and Conversion Error Table SFRYERR and can be viewed on the Processing Errors tab of the SICAS Data Transfer Interface Control Form ROAYDTI. FADS must be configured before use. The Set Up section provides the information to configure the FADS application.

After Set Up is complete, run the RORYFAD process in Audit Mode. Review the results to determine that the Select Code is working properly. To view the Select Code, navigate to the Select Code tab, which is located on the lower portion of the Column Definitions tab on the SICAS Data Transfer Interface Control Form ROAYDTI.

The RORYFAD process must be run for <u>one aid year at a time</u> and will be reporting in the fall only for the prior Aid Year that has just concluded (i.e. Fall of 2011 – schools will be reporting the Aid Year 2010-2011).

Refer to the manual for additional information on any of the items below.

<u>FADS Set Up Overview</u>

1. Review Column Definitions on the Column Definitions tab of the ROAYDTI form

2. Enter Primary and Secondary Contact Information on the Contact Information tab of the ROAYDTI form. Primary Contact information is required for FADS

3. Perform all required crosswalks on the Code Crosswalk tab of the ROAYDTI form including all data elements that have a Crosswalk Source of GTVYVAL

4. Set up the required Fund Categories on the Banner Fund Category Validation Form RTVFCAT

5. Link the Fund Codes set up on RTVFCAT to appropriate Associate Fund Codes on RFRBASE.

6. If a campus has a Winter Term that students register for or aid was given for in an Aid Period, that Winter Term must be added on STVYTTY (starting with winter 2006).

FADS Process Flow Overview

The FADS process can only be run for one Aid Year at a time. All errors must be resolved and the file must be locked, approved, exported and posted before moving on to another Aid Year.

1. Run the SICAS Financial Aid Data Submission Report RORYFAD. The INITIAL RUN should be done in Audi Replace (AR mode. Thereafter:

    a. Use the Audit Replace Mode (AR) the first time the process is run for any given term.
    b. Use the Audit Update Mode (AU) thereafter – only builds changes.

    NOTE: No data is transmitted in either AU or AR modes and because of this no edits are applied to the data. Audit mode is only used to review the data and verify that the elements are being collected. To have data tested against the business rules, it must be transmitted to System Administration.

2. Review and confirm the resulting Student Data using the Data Review tab on the ROAYDTI form and make corrections as needed.

3. Run RORYFAD in Transmit Replace (TR) mode for INITIAL TRANSMISSION OF DATA to the SUNY Data Warehouse (Also referred to as the DTS site) staging area. Thereafter:

    a. Use the Transmit Update Mode (TU) to make corrections to the batch and transmit
    b. Use the Transmit Replace Mode (TR) to make replace the entire batch and transmit

4. Review Errors on the Processing Errors tab of the ROAYDTI form and make corrections as needed.

    NOTE: To maintain synchronization, errors must be corrected in Banner and not with the file correction tools on the System Administration website.

5. Use Transmit Update Mode (TU) to send corrected data. Repeat until all errors are resolved.

    NOTE: Using Update Mode speeds the processing because only updates are sent.

6. Use the SUNY System Administration Web page to review and Lock the batch for approval. Once locked, System Administration must approve and export the data before the campus can post the data in Banner and move to a new Aid Year. Only one AY can be processed at a time.

7. If necessary, to resolve any problems identified by SUNY IR staff, the SUNY IR Staff must un-lock the batch, the campus must re-transmit and re-lock the batch. However, if the batch has been approved by System Administration, it can no longer be unlocked and no changes can be made.

8. After SUNY IR staff Approves the batch and exports the data, Run RORYFAD in Post Batch Mode (PO) to update the Data Posted flag on the SICAS Student Data Submission extract tables.

NOTE: On the ROAYDTI form—Primary Contact information is required for FADS.

For the AY 2011-2012 and going forward, all Cost Of Attendance (COA) elements must be broken out in Banner and reported to SIRIS in the following manner:

1. Tuition
2. Student Fees
3. Room and Board
4. Books and Supplies
5. Personal Expenses
6. Transportation
7. Other Expenses

If an institution cannot provide all seven COA elements broken out as shown above for the AYs 2006-2010 (the past four AYs) – then DO NOT report any of these elements in FADS for those AY's. You MUST break the COA elements out beginning with the 2011-2012 AY.

SRDS Overview

The SICAS Student Revenue Data Submission (SRDS) application extracts student data from Banner. The student revenue information, demographic features, and other characteristics collected are used to develop a large number of annual descriptive statistical summaries used by the Office of Institutional Research for planning, academic and fiscal administration, budget process, legal obligations, and reporting requirements imposed by State and Federal agencies. The SRDS data will be combined with the data from the Financial Aid Data Submission (FADS) to produce the Cost Calculator required by the Higher Education Opportunities Act (HEOA).

The SICAS Student Revenue Data Submission Report TSRYSRD collects the SRDS extract data, transmits the data to the data warehouse staging area, and produces a listing of data and transmission status. Any errors returned when TSRYSRD is run in Transmit Mode are inserted into the SICAS Data Selection and Conversion Error Table SFRYERR and can be viewed on the Processing Errors tab of the SICAS Data Transfer Interface Control Form TOAYDTI. SRDS must be configured before use. The Set Up section provides the information to configure the SRDS application.

After Set Up is complete, run the TSRYSRD process in Audit Mode. Review the results to determine that the Select Code is working properly. To view the Select Code, navigate to the Select Code tab, which is located on the lower portion of the Column Definitions tab on the SICAS Data Transfer Interface Control Form TOAYDTI.

There will be two submission periods for each academic year with one period encompassing charges from July 1, XXXX (XXXX designates the four-digit year) through December 31, XXXX and the second period encompassing charges from January 1, XXXX through June 30, XXXX. The format for the period will consist of MMDDYYYY. The July 1 to December 31 period will be 1231XXXX and the January 1 to June 30 period will be 0630XXXX. The July 1 to December 31 period includes charges for summer and fall and could include charges for winter and spring. The January 1 through June 30 period includes charges for winter and spring and could include charges for summer and fall depending on when billing takes place for each semester.

NOTE: The TSRYSRD process must be run for one submission period at a time. Do not run multiple submissions.

<u>SRDS Set Up Overview</u>

1. Go to the SIRS interface type using the SOAYDTI form to review and update the Campus Type Element. Select the correct Campus Type Element based on the following:

    SO - State Operated campuses

SN - State Operated campuses *not* using URAS will select this code
CC - Community Colleges

Save the changes made. This entry on SIRS will allow the program to decide what data to collect based on what type of campus you are.

2. Review Column Definitions on the Column Definitions tab of the TOAYDTI form.

3. Enter Primary and Secondary Contact Information on the Contact Information tab of the TOAYDTI form. Primary Contact information is required for SRDS

4. Perform all required crosswalks on the Code Crosswalk tab of the TOAYDTI form including all data elements that have a Crosswalk Source of GTVYVAL. Crosswalk the required Revenue Categories (Rev Cat) on the Code Crosswalk tab of the TOAYDTI form for all revenue categories that should be reported according to the data dictionary.

SRDS Process Flow Overview

The SRDS process can only be run for one submission period at a time. All errors must be resolved and the file must be locked, approved, exported and posted before moving on to the next academic submission period.

1. Run the SICAS Student Data Submission Report TSRYSRD. The initial run should be done in Audit Replace (AR) Mode. Thereafter:

   a. Use the Audit Replace (AR) mode for the first time run for any given term.
   b. Use the Audit Update (AU) mode thereafter— this mode only builds changes

   NOTE: No data is transmitted in either AU or AR modes and because of this no edits are applied to the data. Audit mode is only used to review the data and verify that the elements are being collected. To have data tested against the business rules, it must be transmitted to System Administration.

2. Review and Confirm the resulting Student Revenue Data using the Data Review tab on the TOAYDTI form and make corrections as needed.

3. Run TSRYSRD in Transmit Replace (TR) mode for initial transmission of the data to the SUNY Data Warehouse (also referred to as the DTS site) staging area. Thereafter:

   a. Use the Transmit Update (TU) mode to make corrections to the Batch and Transmit.
   b. Use the Transmit Replace (TR) mode to replace the entire Batch and Transmit.

4. Review errors on the Processing Errors tab of the TOAYDTI form and make corrections as needed.

   NOTE: To maintain synchronization, errors must be corrected in Banner and not with the file correction tools on the System Administration website.

5. Use Transmit Update (TU) Mode to send corrected data—repeat until all errors are resolved.

   NOTE: Using Transmit Update (TU) mode speeds the processing because only updates are sent. However, please be aware that new records will also be sent - this includes new revenue transactions.

6. Use the SUNY System Administration Web page to review and Lock the batch for approval. Once locked, System Administration must approve and export the data before the campus can post the data in Banner and move to the next submission period.

    NOTE: Only one submission period can be processed at a time.

7. If after the batch has been locked, there are problems identified by SUNY IR staff, have the IR Staff un-lock your batch, then re-transmit and re-lock the batch. However, if the batch has been approved by System Administration, it can no longer be unlocked and no changes can be made. For this reason, it is critical that each campus review their data for accuracy using the SUNY dashboards BEFORE locking the file.

8. After SUNY IR staff approves the batch and exports the data, run TSRYSRD in Post Batch (PO) mode to update the Data Posted flag on the SICAS Student Data Submission extract tables

9. If, during a SRDS transmission, you receive an error relative to a stranded batch, follow the instructions located in the SORYDTS SICAS DTS Batch Finalization Process section of this manual. Once you have successfully completed the SORYDTS steps, run TSRYSRD in TR mode.

    NOTE: On the TOAYDTI form – Primary Contact information is required for SRDS.

DADS Overview

The SICAS Degree Awarded Data Submission (DADS) application extracts student data from Banner. The student degree information, demographic features, and other characteristics collected are used to develop a large number of annual descriptive statistical summaries used by the Office of Institutional Research for planning, academic and fiscal administration, budget process, legal obligations, and reporting requirements imposed by State and Federal agencies.

The SICAS Degree Awarded Data Submission Report SHRYDAD collects the DADS extract data, transmits the data to the data warehouse staging area, and produces a listing of data and transmission status. Any errors returned when SHRYDAD is run in Transmit Mode are inserted into the SICAS Data Selection and Conversion Error Table SFRYERR and can be viewed on the Processing Errors tab of the SICAS Data Transfer Interface Control Form SOAYDTI. DADS must be configured before use. The Set Up section provides the information to configure the DADS application.

After Set Up is complete, run the SHRYDAD process in Audit Mode. Review the results to determine that the Select Code is working properly. To view the Select Code, navigate to the Select Code tab, which is located on the lower portion of the Column Definitions tab on the SICAS Data Transfer Interface Control Form SOAYDTI.

There will be one primary submission period for the academic year that will collect and submit degrees awarded from October 1, 20XX through June 30, 20YY. The secondary submission period will collect and submit the population of students who were awarded degrees in the summer, July 1, 20YY through September 30, 20YY. Only new students to the population will be sent in the secondary submission.

NOTE: The SHRYDAD process must be run for one submission period at a time. Do not run multiple submissions.

DADS Set Up Overview

1. Review Column Definitions on the Column Definitions tab of the SOAYDTI form.

2. Enter Primary and Secondary Contact Information on the contact Information tab of the SOAYDTI form – Primary Contact Information is required for DADS.

3. Perform all required crosswalks on the Code Crosswalk tab of the SOAYDTI form including all data elements that have a Crosswalk Source of GTVYVAL. Crosswalk the required Award, Award Level, Hispanic Origin, and Sub-Campus codes on the Code Crosswalk tab of the SOAYDTI form for all data elements that should be reported according to the data dictionary.

DADS Process Flow Overview

The DADS process can only be run for one submission period at a time. All errors must be resolved and the file must be locked, approved, exported and posted before moving on to the next academic submission period.

1. Run the SICAS Degree Awarded Submission Report SHRYDAD. The initial run should be done in Audit Replace (AR) Mode. Thereafter:

    a. Use the Audit Replace (AR) mode for the first time run for any given term.
    b. Use the Audit Update (AU) mode thereafter— this mode only builds changes

    NOTE: No data is transmitted in either AU or AR modes and because of this no edits are applied to the data. Audit mode is only used to review the data and verify that the elements are being collected. To have data tested against the business rules, it must be transmitted to System Administration.

2. Review and Confirm the resulting Degree Awarded Data using the Data Review tab on the SOAYDTI form and make corrections as needed.

3. Run SHRYDAD in Transmit Replace (TR) mode for initial transmission of the data to the SUNY Data Warehouse (also referred to as the DTS site) staging area. Thereafter:

    a. Use the Transmit Update (TU) mode to make corrections to the batch and transmit.
    b. Use the Transmit Replace (TR) mode to replace the entire batch and transmit.

4. Review errors on the Processing Errors tab of the SOAYDTI form and make corrections as needed.

    NOTE: To maintain synchronization, errors must be corrected in Banner and <u>not with the file correction tools on the System Administration website.</u>

5. Use Transmit Update (TU) Mode to send corrected data—repeat until all errors are resolved.

    NOTE: Using Transmit Update (TU) mode speeds the processing because only updates are sent. However, please be aware that new records will also be sent - this includes new revenue transactions.

6. Use the SUNY System Administration Web page to review and Lock the batch for approval. Once locked, System Administration must approve and export the data before the campus can post the data in Banner and move to the next submission period.

    NOTE: Only one submission period can be processed at a time.

7. If after the batch has been locked, there are problems identified by SUNY IR staff, have the IR Staff un-lock your batch, then re-transmit and re-lock the batch. However, if the batch has been approved by System Administration, it can no longer be unlocked and no changes can be made. For this reason, it is critical that each campus review their data for accuracy using the SUNY dashboards BEFORE locking the file.

8. After SUNY IR staff approves the batch and exports the data, run SHRYDAD in Post Batch (PO) mode to update the Data Posted flag on the SICAS Student Data Submission extract tables

    NOTE: On the SOAYDTI form – Primary Contact information is required for DADS.

SET UP

The Set Up section provides detailed information on the Set Up requirements for these Interface Types:

- SIRS   SIRIS IR Submission Support
- CDS   Course Data Submission
- SDS   Student Data Submission
- TSDS   Term/Section Data Submission
- FADS   Financial Aid Data Submission
- SRDS   Student Revenue Data Submission
- DADS   Degree Awarded Data Submission

Terms MUST be processed in sequential order for each submission. For example, if you POST your winter EOT, you will be prevented from submitting an EOT file for the previous fall semester. This is true for all of the submissions except FADS which is not term based.

SIRS SIRIS IR Submission Support

The SIRS – SIRIS IR Submission Support interface contains information common to the Data Submission Interface Types.

Open the SICAS Data Transfer Interface Control Form SOAYDTI (Figure 1 SOAYDTI - SIRS Interface displayed on the SOAYDTI Form) and enter SIRS as the Interface Type in the Key Block.

The Column Definitions tab displays data elements that must be cross walked. The data elements that appear on the SIRS interface Column Definitions tab only need to be cross walked once using the SIRS Code Crosswalk tab. Data elements cross walked on the SIRS interface are not available on the individual data submission interfaces (i.e. SDS or TSDS); therefore, the SIRS Code Crosswalk tab must be reviewed to ensure that these data elements are cross walked.

The NATION_CNV data element is common to SDS and TSDS. STATE_CNV and COUNTY_CNV crosswalks are required for SDS. Use the Code Crosswalk tab on the SIRS Code Crosswalk tab to complete the crosswalks for all data elements that appear on the Column Definitions tab.
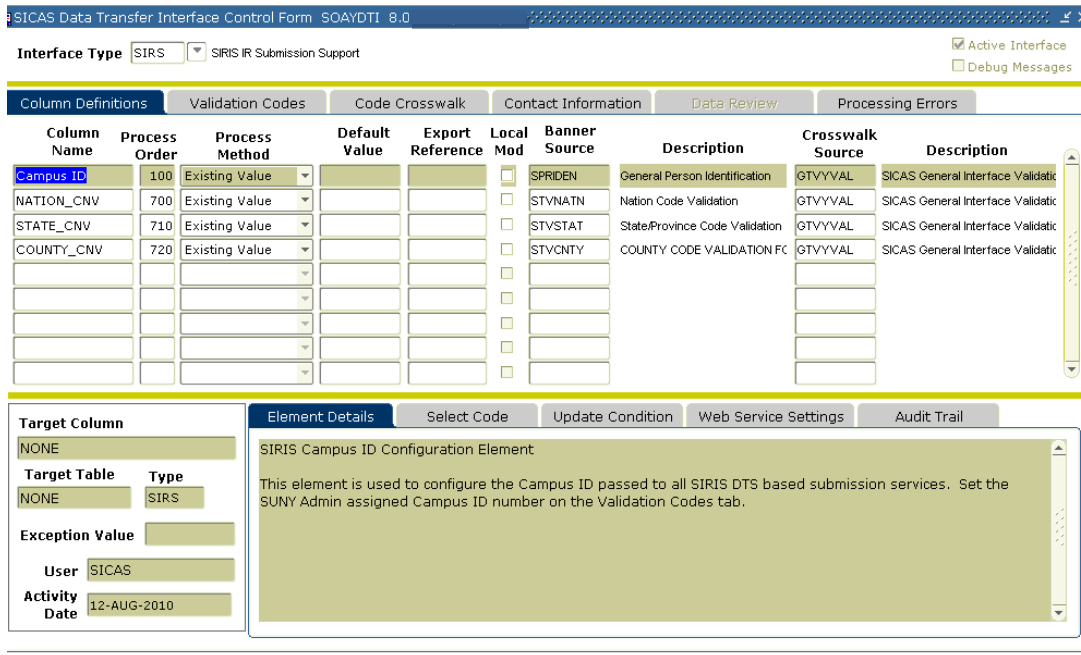
Figure 1 SOAYDTI - SIRS Interface displayed on the SOAYDTI Form

The Campus ID is used by all SIRIS submissions. This code is used to identify your campus to System Administration's Data Transfer System (DTS) when you send a SIRIS Submission. Appendix A contains a list of all SUNY Campus ID's.

The NATION_CNV element is used to crosswalk Banner Nation Codes to System Administration Codes. The NATION_CNV data element crosswalk is required for both SDS and TSDS.

The STATE_CNV element is used to crosswalk Banner State/Province codes (spraddr_stat_code) to System Administration Codes. All U.S. State Codes must be cross walked to the System Administration U.S. State Codes, even if they are the same. Canadian Province Codes must be cross walked to System Administration Canadian province codes. U.S. Military bases and U.S. Territories must be cross walked to 'OT' (Other). Currently, only provinces that are identified with a code in the State/Province field of the Banner address are converted and sent in the SDS data submission. The STATE_CNV data element crosswalk is required for SDS.

The COUNTY_CNV element is used to crosswalk Banner County Codes to System Administration Codes. The COUNTY_CNV data element crosswalk is required for SDS.

CDS Course Data Submission
The Course Data Submission (CDS) application requires configuration before use.

The SICAS Data Transfer Interface Control Form SOAYDTI is used for most of the configuration steps but configuration is also required on the Basic Course Information Form SCACRSE. There is more information in the SOAYDTI and the SCACRSE section later in this document. These configuration steps must be completed before using the Course Data Submission application.

To access the CDS interface, open the SICAS Data Transfer Interface Control Form SOAYDTI and enter CDS as the Interface Type in the Key Block.

CDS Set Up
    1.   Review Column Definitions on the Column Definitions tab of the SOAYDTI form

The CDS column information should be reviewed for accuracy and completeness. The Column Definitions can be viewed on SOAYDTI's Column Definitions tab (Figure 17). For more information on the Column Definitions Tab see the SOAYDTI – Form Details section 'Column Definitions Tab'.

2. On the SOAYDTI form, Interface Type SIRS, enter the SUNY Campus ID on the Validation Codes tab - see Appendix A for the correct SUNY Campus ID code

   The first record on the Validation Codes tab of the SOAYDTI form (Figure 25) contains the SUNY Campus ID code and must be updated prior to using CDS. The Campus ID is a unique identifier assigned to the SUNY Campus. Appendix A contains a list of SUNY Campus ID codes used for all IR submissions. To update the Campus ID, access the SIRS interface on SOAYDTI and enter your Campus ID on the crosswalk tab.

3. Cross walk the Classification Codes for Teacher Certification courses on the Code Cross walk tab of the SOAYDTI form

   A single required crosswalk must occur on the Code Crosswalk tab (Figure 26) of the SOAYDTI form before using CDS. This crosswalk determines which courses are Teacher Certification related. An Interface Code of four indicates that a course is NOT related to Teacher Certification and an Interface Code of one indicates that it is a Teacher Certification related course. For more information on the Code Crosswalk Tab see the SOAYDTI – Form Details section 'Code Crosswalk Tab'.

4. Enter Primary and Secondary Contact Information on the Contact Information tab of the SOAYDTI form - Primary Contact information is required for CDS

   Primary Contact information must be updated prior to using the CDS application. The Contact Information tab of the SOAYDTI form (Figure 27) allows the user to enter one Primary Contact (required) and multiple Secondary Contacts (optional). The Name, Job Title, E-Mail Address, Phone, and User ID are required fields and must be completed. The Contact Address field is not in use at this time and does not need to be completed. For more information on the Contact Information Tab, see the SOAYDTI – Form Details section 'Contact Information Tab'.

5. Review the General Ed Approved course indicator and enter information as necessary on the Basic Course Information Form SCACRSE

   Review the General Ed Approved course indicator to verify that all General Education courses are correctly identified and enter required course information on the SCACRSE form. For more information, see the SCACRSE Basic Course Information Form section later in this document.

6. Enter the required codes on the Crosswalk Validation Form GTVSDAX

   Use the Crosswalk Validation Form GTVSDAX to enter the name of the institution's production Banner database. This must be done before using Web Services to transmit data. The Internal Code is PROD_DB, the Internal Group is SICAS_IR, and the External Code is the Instance Name of the production database. Do not alter this setting unless the name of the production database is changed.

For a comprehensive list of GTVSDAX entries required for the SIRIS Submissions and details on setting these values, see the GTVSDAX Crosswalk Validation Form section later in this document.

CDS Transmission Status:

Ready   The course has been collected in the CDS process and is ready to transmit.
Current   The course has been collected in the CDS process and will not be sent because there have been no changes. This is done so that a campus can extract a full version of their catalog. The posted status will remain No until the entire file is posted.
Non-CDS   The course has been mark with a status of "Do Not Send" on SCACRSE.
Warning   A warning has been received from System Administration during the transmission.
Error   A fatal error has been received from System Administration during the transmission.

SDS Student Data Submission

The Student Data Submission (SDS) application requires configuration before use. Additional information is available later in this document for the SICAS Data Transfer Interface Control Form SOAYDTI, the Basic Course Information Form SCACRSE, the Schedule Form SSASECT, and the Crosswalk Validation Form GTVSDAX. The SDS set up requirements for the General Person Form SPAPERS (Ethnicity, SUNY Hispanic Origin, Race), the Race Code Rules Form GORRACE (IPEDS code crosswalk), the Curriculum Rules Form SOACURR (APES Codes and APES Award Levels), the Student Course Registration Form SFAREGS (Study Abroad Code), the SUNY Student Educational Goal Form SOAYEDG (Educational Goal), and the Prior College Form SOAPCOL (Prior Institution, Home Institution) are in this section.

NOTE:

APES and Award Level information is on the ASC/SDS tab on SOACURR. It is easier for campuses to see which majors have this information added by using the SOIYCUR form rather than having to go into each major through the SOACURR form. The SOIYCUR form displays SICAS fields related to the curriculum information.

SICAS provides three Local Campus Modification functions for SDS. The FZ_SDS_Admission_Status is required and must be coded to return the student's Admission Status. The FZ_SDS_BuildStudents and FZ_SDS_IncludeStudent functions are optional. For more information, see the Local Campus Modification Functions section and Appendix G later in this document.

The data elements Campus_ID, NATION_CNV, STATE_CNV, and COUNTY_CNV are cross walked on the SIRS – SIRIS IR Submission Support interface. To access the SIRS interface, open the SICAS Data Transfer Interface Control Form SOAYDTI and enter SIRS as the Interface Type in the Key Block. The data elements that appear on the Column Definitions tab of the SIRS interface do not appear on the Column Definitions tab of the SDS interface but they are required for SDS and must be cross walked on the Code Crosswalk tab of the SIRS interface. Data elements that are used by more than one Data Submission application—such as NATION_CNV which is used by both SDS and TSDS—only need to be cross walked once using the SIRS Code Crosswalk tab. Refer to the SIRS SIRIS IR Submission Support section under Set Up for more information.

The CITZ CNTRY element (gobintl_natn_code_legal) is the code in the 'Nation of Citizen' field on the Nationality Tab of the International Information Form GOAINTL. The COUNTRY element

(spraddr_natn_code) is used for both U.S. citizens and foreign students. The COUNTRY element refers to the country of the student's permanent address while the CITZ CNTY is the student's country of citizenship. It is not necessary to create a GOAINTL record for every student. If the SDS process does not find a GOAINTL record for a student it will send USA.

Currently, only test scores (TEST SCORE data element—sortest_test_score) are being submitted for admission criteria in the SDS data transmission. The CRITERIA data element (stvtesc_code) must be cross walked to the System Administration Test Codes on the Code Crosswalk tab of SOAYDTI. Only crosswalk the Test Codes marked as an 'Admissions Checklist Request Item' on the Test Code Validation Form STVTESC. The SICAS Student Data Submission Report SGRYSDS only sends the Test Scores associated with the cross walked Test Codes.

The SICAS Student Data Submission Report SGRYSDS requires a Start Date and an End Date for the Visa if a student has more than one Visa entered on the International Information Form GOAINTL. If only one Visa exists, the SGRYSDS process does not require a Start Date in the GOAINTL Visa Information Block. The SGRYSDS process also requires that the Visa be active during the semester you are processing. If the start or end date of a Visa is not within the start and end date of the semester, no visa will be sent for the student. If there are no start and end dates, and there is only one Visa code on GOAINTL, that Visa code will be sent.

The value submitted in the SDS data transmission for Higher Education History, data element HGH_ED_HIST, is determined by a combination of three data elements—the student's Level Code (levl_conv — sgvacur_levl_code), the Student Type (HEH_STYP — sobysds_stud_levl_conv||'-'||sgvacur_styp_code_stdn), and the Admit Type (HEH_ADMT — sgvacur_admt_code).

These are System Administration's values for Higher Education History:

1   Undergraduate First Time
2   Undergraduate Transfer
3   Undergraduate Transition
4   Undergraduate Continuing/Returning
5   Concurrently Enrolled in High School – This may be an Admit Type rather than a Student
           Type as some institutions
6   New Graduate
7   Graduate Continuing/Returning
8   Cross Registered – This may be an Admit Type rather than a Student Type at some
           institutions
9   Joint Program – This must be determined at each institution
10   Unknown

Some institutions may use the Admit Type rather than the Student Type. Each institution must decide if the Higher Education History values are determined by Admit Type or the combination of Level and Student Type or both. Crosswalk the elements used to determine the Higher Education History value. The data elements are cross walked on the Code Crosswalk Tab of the SOAYDTI form. If a campus chooses to only use one element, the process method for the other must be changed to "force to null." This is done by creating a local version of the element (insert a new record and duplicate the record) and changing the processing method. Changing the process method to "force to null" effectively "turns off" that element. It is also not necessary to crosswalk the element that is not in use.

The LEVL_CONV element must be cross walked if the institution uses Student Type to determine the Higher Ed History. Crosswalk the Level Code on the Level Code Validation Form STVLEVL to the System Administration values '01' for Undergraduate or '02' for Graduate.

The Inst IPED element is only sent for Undergraduate Transfer students (Higher Education History = 2) and New Graduate students (Higher Education History = 6). For New Graduate students, the Prior College entered on SOAPCOL should be the college that awarded their Bachelor's degree.

For a New Graduate student that received their Bachelor's degree from the same college they are attending for their Graduate degree, the Campus CEEB Code is translated to the IPEDS Code and sent as the Prior College. The CampusCEEB element (Banner Source of GTVYVAL) must be cross walked to the Campus IPEDS Code (Code from the STVSBGI form) on the SUNY IPEDS Code Validation Form STVYIPD.

The CampusCEEB element retrieves the Campus CEEB Code. The SameCollge element retrieves the Campus CEEB Code and converts it to the IPEDS Code for New Graduate students who received their Bachelor's degree from the same college they are attending for their Graduate degree. The application assumes the student received a Bachelor's degree from the same campus if the Degree and Other Formal Awards Form SHADEGR indicates that the student was awarded a Bachelor's degree.

The Special Population Code is defined on the Student Attribute Validation Form STVATTS. Attribute(s) are added to individual students on the Additional Student Information Form SGASADD for the appropriate term. These are the System Administration values for Special Population Code:

1   Special Auditor
2   State Aidable Auditor
3   College in the High School
4   First Generation College Student
5   Disabled
6   Displace Homemaker
7   Single Parent
8   Limited English Proficient
9   Economically Disadvantaged

To access the SDS interface, open the SICAS Data Transfer Interface Control Form SOAYDTI and enter SDS as the Interface Type in the Key Block.

SDS Set Up

1.   Enter the required codes on the Crosswalk Validation Form GTVSDAX

Use the Crosswalk Validation Form GTVSDAX to verify External Code values required for the SDS application. This table lists the Codes that must be reviewed and verified before running the SDS process SGRYSDS. External Codes may vary by institution. The user must enter Codes specific to their institution.

GTVSDAX Entries Required for the SDS Application		
Internal Code	**Internal Group**	**Displays External Code For**
PROD_DB	SICAS_IR	Name of Production Banner Database Required to use Web Services
USA	ADDRESS	Permanent address type for U.S. students Required for SDS
FOREIGN	ADDRESS	Permanent address type for Foreign students Required for SDS

PELL_FUND	SICAS_SDS	Pell Grant Fund Code Required for SDS
TAP_FUND	SICAS_SDS	TAP Grant Fund Code Required for SDS
Multiple	SICAS_TERM_TYPE	Last two digits of Term Code (multiple entries) Required for SDS and TSDS
CRSELVL_UG	SICAS_SDS	Level code for Undergraduate Students
CRSELVL_GR	SICAS_SDS	Level Code for Graduate Students

For a comprehensive list of GTVSDAX entries required for the SIRIS IR Submissions and details on setting these values, see the GTVSDAX Crosswalk Validation Form (Figure 40) section later in this document.

2. Mark courses included in the Course Data Submission (CDS) 'Ready' and enter a course level for those same courses on the Basic Course information Form SCACRSE

   Courses that are included in the Course Data Submission (CDS) must have an Action status of 'Ready' (Ready to Submit) and have a Course Level designated in the SUNY Institutional Research Data block of the Basic Course Information Form SCACRSE.

   For more information on the SCACRSE form, see the Basic Course Information Form SCACRSE section later in this document.

3. Mark the sections included in the Term/Section Data Submission (TSDS) 'Ready' on the Schedule Form SSASECT

   Sections included in the Term/Section Data Submission (TSDS) must have an Action status of 'Ready' (Ready to Submit) on the SUNY IR Submissions Detail block of the Schedule Form SSASECT. Only students enrolled in Sections that meet SIRIS criteria are included in the Student Data Submission (SDS).

   Refer to the Schedule Form SSASECT section, later in this document, for more information on the SSASECT form.

4. General Person Form SPAPERS

   The General Person Form SPAPERS contains selections for New Ethnicity, SUNY Hispanic Origin (enter if New Ethnicity indicates 'Hispanic or Latino'), and Race.

Figure 2 SPAPERS - General Person Form showing New Ethnicity, SUNY Hispanic Origin and Race values.

The General Person Form SPAPERS (Figure 2 SPAPERS - General Person Form showing New Ethnicity, SUNY Hispanic Origin and Race values.) New Ethnicity field displays the Federal category of 'Not Hispanic or Latino', 'Hispanic or Latino', or 'None'. If 'Not Hispanic or Latino' is selected in the New Ethnicity field on SPAPERS, a '1' is sent in the SDS data transmission for the Ethnicity data element. If the New Ethnicity field is null, an '8 – Unknown' is sent. If 'Hispanic or Latino' ethnicity is indicated, a SUNY Hispanic Origin may be entered. If a SUNY Hispanic Origin is selected, the crosswalk value for the Hispanic data element entered in the Interface Code field (the System Administration value) on the Code Crosswalk tab of SOAYDTI is sent in the SDS data transmission for the Hispanic data element. There is more information on this crosswalk in Step 12 - Code Crosswalk for the Hispanic Data Element.

The valid values for the SUNY Hispanic Origin field are found in the SICAS Ethnic Hispanic Origin Code Validation Table STVYETH. View this table using the SICAS General Validation Table Inquiry Form GTIYVTL by entering STVYETH as the Validation Table in the Key Block (Figure 3 GTIYVTL - SICAS General Validation Table Inquiry Form showing valid values for SUNY Hispanic Origin). SUNY Hispanic Origins are inserted as SPRYSID_SUNY_ID into the SICAS SUNY ID Repeating Table SPRYSID for the associated Banner PIDM with the SPRYSID_TYPE of HISP-ORIG.

Validation occurs between the New Ethnicity field and the SUNY Hispanic Origin field—an entry in the SUNY Hispanic Origin field is not allowed unless 'Hispanic or Latino' is selected for New Ethnicity however, the application does not enforce selection of a SUNY Hispanic Origin if the New Ethnicity field is 'Hispanic or Latino'. Remember to enter a selection for the SUNY Hispanic Origin when New Ethnicity is 'Hispanic or Latino'.

The error message 'NEW ETHNICITY column must be set to 'Hispanic or Latino' before committing this value' displays when a SUNY Hispanic Origin is entered without selecting 'Hispanic or Latino' for the New Ethnicity field.

The Race field pull-down list on the General Person Form SPAPERS (Figure 2 SPAPERS - General Person Form showing New Ethnicity, SUNY Hispanic Origin and Race values.) displays the Institutional Race Codes and Descriptions found on the Race Code Rules Form GORRACE (Figure 4 GORRACE - Race Code Rules Form). An individual may have more than one Race selected. If Race is   unknown, leave the field blank. A Code of 6 – Unknown is sent if a Race is not entered.

The SICAS Student Data Submission Report SGRYSDS crosswalks the entries for the Race field on SPAPERS (the Institutional Race Code on GORRACE - gorrace_race_code) to the Regulatory Race Codes on the GORRACE (gorrace_rrac_code). The Regulatory Race code on GORRACE is translated by the SICAS Student Data Submission Process SGRYSDS to the System Administration value before it is sent in the SDS data transmission for the Race data element.

Figure 3 GTIYVTL - SICAS General Validation Table Inquiry Form showing valid values for SUNY Hispanic Origin

5. Cross walk institution Race Codes to the Regulatory Race Codes on the Race Code Rules Form GORRACE

    Crosswalk Institution Race Codes to Regulatory Race Codes on the Race Code Rules Form GORRACE (Figure 4 GORRACE - Race Code Rules Form).

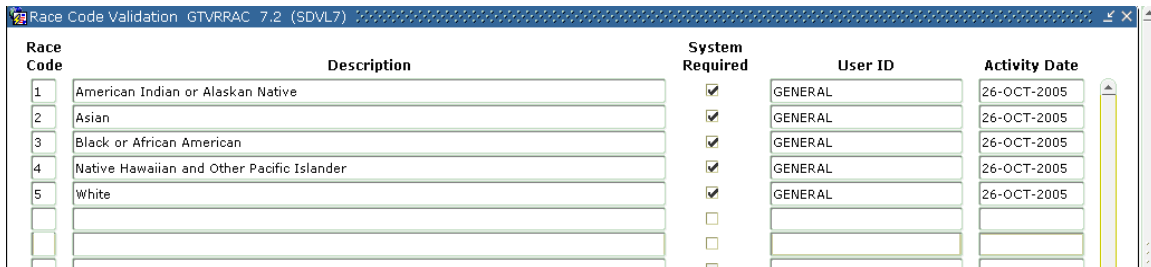**Figure 4 GORRACE - Race Code Rules Form**

An institution may add additional Institutional Race codes and Descriptions on the GORRACE form (Figure 4 GORRACE - Race Code Rules Form). Any Institution Race information added to GORRACE will appear in the pull-down list for the Race field on

SPAPERS (Figure 2 SPAPERS - General Person Form showing New Ethnicity, SUNY Hispanic Origin and Race values.

All Institutional Race codes must be cross walked to a Regulatory Race code on GORRACE. The Regulatory Race code on GORRACE is translated by the SICAS Student Data Submission Process SGRYSDS to the System Administration value before it is sent in the SDS data transmission for the Race data element.

The Regulatory Race codes and Descriptions on the GORRACE form are validated against the Race Code Validation Form GTVRRAC (Figure 5 GTVRRAC - Race Code Validation Form). Only the codes listed in Figure 5 GTVRRAC - Race Code Validation Form should be used. Campuses should not add Race codes to those mandated by the Federal Government, this includes adding a code of "unknown" or "Opt Out."

Please note that when ellucian shipped the Regulatory Race codes in the Student 7.2 release, they shipped the codes in the wrong order. The SICAS SDS process will convert the ellucian Race codes to the actual Regulatory codes during processing. These codes will be stored in the SORYSDS table.

Figure 5 GTVRRAC - Race Code Validation Form

SICAS shipped a web survey that can be used to collect the Race, Ethnicity and Hispanic Origin data from campus constituents.   It is possible to force users to see the survey. For information on using SICAS forced surveys, please refer to the General Web Application Toolset manual on the SICAS Center website.

Enter an individual's Ethnicity, Hispanic Origin, and Race information on the General Person Form SPAPERS

6. Enter APES Codes and APES Award Levels in the SUNY SDS/ASC Data block on the Majors and Departments tab of the Curriculum Rules Form SOACURR.

Enter the Academic Program Enterprise System (APES) Codes and the APES Award Levels in the SUNY SDS/ASC Data block on the Majors and Departments tab of the Curriculum Rules Form SOACURR. To access the SDS/ASC Data block, click on the SDS/ASC button on the Majors and Departments tab. It is easier for campuses to see which majors have this information added by using the SOIYCUR form rather than having to go into each major through the SOACURR form. The SOIYCUR form displays SICAS fields related to the curriculum information.

The APES Codes and the APES Award Levels are stored in the SICAS Curriculum/Major Shadow Table SORYCMJ (compliments the SORCMJR table). Be sure you are adding the APES codes and award levels to the latest version of the major by checking the "From" and "To" terms in the data block on the form. The "From" term in **Error! Reference source not found.** is indicated by the red arrow.

An APES code must be entered for each major within a program related to SIRIS. Enter the Award Level for the program in the Award Level 1 field (Award Level 1 is the first box). If the program only has one Award Level, leave the Credit field blank (the Credit field is the second box). If it is a Combined Program and awards two different degrees, the second Award Level is entered in the Award Level 2 field (the Award Level 2 field is the third box). In this case, set the number of Credits equal to the number of credits needed to obtain the first Award Level. If the student's accumulated credits exceed this number, the Award Level 2 field is sent as the Award Level element in the SDS submission.

Figure 6 SOACURR - Curriculum Rules Form showing the SUNY SDS/ASC Data Block

All students who have chosen a major must have an APES Program Code entered that corresponds to that specific major.

If a student is 'In a Program, Major Not Chosen', enter the appropriate APES Award Level (2 = Associate or 3 = Baccalaureate) and leave the APES Program Code blank. If a student is 'Not in a Program', enter an APES Award Level of '8' and leave the APES Program Code blank.

System Administration is using APES codes to replace APIS codes. The APES codes are just numbers and are not a concatenation of meaningful information like the APIS codes (i.e. Award Level, Campus, Curriculum, and Degree).

The APES Codes are validated on the SUNY APES Code Validation Form STVYAPE. If APIS codes have been set up, they are cross walked to APES codes during the installation of the SDS application and can be viewed and edited on the STVYAPE form.

The APES Codes on the STVYAPE (Figure 7 STVYAPE - SUNY APES Code Validation Form) form must be reviewed for accuracy (even if there is no crosswalk)—make adjustments as necessary.

Figure 7 STVYAPE - SUNY APES Code Validation Form

7. Enter a Study Abroad Code for any student enrolled in a study abroad program on the Registration tab of the Student Course Registration Form SFAREGS.

   Enter a Study Abroad Code for any student enrolled in a study abroad program on the Registration tab of the Student Course Registration Form SFAREGS (Figure 8 SFAREGS Student Course Registration Form showing the Study Abroad Code on the Registration Tab). The Study Abroad Code is stored, for the student by term (entered in the Key Block), in the   SUNY Repeating Data table associated with the Person table SPRYVAL—the SPRYVAL_TYPE = 'STDY_ABRD' and the SPRYVAL_VALUE = the Study Abroad Code. The Study Abroad Code is validated against the SICAS SDF Study Abroad Code Validation table STVYSSA.

Figure 8 SFAREGS Student Course Registration Form showing the Study Abroad Code on the Registration Tab

8. Enter the student's Education Goal on the SUNY Student Educational Goal Form SOAYEDG (Figure 9 SOAYDEG - SUNY Student Educational Goal Form) or collect the information using the SICAS Student Educational Goal Survey.

Figure 9 SOAYDEG - SUNY Student Educational Goal Form

Use the pull-down list for the SOAYEDG Educational Goal field to select an Educational Goal from a list of valid entries.

Valid Entries:

1A Transfer to another SUNY college after earning a degree/certificate
1B Transfer to a non-SUNY college after earning a degree/certificate
2A Transfer to a SUNY college without earning a degree/certificate
2B Transfer to a non-SUNY college without earning a degree/certificate
3   Earn a degree/certificate and seek employment rather than pursue further post-secondary education
4   Learn new skills or upgrade existing skills without earning a degree/certificate
5   Seek enrichment rather than to pursue a degree/certificate
6   Obtain a GED through the accumulation of college credits
7   Uncertain

The Educational Goal codes are stored in the SICAS Educational Goal Validation Table STVYEDG. This table serves as a crosswalk to the System Administration values (SUNY_Code); it is the SUNY Code that is sent in the SDS submission. Inactive Educational Goal codes are indicated with an Active code of 'N'. The STVYEDG validation table can be viewed using the SICAS General Validation Table Inquiry Form GTIYVTL (Figure 10 GTIYVTL - SICAS Educational Goal Validation Table STVYEDG as viewed on the SICAS General Validation Table Inquiry Form.

Figure 10 GTIYVTL - SICAS Educational Goal Validation Table STVYEDG as viewed on the SICAS General Validation Table Inquiry Form

For more information on the SICAS Student Educational Goal Survey, refer to the Student Educational Goal Web manual on the SICAS website.

9. Enter an institution code in the Prior College field and indicate if it is the student's home Institution or a prior institution on the Prior college and degree tab of the Prior College Form SOAPCOL. This is for transfer students or students whose home institution is not this campus.

   Enter an institution code in the Prior college field and indicate if it is the student's Home Institution or a Prior Institution on the Prior college and Degree tab of the Prior College Form SOAPCOL (Figure 11 SOAPCOL - Prior College Form). This entry is for undergraduate Transfer students (Higher Education History = 2), New Graduate students (Higher Education History = 6), or students whose Home Institution is not this campus.

   For New Graduate students, the Prior College is the college that awarded their Bachelor's degree.

   The Home Institution box should be checked for those students who are matriculated at another institution and are taking courses from your institution.

   If Home Institution is checked, a type of PCOL-HOME is entered in the SPRYSID_TYPE field of the SICAS SUNY ID Repeating Table SPRYSID and the Institution Code is stored in the SPRYSID_SUNY_ID field. If the institution code is changed, the entry on SPRYSID is updated. Only one Home Institution can be specified.

According to the SUNY data dictionary, campuses must report a prior college for all transfer students. To accomplish this, all transfer students (undergrad and grad) must have a prior institution checked to indicate the last institution the student attended. The form will not permit more than one institution to be labeled as the prior institution.

If Prior Institution is checked, a type of PCOL-PRIOR is entered in the SPRYSID_TYPE field of the SICAS SUNY ID Repeating Table SPRYSID and the Institution Code is stored in the SPRYSID_SUNY_ID field. If the institution code is changed, the entry on SPRYSID is updated. Only one Prior Institution can be specified.

Figure 11 SOAPCOL - Prior College Form

10. Enter the SUNY Campus ID on the Validation Codes tab for the SIRS interface on the SOAYDTI form – see Appendix A for the correct SU"NY Campus ID code.

   The Campus ID is a unique identifier assigned to the SUNY Campus; it must be updated prior to using SDS. Appendix A contains a list of SUNY Campus ID codes used for all IR submissions. Update the Campus ID on the Validation Codes tab of the SOAYDTI form (Figure 12 SOAYDTI Form showing Interface Type SDS, Code Crosswalk Tab, Interface element Hispanic). Campus ID is found on the SIRS interface. Navigate to the Campus ID data element on the Validation Codes tab and enter the institution's Campus ID in the Code field.

11. Cross walk the Banner Codes for the Hispanic data element to System Administration values on the Code Crosswalk tab of the SOAYDTI form.

   The Hispanic data element must be cross walked to System Administration values on the Code Crosswalk tab of the SICAS Data Transfer Interface Control Form SOAYDTI.

   The Hispanic data element for SDS is not sent to System Administration but is used to convert the Banner Hispanic Codes (Banner Value) to System Administration's SUNY Hispanic Origin values (Interface Code) on the Code Crosswalk Tab of SOAYDTI.

Crosswalk the Banner Values (Banner Hispanic codes stored in the SICAS Ethnic Hispanic Origin Code Validation Table STVYETH) to the System Administration's values for the Hispanic ethnicity breakdown on the Code Crosswalk tab of SOAYDTI. Rollback and enter the Interface Element 'Hispanic' in the Query Block of the Code Crosswalk tab and then perform Next Block to display the Banner Hispanic Codes in the Banner Value field. Enter corresponding System Administration values in the Interface Code field. Use the Interface Code down arrow to select from a list of System Administration valid values for the Data Element 'Hispanic'.

For the Ethnicity data element, the SDS data transmission sends a 1 if 'Not Hispanic or Latino' is selected for the New Ethnicity field on the General Person Form SPAPERS (Figure 2 SPAPERS - General Person Form showing New Ethnicity, SUNY Hispanic Origin and Race values.). If the New Ethnicity field is null, an '8 - Unknown' is sent. If 'Hispanic or Latino' is selected, then the Interface Code for the System Administration value is sent in the SDS data submission for the Hispanic data element.
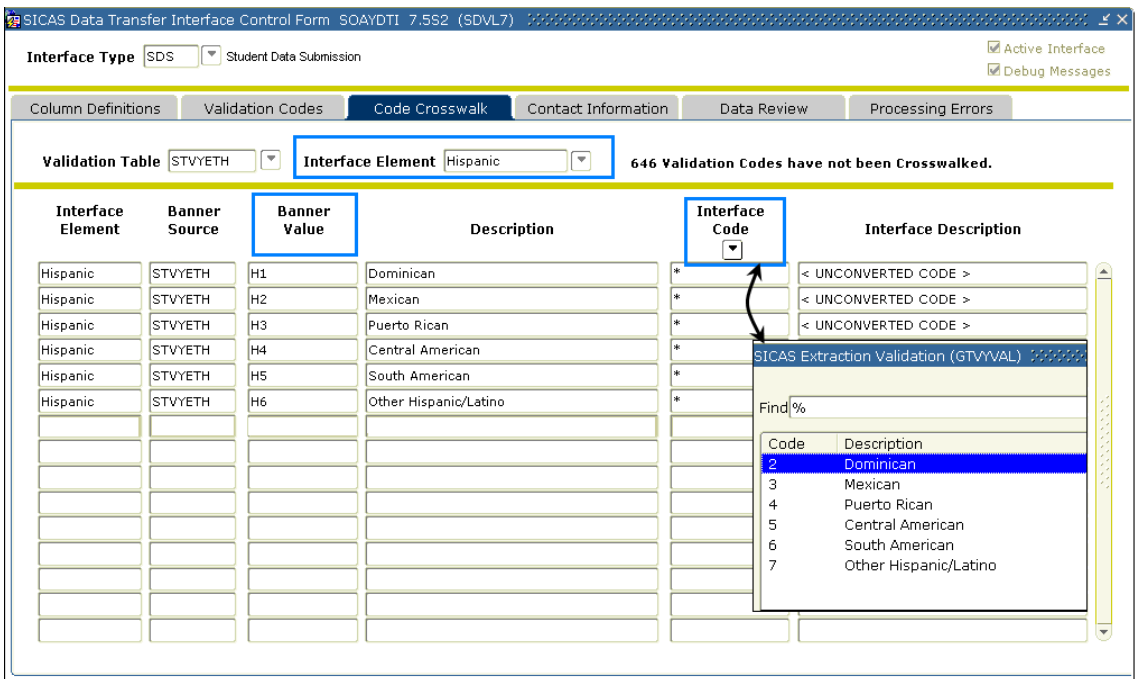
Figure 12 SOAYDTI Form showing Interface Type SDS, Code Crosswalk Tab, Interface element Hispanic

12. Use the Code Crosswalk tab of the SOAYDTI Form to crosswalk all data elements that have a Crosswalk Source of GTVYVAL. Data elements are listed on the SOAYDTI Column Definitions tab.

    On SIAYDTI, for the Interface Code SIRS, verify that the data elements in the Column Definitions are cross walked.

    All Data Elements in the Column Name field of the Column Definitions tab of the SICAS Data Transfer Interface Control Form SOAYDTI that have a Crosswalk Source of GTVYVAL must be cross walked to System Administration values on the Code Crosswalk tab of the SOAYDTI form. The System Administration values are defined on the Validation Codes tab of the SOAYDTI form.

    For more information on the Code Crosswalk Tab see the SOAYDTI – Form Details section 'Code Crosswalk Tab'.

Verify that the date elements CampusID, NATION_CNV, STATE_CNV, and COUNTY_CNV are cross walked on the SIRS – SIRIS IR Submission Support interface. To access the SIRS interface, open the SICAS Data Transfer Interface Control Form SOAYDTI and enter SIRS as the Interface Type in the Key Block. The data elements that are on the Column Definitions tab of the SIRS interface do not appear on the Column Definitions tab of the SDS interface but they are required for SDS and must be cross walked on the Code Crosswalk tab of the SIRS interface. Data elements that are used by more than one Data Submission application—such as NATION_CNV which is used by both SDS and TSDS—only need to be cross walked once using the SIRS Code Crosswalk tab.

Refer to the SIRS – SIRIS IR Submission Support section under Set Up for more information.

13. Enter a VISA code and a Country of Citizenship on the International Information Form GOAINTL for all foreign students. This information is required in the submission and is also used to determine what information should be transmitted in the address fields.

14. Enter Primary and Secondary Contact Information on the Contact Information tab of the SOAYDTI form. Primary Contact information is required for SDS.

Primary Contact information must be updated prior to using the SDS application. The Contact Information tab of the SOAYDTI form (Figure 12 SOAYDTI Form showing Interface Type SDS, Code Crosswalk Tab, Interface element Hispanic) allows the user to enter one Primary Contact (required) and multiple Secondary Contacts (optional). The Name, Job Title, E-Mail Address, Phone, and User ID are required fields and must be completed. The Contact Address field is not in use at this time and does not need to be completed.

For more information on the Contact Information Tab see the SOAYDTI – Form Details section 'Contact Information Tab'.

TSDS Term/Section Data Submission

The Term/Section Data Submission (TSDS) application requires configuration before use. The SICAS Data Transfer Interface Control Form SOAYDTI is used for most of the configuration steps but configuration is also required on the Basic Course Information Form SCACRSE, the Crosswalk Validation Form GTVSDAX, and the Schedule Form SSASECT. These configuration steps must be completed before using the Term/Section Data Submission application.

There is more information in the SOAYDTI – SICAS Data Transfer Interface Control Form section, the GTVSDAX – Crosswalk Validation Form section, the SCACRSE – Basic Course Information Form sections, and the SSASECT – Schedule Form section later in this document.

There are several elements that may or may not be required for your institution. For example the building and room meeting information is not required for Community Colleges and these elements need to have the processing method changed to "force to null." The instructor information is required for Community Colleges and the process method needs to be changed to "Use Select Code." This is done by creating a local modification and changing the process method. Instructions on how this is done can be found in the "Local Record Modification" section of this manual.

To access the TSDS interface, open the SICAS Data Transfer Interface Control Form SOAYDTI and enter TSDS as the Interface Type in the Key Block.

TSDS Set Up

1. Review Column Definitions on the Column Definitions tab of the SOAYDTI form

    The TSDS installation process attempts to extract and convert data for use in this application. The TSDS column information should be reviewed for accuracy and completeness. For more information on the Column Definitions Tab see the SOAYDTI – Form Details section 'Column Definitions Tab'.

2. Enter the SUNY Campus ID on the Validation Codes tab of the SOAYDTI form – see Appendix A for the correct SUNY Campus ID code

3. Primary Contact information must be updated prior to using the TSDS application. The Contact Information tab of the SOAYDTI form (Figure 27) allows the user to enter one Primary Contact (required) and multiple Secondary Contacts (optional). The Name, Job Title, E-Mail Address, Phone, and User ID fields are required and must be completed. The Contact Address field is not in use at this time and does not need to be completed. For more information on the Contact Information tab see the SOAYDTI – Form Details section 'Contact Information Tab'.

4. Perform all required crosswalks on the Code Crosswalk tab of the SOAYDTI form including all data elements that have a Crosswalk Source of GTVYVAL-Data elements are listed on the SOAYDTI Column Definitions tab; verify that the data elements on the SOAYDTI form SIRS interface Column Definitions tab are cross walked

    All Data Elements in the Column Name field of the Column Definitions tab of the SICAS Data Transfer Interface Control Form SOAYDTI that have a Crosswalk Source of GTVYVAL must be cross walked to System Administration values on the Code Crosswalk tab of the SOAYDTI (Figure 26) form. The System Administration values are defined on the Validation Codes tab of the SOAYDTI form.

    For more information on the Code Crosswalk Tab see the SOAYDTI – Form Details section 'Code Crosswalk Tab'.

    Verify that the date element NATION_CNV is cross walked on the SIRS – SIRIS IR Submission Support interface. To access the SIRS interface, open the SICAS Data Transfer Interface Control Form SOAYDTI (**Error! Reference source not found.**) and enter SIRS as the Interface Type in the Key Block. The NATION_CNV data element does not appear on the Column Definitions tab of the TSDS interface but it is required for TSDS and must be cross walked on the Code Crosswalk tab of the SIRS interface. Data elements that are used by more than one Data Submission application—such as NATION_CNV which is used by both TSDS and SDS—only need to be cross walked once using the SIRS Code Crosswalk tab. Refer to the SIRS – SIRIS IR Submission Support section under Set Up for more information.

5. Enter information as necessary on the Basic Course Information Form SCACRSE

    Enter required information on the SCACRSE form. For more information see the SCACRSE Basic Course Information Form section later in this document.

6. Enter section information as necessary on the Schedule Form SSASECT

    Enter required section information on the SSASECT form. For more information, see the SSASECT Schedule Form section later in this document.

7. Enter the required codes on the Crosswalk Validation Form GTVSDAX

Use the Crosswalk Validation Form GTVSDAX to verify External Code values required for the TSDS application. This table lists the Codes that must be reviewed and verified before running the TSDS process SSRYTSD. External Codes may vary by institution. The user must enter Codes specific to their institution.

GTVSDAX Entries Required for the TSDS Application		
Internal Code	**Internal Group**	**Displays External Code For**
PROD_DB	SICAS_IR	Name of Production Banner Database Required to use Web Services
FIRST_TERM	SICAS_SSBYSEC	The First Term to collect Term/Section data Required for TSDS
PSI_ACTIVE	SICAS_SLRRDEF	Report PSI information to System Admin (Y, N) Required for TSDS
Multiple	SICAS_TERM_TYPE	Last two digits of Term Code (multiple entries) Required for TSDS and SDS

For a comprehensive list of GTVSDAX entries required for the SIRIS Submissions and details on setting these values, see the GTVSDAX Crosswalk Validation Form (Figure 40) section later in this document.

FADS Financial Aid Data Submission

The SICAS Financial Aid Data Submission (FADS) application extracts student data from Banner. The financial aid information, demographic features, and other characteristics collected are used to develop a large number of annual descriptive statistical summaries used by the Office of Institutional Research for planning, academic and fiscal administration, budget process, legal obligations, and reporting requirements imposed by State and Federal agencies. The FADS data will be combined with the data from the Student Revenue Data Submission (SRDS) to produce the Cost Calculator required by the Higher Education Opportunities Act (HEOA).

The SICAS Financial Aid Data Submission Report RORYFAD collects the FADS extract data, transmits the data to the data warehouse staging area, and produces a listing of data and transmission status. Any errors returned when RORYFAD is run in Transmit Mode are inserted into the SICAS Data Selection and Conversion Error Table SFRYERR and can be viewed on the Processing Errors tab of the SICAS Data Transfer Interface Control Form ROAYDTI. FADS must be configured before use. The Set Up section provides the information to configure the FADS application.

After Set Up is complete, run the RORYFAD process in Audit Mode. Review the results to determine that the Select Code is working properly. To view the Select Code, navigate to the Select Code tab, which is located on the lower portion of the Column Definitions tab on the SICAS Data Transfer Interface Control Form ROAYDTI.

The RORYFAD process must be run for <u>one aid year at a time</u>. Schools will be reporting in the fall, only for the prior Aid Year that has just concluded (i.e. Fall of 2011 – schools will be reporting the Aid Year 2010-2011).

<u>FADS Set Up</u>

1. Review Column Definitions on the Column Definitions tab of the ROAYDTI form

The FADS installation process attempts to extract and convert data for use in this application. The FADS column information should be reviewed for accuracy and completeness. For more information on the Column Definitions Tab see the SOAYDTI/ROAYDTI/TOAYDTI Form Details section "Column Definitions Tab."

2. Enter Primary and Secondary Contact Information on the Contact Information tab of the ROAYDTI form-Primary Contact information is required for FADS. Make certain Debug Messages and Active Interface is checked up on the right hand corner of the ROAYDTI form.

Primary Contact Information must be updated prior to using the FADS application. The Contact Information tab of the ROAYDTI form (Figure XX) allows the user to enter one Primary Contact (required) and multiple Secondary Contacts (optional). The Name, Job Title, E-Mail Address, Phone and User ID fields are required and must be completed. The Contact Address field is not in use at this time and does not need to be completed. For more information on the Contact Information tab see the SOAYDTI/ROAYDTI/TOAYDTI Form Details section "Contact Information Tab."

3. Set up the required Fund Categories on the Banner Fund Category Validation Form RTVFCAT

Step 1: The following Fund Categories must be set up on the Banner Fund Category Validation Form - RTVFCAT:

- EOP   EOP Category
- ESDHP   Empire State Diversity Honors Program
- NAGA   Native American Grant Aid (Federal)
- NAGS   Native American Grant Aid (State)
- SUSTA   SUSTA Category
- SUNY   State University of New York
- OTSTAT   State Other Than New York State
- UNREST   College or College Foundation – Unrestricted

To Set Up a Fund Category:

- Type in the Code for the Fund Category
- Type in a Description of the Fund Category
- Save

NOTE: If no Fund Category exists – no need to set up a category (i.e. – if a school doesn't have any SUNY Aid – then there is no need to set up a category).

Figure 13 RFVFCAT

Step 2: RFRBASE Banner Fund Base Data Form

The Fund Category must subsequently be linked to an appropriate <u>Associate Fund Code</u> on RFRBASE.

- Place the cursor on the Fund Code that you want to add a Fund Category for;
- Next block

Figure 14 RFRBASE Fund Base Data Form

Step 3: RFRBASE Banner Fund Base Data Form

- Click on Fund Category drop-down box
- Select a Fund Category
- The Description will populate based on the Fund Code the cursors was placed on(see previous step)
- Save

Figure 15 RFRBASE Fund Base Data Form - page two

Step 4: STVYTTY SICAS State Award Validation Form

If a school has a Winter Term the student registers for or aid was given for in an Aid Period, that Winter Term must be added on STVYTTY (starting with Winter 2006).

Figure 16 STVYTTY SICAS State Award Term Validation Form

4. Perform all required crosswalks on the Code Crosswalk tab of the ROAYDTI form including all data elements that have a Crosswalk Source of GTVYVAL

    For information on how to crosswalk elements on ROAYDTI, please see the SOAYDTI/ROAYDTI/TOAYDTI section below.

    Housing Plans - FC122:

    The Campus must coordinate the Institutional Housing Plans from the FAFSA with the actual housing budget and needs to crosswalk every budget code:
    1) On campus;
    2) Off-campus with family;
    3) Off-campus without family; and
    4) Off-campus unknown

    Cost of Attendance (COA) Components - FC130 to FC190:

    COA Components *must* be set up in 2011-2012 and going forward. RTVCOMP/ COMP cannot be set up if the school budget components are NOT separating out TUITION from FEES and all COA components are not broken up as listed below. THIS IS KEY to being able to set up crosswalks.

    School must identify the cost of attendance components as broken out in the following manner:

1.)	'TUITION'	Tuition	(FC130)
2.)	'FEE'	Student Fees	(FC140)
3.)	'ROOM/BOARD'	Room and Board	(FC150)
4.)	'BOOKS'	Books and Supplies	(FC160)
5.)	'PERSONAL'	Personal Expenses	(FC170)
6.)	'TRANSPORT'	Transportation	(FC180)
7.)	'OTHER'	Other Expenses	(FC190)

Award and Resource Program - FA010:
Aid (not resources) is selected and converted using select code in the background. Awards that are not on the list below must be cross walked. All Resources must be cross walked on the cross walk tab of ROAYDTI.

School must identify the program for the appropriate award or resource:

01 Pell Grant	12 TEACH	24 FFEL Unsub
02 SMART	17 Perkins	25 Direct Unsub
03 FSEOG	18 Nursing	27 FFEL PLUS
04 ACG	19 health Professions	28 Direct PLUS
05 TAP	20 FWS	29 FFEL Graduate Plus
07 EOP	21 SUSTA	30 Direct Graduate PLUS
08 APTS	22 ESDHP	31 Alternative Loans
10 Native American (Federal)	23 FFEL Sub	
11 Native American (State)		

For Veterans Awards Program Codes 35- 60
Crosswalk payment detail codes using the TBVYPCD Validation table and AWD Prog6 Interface.

Other Award or Resource Source - FA030:
If the Award Program is other (999) the source or agency of the financial award must be reported. This element is selected and converted using select code in the background. No crosswalk is necessary. If no Aid source can be found, 009 will be sent.

The Campus must identify the source of the appropriate award or resource (7 SUNY no longer is valid):

1. Federal
2. State – New York State
3. Institutional – Restricted
4. Institutional – Unrestricted
5. Private – Non-Federal/Non-State
6. Other State – State other than New York
8. Other
9. Unknown

Basis of Awards or Resources - FA040: This element MUST be cross walked on the Code Crosswalk tab of ROAYDTI

School must identify the basis of the appropriate award or resource:

Athletic	Athletic, Merit, Need, Other	Group, Other,
Athletic, Group	Athletic, Need	Group, Need, Other
Athletic, Group, Merit	Athletic, Need, Other	Merit
Athletic, Group, Merit, Need Other	Athletic, Other	Merit, Need
Athletic, Group, Merit, Other	Group	Merit, Need, Other
Athletic, Group, Need	Group, Merit	Merit, Other
Athletic, Group, Need, Other	Group, Merit, Need	Need
Athletic, Group Other	Group, Merit, Need, Other	Need, Other
Athletic, Merit	Group, Merit, Other	Other
Athletic, Merit, Need	Group, Need	Unknown

Other Types of Resources - FA050:
AID (not RESOURCES) is selected and converted using select code in the background UNLESS the fund is designated as other (OTHR) on RFRBASE. The designation on Funds marked other must be changed to something else (i.e. EXTN-External).

School must identify types of the appropriate resource:

1. Grant or Scholarship
2. Work Study
3. Loan
4. Waiver
5. Unknown

Summary: If an award is listed in element FA 010 (01-31) it only needs to have the Basis cross walked. If an award is not listed it needs to have the Source and Basis cross walked. All Resources must be cross walked for Source, Basis and Type.

6. Link the Fund Categories set up on RTVFCAT (Figure 13) to appropriate Associate Fund Code on RFRBASE

7. If a campus has a Winter Term the student registers for or aid was given for in an Aid Year, that Winter term must be added on STVYTTY (Figure 16) (starting with winter 2006).

SRDS Student Revenue Data Submission

The SICAS Student Revenue Data Submission (SRDS) application extracts student data from Banner. The student revenue information, demographic features, and other characteristics collected are used to develop a large number of annual descriptive statistical summaries used by the Office of Institutional Research for planning, academic and fiscal administration, budget process, legal obligations, and reporting requirements imposed by State and Federal agencies. The SRDS data will be combined with the data from the Financial Aid Data Submission (FADS) to produce the Cost Calculator required by the Higher Education Opportunities Act (HEOA).

The SICAS Student Revenue Data Submission Report TSRYSRD collects the SRDS extract data, transmits the data to the data warehouse staging area, and produces a listing of data and transmission status. Any errors returned when TSRYSRD is run in Transmit Mode are inserted into the SICAS Data Selection and Conversion Error Table SFRYERR and can be viewed on the Processing Errors tab of the SICAS Data Transfer Interface Control Form TOAYDTI. SRDS must be configured before use. The Set Up section provides the information to configure the SRDS application.

After Set Up is complete, run the TSRYSRD process in Audit Mode. Review the results to determine that the Select Code is working properly. To view the Select Code, navigate to the Select Code tab, which is located on the lower portion of the Column Definitions tab on the SICAS Data Transfer Interface Control Form TOAYDTI.

The TSRYSRD process is run twice a year and must be run for one submission period at a time.

SRDS Set Up

1. Review Column Definitions on the Column Definitions tab of the TOAYDTI Form.

The SRDS installation process attempts to extract and convert data for use in this application. The SRDS column information should be reviewed for accuracy and completeness. For more information on the Column Definitions Tab see the SOAYDTI/ROAYDTI/TOAYDTI Form Details section "Column Definitions Tab."

2. Enter Primary and Secondary Contact Information on the Contact Information tab of the TOAYDTI Form. Primary contact information is required for SRDS.

   Primary Contact Information must be updated prior to using the TSDS application. The contact Information tab of the SOAYDTI form (Figure XX) allows the user to enter one Primary Contact (required) and multiple Secondary Contacts (optional). The Name, job title, E-Mail Address, Phone and User ID fields are required and must be completed. The Contact Address field is not in use at this time and does not need to be completed. For more information on the Contact Information tab see the SOAYDTI/ROAYDTI/TOAYDTI Form Details section "Contact Information Tab."

3. Perform all required crosswalks on the Code Crosswalk tab of the TOAYDTI Form including all data elements that have a Crosswalk Source of GTVYVAL.

   Only those Detail Codes that appear as charges and are mapped to approved URAS codes will need be cross walked. Community Colleges do not use the URAS system and do not have to crosswalk detail codes to URAS code.

   For State Operated Mode (SIRS Interface, Campus Type Validation Code = SO)

   - Charge Detail Codes mapped to accounts in the 3000-4999 range will be included.
   - Charge Detail Codes mapped to accounts in the 2400-2499 range will be included.
   - All displayed Detail Codes probably need to be cross walked – map exceptions to "NA" – This Revenue Item is not within the URAS Accounts to Report.

   For State Operated – Not using Finance/URAS Accounting Mode (SIRS Interface, Campus Type Validation Code = SN)

   - Charge detail codes that are not REFUND Codes (REFUND_IND of Y, M or P) will be included.
   - REFUNDABLE Detail Codes will be included.
   - Campus will have to implement a new FZ_SRDS_URAS_Account local mod point so that they can find the URAS Account codes in some other local manner.

   For Community College Mode ("SIRS Interface, Campus Type Validation Code = CC)

   - Only Charge Detail Codes that are not marked as Refundable will be included.
   - Any Charge Detail Code that is not correctly mapped to a Revenue Category should be mapped to "NA" – this Revenue Item is not within the URAS Accounts to Report.
   - "NA" mapped Charges will not be extracted and sent to System Administration.

It is required that you explicitly map Detail Codes that should not be used to "NA" – this Revenue Item is not within the URAS Accounts to Report. The process is using that code to specifically block detail codes in certain categories from triggering inclusion of the student in the submission; therefore, it is required that campuses explicitly indicate the "NA" detail codes.

The Detail Code categories TUI, FEE, HOU, and MEA will force a student into the population, but Detail Codes marked "NA" (even when in those categories) will not. So, if a student has a

FEE used to charge for Driver's Ed, which is marked "NA" and no other TUI, FEE, HOU or MEA codes on their account and does not appear on SDS for that period, then the FEE is not reported, nor is the Parking Permit charge which is mapped, because the student is never included in the population. Codes which are left unmapped (i.e., they contain "*") will continue to force the student into the population and generate errors.

NOTE: Revenue charges that are mapped to a URAS code not on the approved list should be mapped to NA – This Revenue Item is not within the URAS Accounts to Report. The interface code NA has been developed to be used for those revenue categories that should not be reported by this process. These might include detail codes for items such as returned check fees, manual refunds that are listed as a charge.

NOTE: Course fee detail codes that are NOT attached to a course section (CRN) should not be cross walked.

Special note concerning high school students:

High school students fall into at least two categories:

a. On-campus students – charged full-tuition and treated the same as other students on the campus (i.e., competing for seats in sections, etc.) These students are reported the same as other students in all SIRIS submissions for both state-operated campuses and community colleges.

b. Off-campus students (College in High School) – charged less than full tuition. Full tuition only should be coded as revenue category 001. Less that than full tuition should be revenue category 043. State – operated campuses would include on SRDS but no SDS or TSDS. Community colleges would include on SDS and TSDS and would report using revenue category 042 for less than full tuition.

   State Op students can and should be included only if they are paying full SUNY tuition for the course. Community colleges adopted a pattern to report college in the high school students in their SIRIS submissions.

   State-operated campuses fall under the policy of including high school students in the student submission only if the students are paying full tuition for the course taught on the campus. Students taking college courses in the high schools are not reported in student submissions.

   All revenue associated with these students should be reported on SRDS, regardless of whether or not they appear on the student submission.

   For information on how to crosswalk elements on TOAYDTI, please see the SOAYDTI/ROAYDTI/TOAYDTI section below.

DADS Degree Awarded Data Submission

The SICAS Degree Awarded Data Submission (DADS) application extracts student data from Banner. The degrees awarded information, demographic features, and other characteristics collected are used to develop a large number of annual descriptive statistical summaries used by the Office of Institutional Research for planning, academic and fiscal administration, budget process, legal obligations, and reporting requirements imposed by State and Federal agencies.

The SICAS Degree Awarded Data Submission Report SHRYDAD collects the DADS extract data, transmits the data to the data warehouse staging area, and produces a listing of data and transmission status. Any errors returned when SHRYDAD is run in Transmit Mode are inserted

into the SICAS Data Selection and Conversion Error Table SFRYERR and can be viewed on the Processing Errors tab of the SICAS Data Transfer Interface Control Form SOAYDTI. DADS must be configured before use. The Set Up section provides the information to configure the DADS application.

After Set Up is complete, run the SHRYDAD process in Audit Mode. Review the results to determine that the Select Code is working properly. To view the Select Code, navigate to the Select Code tab, which is located on the lower portion of the Column Definitions tab on the SICAS Data Transfer Interface Control Form SOAYDTI.

The SHRYDAD process is run twice a year and must be run for one submission period at a time.

DADS Set Up Overview

DADS Set Up

1. Review Column Definitions on the Column Definitions tab of the SOAYDTI Form.

    The DADS installation process attempts to extract and convert data for use in this application. The TSDS column information should be reviewed for accuracy and completeness. For more information on the column Definitions Tab see the SOAYDTI/ROAYDTI/TOAYDTI Form Details section "Column Definitions Tab."

2. Enter Primary and Secondary Contact Information on the Contact Information tab of the TOAYDTI Form. Primary Contact information is required for DADS.

    Primary Contact information must be updated prior to using the SRDS application. The Contact Information tab of the TOAYDTI form (Figure XX) allows the user to enter one Primary Contact (required) and multiple Secondary Contacts (optional). The Name, Job Title, E-Mail Address, Phone, and User Id fields are required and must be completed. The Contact Address field is not in use at this time and does not need to be completed. For more information on the Contact Information tab see the SOAYDTI/ROAYDTI/TOAYDTI Form Details section "Contact Information Tab."

3. Perform all required crosswalks on the Code Crosswalk tab of the SOAYDTI Form including all data elements that have a Crosswalk Source of GTVYVAL.

    All Data Elements in the Column Name field on the Column Definitions tab of the SICAS Data Transfer interface Control Form SOAYDTI that have a Crosswalk Source of GTVYVAL must be cross walked to System Administration values on the code Crosswalk tab of the SOAYDTI (Figure XX) form. The System Administration values are defined on the Validation Codes tab of the SOAYDTI form.

    Only those Award Codes, Award Level Codes, Hispanic Origin Codes, and Sub-Campus Codes used by the campus must be cross walked.

    For information on how to crosswalk elements on SOAYDTI, please see the SOAYDTI/ROAYDTI/TOAYDTI Form Details section "Code Crosswalk Tab."

SOAYDTI/ROAYDTI/TOAYDTI SICAS Data Transfer Interface Control Form

Course Data, Student Data, Term/Section Data and Degree Awarded SIRIS Submissions are configured and reviewed using the SICAS Data Transfer Interface Control Form SOAYDTI.

The Financial Aid Data Submission is configured and reviewed using the SICAS Data Transfer Interface Control Form ROAYDTI.

The Student Revenue Data Submission is configured and reviewed using the SICAS Data Transfer Interface Control Form TOAYDTI.

SOAYDTI, ROAYDTI and TOAYDTI are GORYDTI displayed with a different name. Different form names are used so that access can be restricted to the appropriate processes when appropriate.

The DTI form is used to review and configure the columns being extracted (Columns Definitions Tab), maintain a 'virtual validation table' for each column (Validation Codes Tab), provide a place to enter all required crosswalks (Code Crosswalk Tab), hold primary and secondary contact information for the person or persons responsible for the data submission (Contact Information Tab), review data collected during processing (Data Review Tab), and display errors and warnings returned during data transmission (Processing Errors Tab).

Tab-Level Security

The DTI form is designed to use Banner Tab-Level Security to control user access to each individual tab on the form. With Tab-Level Security, if a user does not have access, the tab will not be visible; if a user is granted Query Access, they can view and query the data on the tab but not change it; if a user has Full Maintenance Access to the tab, they have edit capabilities.

If a security class has permissions for a form, and if tab-level security is not setup, the members of that class will have access to all of the form's tabs at that security level. Security access must be granted to each tab to create restrictions for individual tabs. Tab access is always limited by a user's level of access to the form containing the tab. For example, if a user has query access to the form, then the user will have query only access to the tab even if given full access to a tab. A tab may never provide more access than the form that contains it.

Before assigning any Security access, the DBA must understand that when the user calls SOAYDTI, ROAYDTI or TOAYDTI, it is really GORYDTI that is displayed with a different name. The object that a user receives security access for is GORYDTI. SOAYDTI/ROAYDTI/TOAYDTI only require form access. Any form or tab-level access granted on GORYDTI is reflected on the DTI forms when the user calls it. If access is not granted on GORYDTI, the user will not be allowed to view data on the DTI form.

To enforce Tab-Level Security for the DTI forms, the DBA should assign Maintenance Access for SOAYDTI/ROAYDTI/TOAYDTI to all users that require access to the form. Then, assign Query or Maintenance Access to GORYDTI. Once this is done, assign individual Tab-Level access for GORYDTI. Remember that a tab may never provide more access than the form that contains it. If Query Access is assigned for GORYDTI, then the user can only be granted Query Access (or no access) to the tabs on GORYDTI.

Maintenance Access to the Column Definitions tab should be limited to individuals responsible for maintaining the interface. This is usually IT personnel or IR personnel. The Validation Codes tab is setup by SICAS and should not be modified by individual campuses except when directed by

SICAS or SUNY System Administration, therefore Maintenance Access to this tab should be granted to individuals responsible for modifying the Validation Codes. Maintenance Access to the Code Crosswalk tab and the Contact Information tab is granted to individuals responsible for maintaining the interface. The Data Review tab and the Processing Errors tab have no maintenance functionally, therefore Query Access or Maintenance Access provide the same level of data access.

For more information on assigning Banner Tab-Level Security, refer to 'Section 1 Tab-Level Security – Technical' in the Banner General Release Guide, September 2007-Release 7.5. This document is available on the SICAS Center's website.

Web Services/Proxy Client/Certificate

A SICAS process extracts data for IR submissions from Banner and converts it into an XML document specifically formatted for SUNY's Data Warehouse. A Web Service accepts this XML and transmits it to the Data Warehouse staging area using a Proxy Client. System Administration's application performs edit checks and returns a response XML document with the data it received and message entries describing any problems it encountered. These errors and warnings are available immediately for review on SOAYDTI/ROAYDTI/TOAYDTI's Processing Errors Tab.

The Web Service does not produce a local file; therefore Personally Identifiable Information (PII) is not placed in readable files although any reports generated are stored locally and must be protected. Using Web Services ensures data synchronization between Banner and System Administration's Data Warehouse as long as all data corrections are performed in Banner and not on System Administration's website.

The Web Service takes between 1.25 and 1.50 seconds to execute therefore the theoretical maximum transmission is between 40 and 48 records per minute. Transfer rates of 35 records per minute are more common. The Web Service Call start- and end-time values are available in the SICAS Web Service XML Storage Table GORYXML.

Proxy Client/Certificate

The Proxy Client is a Java applet developed by System Administration that provides message encryption, digital signing, and HTTPS protocol for IR data submissions using Web Services. The Proxy Client uses SSL (Secure Sockets Layer) and WS-Security to provide a set of mechanisms for Web Services to secure the data transmitted in the SOAP message exchanges. A digital certificate and private key is used to create the WS-Security signature on SOAP requests. There is more information about the SOAP envelope later in this document in the Web Service Configuration Tab section, found under the main heading of the DTI Form Details, Column Definitions Tab—Web Service Configuration Tab, and the Web Service Request and Web Service Response Tabs, found in the lower section of the Processing Errors Tab. The Web Service Request and Web Service Response tabs only display if information is available for viewing.

Since Course Data Submission (CDS), Student Data Submission (SDS), Term/Section Data Submission (TSDS), Degree Awarded (DADS), Student Revenue (SRDS) and Financial Aid Data Submission (FADS) are transmitted using Web Services only, the Proxy Client and certificate must be installed and working before these applications can be used. The proxy client is required for Audit runs also because communication does take place between the campus and System Admin, therefore the proxy client must be properly configured and running before running any of the SIRIS processes in any mode.

Alter code within the Local Campus Modification Function FZ_PROXY_IR to determine the local address of the Proxy Address. The Proxy Client software and a text file called README.TXT that contains installation details are located in a ZIP file included in the installation package. A copy of README.TXT and a copy of the Local Campus Modification Function FZ_PROXY_IR are available in Appendix C of this manual.

Changes to the proxy client configuration for the FADS submissions.

 All future SIRIS submissions will use the FADS mechanism rather than the original one and as the second versions of CDS, TSDS and SDS are released they too will adopt the new method of transmitting.

The differences are:

• CDS, SDS and TSDS generate all XML within the DB and then use HTTP facilities in the Database to transmit the XML and process the results.

• Due to limitations of Oracle XML and HTTP libraries we have moved that task for data transactions to JAVA (but not the Java built into Oracle as it too is limited).

• The utility calls for checking status of submissions and Initializing and Finalizing batches continue to work the OLD way in the DB since Status Calls happen whether or not transmission is requested and we chose not to rewrite working INITIALIZATION and FINALIZATION routines that don't suffer from the limitations Oracle libraries present?

• The submission transaction calls now happen in a Java process running on the server that hosts Job Submission—XML generation and HTTP transmission. The minimum Java JVM version is 1.5 (SE or EE).

• Therefore, if a campus doesn't host Job Submission on the DB server then calls will come from TWO sources and TWO IP addresses must be configured—the Proxy Client only accepts ONE configuration for this Client Restriction parameter as a regular expression so a somewhat complicated config string is required.

• This only impacts FADS today but within a few months FADS, SRDS and DADS will all use this method.

• SDS has already been altered to work this way as an OPTION at SICAS but we have decided to release that change later.

• TSDS would greatly benefit from this change and will likely be the next interface to move to the Java transmission method.

The new handler.xml restrict client syntax is: (Note this is an EXAMPLE only)

Value="^141\.11\.11\.((51)|(61))$"

Which would securely allow for IP Addresses calling the Proxy Client:

141.11.11.51 Database Server
141.11.11.61 Code Tree / Job Submission App Server

Information on obtaining the digital certificate required for accessing secured Web Services at SUNY System Administration is available in Appendix D of this manual.

For additional information on the Proxy Client or the digital certificate, email the System Administration Security team at secteam@sysadm.suny.edu.

Local Record Modifications

The DTI forms were designed to allow campuses to make local modifications that will not be overwritten when an upgrade is installed. If a campus does not store data in a field that SICAS is using in the collection process, the campus can make a local mod on this form to collect the data from the appropriate field. A local copy of the record is created by inserting a new record and duplicating the SICAS record and altering it to meet the needs of an individual campus. Alterations can be made on the select code tab. Do not delete the SICAS record. Local Records can be modified in any way. These modifications are preserved when a new release is installed.

Campuses that create local versions of the elements need to be aware of changes that SICAS makes. If SICAS produces a patch or a new version of an element the local version will not be impacted. If the SICAS patch makes a change to how an element is collected, the campus may wish to switch back to using the SICAS version. To switch back to the SICAS version simply change the processing method on the local record to "Use SICAS."

There are also several elements that may or may not be required for your institution. For example the building and room are not required for Community Colleges and these elements need to have the processing method changed to "force to null." The instructor information is required for Community Colleges and the process method needs to be changed to "Use Select Code." This is done by creating a local modification and changing the process method.

 To create a Local Modification:

- Login with Banner Maintenance Access to the Column Definitions Tab on the DTI form
- Select the record that must be altered
- Duplicate the record—Pick 'Record' from the menu Select 'Insert' and then Select 'Duplicate'
- Alter the new record as needed—the SICAS record is now grey indicating that it is inactive and the active local record is black
- The 'Local Mod' indicator box is automatically checked

This example uses the CDS interface. To create a Local Mod for the 'Teach Cert' data element:

- Create a local copy of the 'Teach Cert' data element

    o Highlight the SICAS element 'Teach Cert'
    o Duplicate the record (described above)

- Set the Processing Method for the new local record to 'Send the Default'—this will send all records as non-Teacher Certification courses (the Default Value of '4' indicates a non-Teacher Certification Course)

Or

- Set the Processing Method to 'Use Select Code' and change the SQL Select Code to identify Teacher Certification Courses (the Select Code Tab is located in the lower portion of the Column Definitions Tab)

Figure 17 SOAYDTI Column Definitions Tab

Masking Personally Identifiable Information (PII)

Occasionally during the course of resolving errors in the SIRIS processes it is necessary for a campus to extract the XML that was sent and received from the Processing Errors tab. This XML contains PII which must be masked. In most cases, PII is not needed to determine the cause of an error and SICAS employees are not permitted to accept PII from campuses. Also, email is an unsecure form of communication and sending PII via email is not advisable.

The Mask PII feature on the Processing Errors tab will mask data based on the rules established in the PII Masks element. SICAS has shipped a set of default settings for the Masking element. Each campus should review the default setting to determine if they meet local security requirements.

When reviewing the PII Masks element a "~" indicates that data will be sent and a "?" indicates data that will be masked. For example if Student Last Name is ~???????????, the first initial of the students last name will be sent and the rest will be masked when the XML is extracted.

Figure 18 illustrates the masking defaults.

Figure 18 The DTI form PII Masks element.

SOAYDTI/ROAYDTI/TOAYDTI FORM DETAILS

This section provides details about the tabs and fields on the SICAS Data Transfer Interface Control Form SOAYDTI/ROAYDTI/TOAYDTI. Three versions of this form have been created so that access can be granted separately for to the Financial Aid Data Submission (FADS), Student Revenue Data Submission (SRDS) and the Course (CDS), Student (SDS), Term Section (TSDS) and Degree Awarded (DADS) submissions. It is possible to place the FADS and SRDS

interfaces on the SOAYDTI form so that one person(s) has access to all the submissions. This can be done by running the istvyinf_rs.sql script.

Figure 19 The DTI Form – Column Definitions Tab shown for the CDS interface.

Key Block

Interface Type – Select the desired application.

- SIRS for SIRIS IR Submission Support
- CDS for Course Data Submission
- SDS for Student Data Submission
- TSDS for Term/Section Data Submission
- FADS for Financial Aid Data Submission
- SRDS for Student Revenue Data Submission
- DADS for Degree Awarded Data Submission

Active Interface – This controls whether or not the application is active—SIRS, CDS, SDS,TSDS, FADS, SRDS and DADS must always be active.

Debug Messages – This controls whether or not debug information displays in the SICAS Data Selection and Conversion Error Table SFRYERR—debug information can be viewed on the Processing Errors Tab. The Debug Messages box should not be checked for normal processing runs, but it is often used during initial runs or to troubleshoot problems. The information provided from 'Debug Messages' is useful in diagnosing processing failures and performance problems.

Column Definitions Tab
The Column Definitions Tab defines all of the data elements used for the IR Submission interface selected in the Key Block. Elements that are black are active and elements that are grey are inactive (see the Local Record Modifications section). The Process Method column indicates what method is being used to process the element including 'Use Select Code', if the Select Code associated with that element is in use.

Column Name – The Name of the column (data element). It is possible to damage or even cripple the IR submission process by improperly changing data elements stored on the DTI form, therefore column names should not be changed.

Process Order – The Order the Column Names are processed; Columns are listed in this order.

Process Method – The Method used to process the Column.
    Valid Entries: Use Select Code
                      Send the Default
                      Force to NULL
                      Existing Value
                      INACTIVE
                      Use SICAS - if a local record exists, it is ignored and the SICAS record is used

Default Value – The Value assigned when the Select Code does not return data (No Data Found) or when the Process Method is set to 'Send the Default'.

Exception Value – The Exception Value placed in the column when data errors other than NO DATA FOUND are encountered during processing.

Local Mod – Indicates a Local Modification override of SICAS provided data element records. SICAS will not modify Local records. If present, Local records are processed instead of SICAS records unless the Processing Method is set to 'Use SICAS'.

Banner Source – The name of the Banner Table providing data or validation for the Column. If the data is selected using a function, this is the primary table used by that function to obtain a return value.

Description – Banner Source Description — a description of the Banner Source Table.

Crosswalk Source – How the data element value is derived. Possible Crosswalk Sources include:

  The Name of the Validation Table/Form that contains the data element value used to convert Banner codes.

  Direct – The Value is extracted directly from Banner without validation conversion.

  GTVYVAL – The Value is from the SICAS General Interface Validation Table GTVYVAL which stores the System Administration Crosswalk values and is maintained on the Validation Codes Tab. Use the Code Crosswalk tab to crosswalk all Data Elements with a Crosswalk Source of GTVYVAL.

  Calculated – The value is calculated from a number of different fields.

  Constant – The value is converted to a constant value.

Description – The Crosswalk Source Description - a Description of the Validation Table or method used.

Column Details Block

The Column Details Block is located in the lower portion of Column Definitions Tab. The information displayed here pertains only to the active record.

Target Column – The name of the Column within the Target Table the active record updates.

Target Table – The name of the extract table the active record updates.

In CDS, there are two extract tables:

SICAS Base Course Data Extract Table SOBYCDS – Base Course Data
SICAS Equivalent Course Extract Table SORYCDS – Equivalent Course Data

In SDS, there are two extract tables:

SICAS Student Submission Base Table SOBYSDS – Base Student Data
SICAS Student Submission Repeating Data Table SORYSDS – Repeating Data for
   Admissions/Placement Criteria, Disability Status, and Race Codes

In TSDS, there are several extract tables:

SICAS Section Data Submission Table SOBYSEC – Base Section Data
SICAS Section Data Submission Meeting Table SORYMET – Section Meetings Data
SICAS Section Data Submission Meeting Table SORYMST – Section Meetings Data for
   Sections meeting at the same time
SICAS Section Student Base Table SOBYSTU – Student Data
SICAS Section Student Repeating Table SORYSTU – Student Data
SICAS Section Faculty Base Table SOBYINS – Instructor Data
SICAS Section Faculty Repeating Effort Table SORYINS – Instructor Data

In FADS, there are two extract tables:
SICAS Financial Aid Data Submission Table ROBYFAD – Financial Aid Applicant Data
SICAS Financial Aid Award Table RORYAWD – Financial Aid Award and Resource Data

In SRDS, there are two extract tables:
SICAS Student Revenue Data Submission Student Base Table TOBYSRD – Student
   Revenue Data
SICAS Student Revenue Submission Repeating Data Table TORYSRD – Student Data

In DADS there are three extract tables:
SICAS Student Submission Repeating Data Table SORYDEG
SICAS Degree Awarded Submission Student Base Table – SOBYDAD
SICAS SUNY Program Validation (APES & APIS) - STVYAPE

Type – The Type is used to separate elements into groups. Each group is called at a different time by the processing routine.

CDS has two types. All BASE elements are processed before RPTG (repeating) elements.

BASE   updates SOBYCDS – the SICAS Base Course Data Extract Table
RPTG   updates SORYCDS – the SICAS Equivalent Course Extract Table

SDS has five types. All Base elements are processed before Repeating elements.

BASE   updates SOBYSDS – the SICAS Student Submission Base Table
RACE   updates SORYSDS – the SICAS Student Submission Repeating Data Table
DISB   updates SORYSDS – the SICAS Student Submission Repeating Data Table
ADMT   updates SORYSDS – the SICAS Student Submission Repeating Data Table

TSDS has six types. They are processed in the order listed:

SECT   updates SOBYSEC – the SICAS Section Data Submission Table
MEET   updates SORYMET and SORYMST – the SICAS Section Data Submission
            Meeting Tables
BINS   updates SOBYINS – the SICAS Section Faculty Base Table
RINS   updates SORYINS – the SICAS Section Faculty Repeating Effort Table
BSTU   updates SOBYSTU – the SICAS Section Student Base Table
RSTU   updates SORYSTU – the SICAS Section Student Repeating Table

FADS has XX types. They are processed in the order listed:

BASE   updates ROBYFAD – the SICAS Financial Aid Data Submission Table
ATRM   updates RORYAWD – the SICAS Financial Aid Awards Repeating Table
ARSC   updates RORYAWD – the SICAS Financial Aid Awards Repeating Table

SRDS has two types. They are processed in the order listed:

BASE   updates TOBYSRD – the SICAS Student Revenue Data Submission Student
            Base Table
CHRG   updates TORYSRD – the SICAS Student Revenue Submission Repeating Data
            Table

DADS has three types. They are process in the order listed:

BASE   updates SOBYDAD – the SICAS Degree Awarded Submission Student Base
            Table
RACE   updates SORYDAD – the SICAS Student Submission Repeating Data Table
DEGR   updates SORYDEG - the SICAS Student Submission Repeating Data Table

Exception Value - The Exception Value placed in the column when data errors other than NO
        DATA FOUND are encountered during processing.

User – The User who created or last altered the active record.

Activity Date – The Date the active record was created or last updated.

Element Details Tab

The Element Details Tab is located in the lower portion of the Column Definitions Tab of the
SOAYDTI/ROAYDTI/TOAYDTI forms. This section contains a detailed description of the active
record, its source within Banner, and possible values or potential exceptions. The element
definitions and formatting information are from the SUNY Data Dictionary developed by the Office
of Institutional Research. A current copy of the data dictionaries for all SIRIS applications is
available on the SUNY/ITEC Confluence website http://confluence.itec.suny.edu.

Figure 20 The DTI Form – Element Details Tab in lower portion of Column Definitions Tab

Select Code Tab

The Select Code Tab is located in the lower portion of the Column Definitions Tab of the SOAYDTI/ROAYDTI/TOAYDTI form. This section contains the SQL Select Code used to populate the column when the 'Use Select Code' Process Method is selected.

Figure 21 The DTI Form – Select Code Tab in lower portion of Column Definitions Tab

Update Condition Tab

The update condition allows the campus to set a specific rule for updating the value of the element. Figure 22 shows an example of an update condition for the Visa Code element. In this example, the visa code will only be updated if there is no data present.

Figure 22 The DTI Form - Update Condition Tab

Web Service Settings Tab

The Web Service Settings Tab is located in the lower portion of the Column Definitions Tab of the SOAYDTI/ROAYDTI/TOAYDTI forms. This section contains information pertaining to the Web Service call that sends data to System Administration. There should be no need to alter data on this tab with the possible exception of changing the 'Web Service Return Processing' value. This value can be adjusted to eliminate or require confirmation. Additional fields should not be set to replace local data with System Administration returned values. Regardless of this setting, the SUNY ID is the only value moved from the staging area back into the Banner tables.

The Term/Section Data Submission XML documents are too large to be returned for data validation. Changing the 'Web Service Return Processing' value will not alter the validation since System Administration is not returning the data. Data should be reviewed on the System Administration website to determine if it was transferred accurately. Data will be returned if errors are encountered.

Figure 23 The DTI Form – Web Service Settings Tab in lower portion of Column

XML Tag Definitions – This field contains any XML tag (snippets from the hierarchy) that will be updated using the active column's data for transmission. The SOAP Envelope used for transmission could contain any or all of these snippets and each should be updated with the column's value. The data location being updated is marked with a '?' and the snippets are delimited with '*'. These values are determined by the WSDL file used to specify the Web Service and should not be altered locally. Updates to these values will be shipped with SICAS releases when new WSDLs are published by System Administration.

Alternate XML Column Specification – This field allows the data to undergo reformatting or conversion for use in XML. For example, a date can be reformatted from a Banner format to a System Administration format. In some cases, the desired value for an extract file is not the same as that desired for the SOAP Envelope XML. The Alternate XML Column Specification allows the extract tables to contain the file version but the XML transmission will contain the desired XML value. Use of TRANSLATE NVL or DECODE is most common in this field. Use of the Target Column is expected but any column in the Target staging table is available during the conversion. Alternate XML Column Specification should not need to be altered but could be used to improve format match of Sent and Returned Data.

Web Service Return Processing – Web Service Return Processing is set by SICAS and should not be altered locally. When the Web Service transmits data, it will return an XML Response SOAP Envelope. The content of this envelope is the data sent (as understood by the receiving server) and any error messages related to processing the data. The error messages are always inserted into the SICAS Data Selection and Conversion Error Table SFRYERR and can be viewed on the Processing Errors Tab of the DTI forms, but the returned data can be handled in the following ways:

Confirm Returned Value – This option compares the sent and returned values and inserts an Error Record into the SICAS Data Selection and Conversion Error Table SFRYERR, if they do not match. This assures the correct information is received and interpreted correctly by the server.

Due to the size of the SDS and TSDS submission files, most data are not returned and   therefore cannot be confirmed.

Update NULL Local Value – This option copies information back into the extract column, if that extract column is NULL. If the extract column is NOT NULL, the values are compared (see Confirm Returned Value above).

CDS uses this option to capture SUNY IDs assigned to new courses. A routine at the end of processing moves these values back into the shadow table SICAS created to maintain CDS related data. This SUNY Course ID is displayed on the Basic Course Information Form SCACRSE.

SDS uses this option to capture SUNY IDs assigned to new students. A routine at the end of processing moves these values back into the SICAS SUNY ID Repeating Table SPRYSID.

TSDS, FADS and SRDS do not return any data intended to update local data. This option should not be used in these interfaces.

Update Local Value – This option copies information back into the extract column. A routine at the end of processing can then move these values back into the shadow table SICAS created. It is not useful to add this setting to any of the columns; currently only the SUNY IDs support copy back into Banner.

This option is not in use with CDS, SDS,TSDS, FADS or SRDS.

No Action on Return – This option ignores any returned value. This allows System Administration to return nothing for a specific field without inserting an error. This is used for values System Administration does not wish to return like Social Security Numbers, since each transfer is a measurable security risk.

This option is not in use with CDS. Most elements in SDS,TSDS, FADS and SRDS use this option.

No Return is Allowed – This option indicates that a returned value is considered an error and a record should be inserted into the SICAS Data Selection and Conversion Error Table SFRYERR.

This option is not in use with CDS.

SDS, TSDS, FADS and SRDS use this option for Social Security Numbers submitted to System Administration. This option limits the transfer of sensitive personal data since each transfer is a measurable security risk.

Audit Trail Tab

The Audit Trail tab was added to document when changes are made to elements. SICAS will enter information in the audit trail when changes to an element are shipped. It is recommended that campuses use the audit trail with as much detail as possible when creating local versions of elements. This will eliminate confusion in the future when attempting to debug problems.
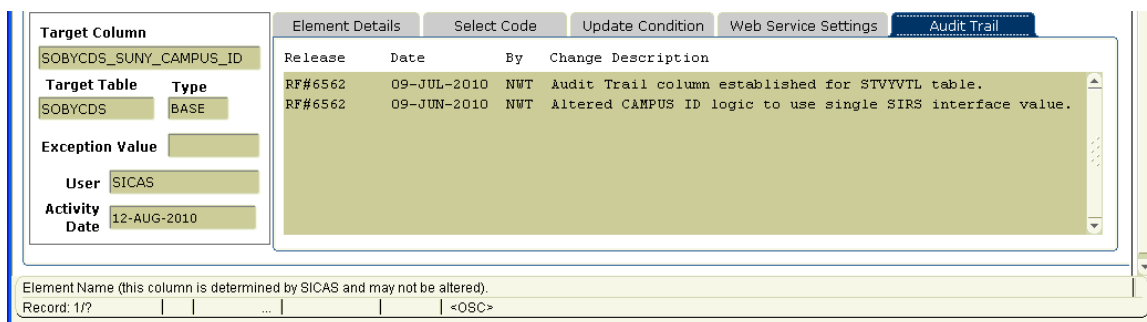
Figure 24 The DTI Form – Audit Trail Tab in lower portion of Column Definitions Tab

Validation Codes Tab

The Validation Codes Tab contains Validation Codes determined by System Administration. This tab allows the application to maintain a 'virtual validation table' for each column. Only columns

that require crosswalks need to be entered here. Other columns may have validations setup here to support descriptions on the Data Review tab; this is helpful for reporting but is not required. Each record is related to a single column from the Column Definitions Tab. This page should be used for 'one time' validations related to a single or a small number of columns with relatively few values. Other validation tables and views can be used when a single validation table will convert many columns.

SICAS delivers the Validation Code values and no edits should be necessary. If System Administration adds a value to an element or if a value is missing, the new value may be inserted on this tab by inserting a new record and adding the correct information in. The new values must match what System Administration is expecting or an error will be returned.
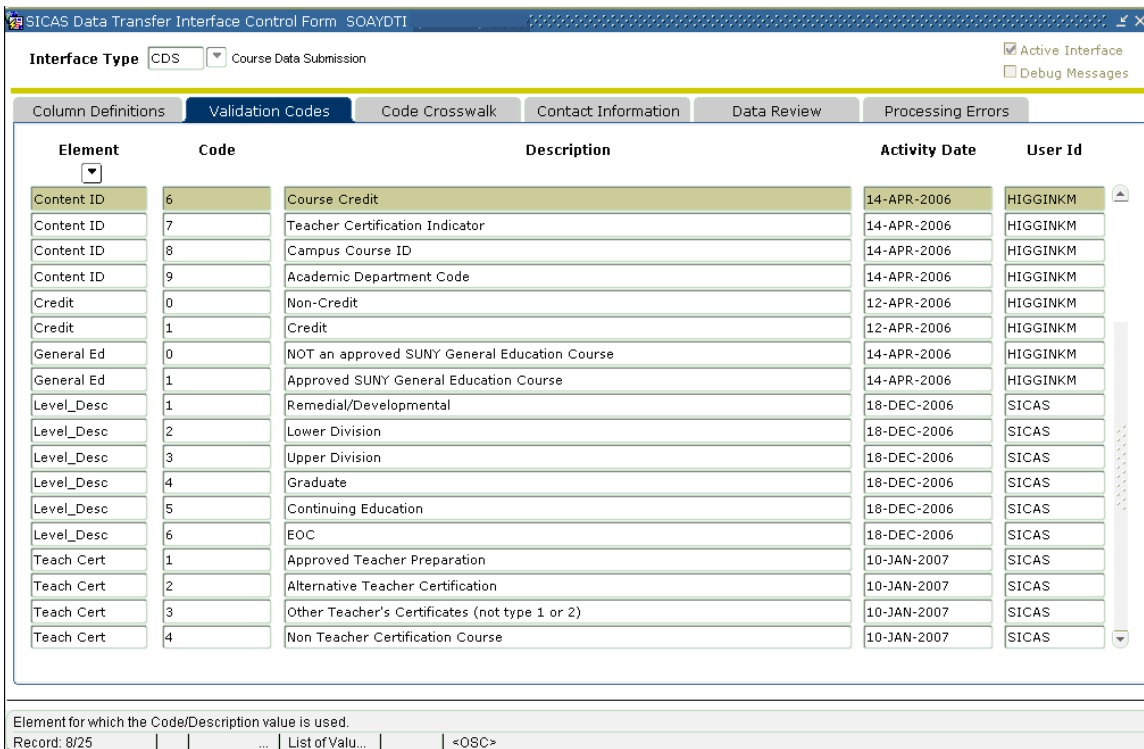
Figure 25 SOAYDTI Form – Validation Codes Tab shown for Interface Type CDS

Element – The Code value is valid for this item. The Element corresponds with the Column Name from the Column Definitions Tab.

Code – A valid code value for the element being validated.

Description – A Description of the code.

Activity Date – The Date this validation record was created or last updated.

User ID – The User ID of the person who created or last updated the validation record.

Code Crosswalk Tab

The Code Crosswalk Tab is used to crosswalk local codes to the established System Administration codes that are displayed on the Validation Codes Tab. The pull-down list for the

Interface Code column contains valid System Administration codes. Enter all required crosswalks on this tab. An asterisk appears in the Interface Code field when a value is not cross walked.

The only required crosswalk in CDS is for Teacher Certification courses.

In addition to entering the crosswalk for the 'Hispanic' data element, the SDS interface must have all values with a Crosswalk Source of GTVYVAL on the SOAYDTI Column Definitions tab cross walked on the Code Crosswalk tab. The lone exception to this rule is the codes used for the Criteria element. For the Criteria element you only need to crosswalk those values your campus collects and uses in the admissions decision process. Any test score that is not used or that the campus does not wish to send should be left un-cross walked.

For TSDS, not all the values on the Code Crosswalk Tab need to be cross walked. For example, for the Interface Element 'Dept Acct', crosswalk Campus Department Codes to the System Administration Department Codes but departments that do not provide courses (such as the Admissions Office) do not need to be cross walked. An asterisk appears in the Interface Code field when a value is not cross walked. Occasionally, Validation Codes are in separate tables instead of on the Validation Codes Tab. TSDS uses the SICAS IPEDS Instructional Code Validation Table STVYIPD, which contains the valid IPEDS codes. The STVYIPD table can be reviewed on the SICAS General Validation Table Inquiry Form GTIYVTL. Please see the STVYIPD section of this manual for more information.

Figure 26 The DTI Form – Code Crosswalk Tab shown for the CDS interface

Query Block

The Validation Table and Interface Element fields are used to query all crosswalk records for this application. Either field will reduce the number of crosswalk records returned. Both fields are filled in when a value is selected from either pull-down list. If both of these fields are left blank, all crosswalk records are displayed. In CDS, this block can be skipped since there is only one required crosswalk.

Validation Table – The Table or other source with the External Code values.

Interface Element – The item being cross walked. The Element corresponds with the Column Name from the Column Definitions Tab.

Crosswalk Block

The Crosswalk Block displays records that can be cross walked.   Values can be typed in or selected from a valid list generated from the validation table or other source. Invalid entries are marked with a comment in the Description field. Some programs tolerate un-cross walked values while others do not. All values that require crosswalks should be cross walked for CDS, SDS and SRDS.

Interface Element – The item being cross walked. The Element corresponds with the Column Name from the Column Definitions Tab.

Banner Source – The name of the Banner Table providing the data. The values in this table need to be cross walked.

Banner Value – A single value from the Banner Source that needs to be cross walked.

Description – The Description of the Banner Value.

Interface Code – A Code selected from the validation source—a validation table, view, or general column validation from the Validation Codes Tab. The pull-down list contains the valid codes for the active record. If these codes are not sufficient, use a Local Modification.

Interface Description – The Description of the selected Interface Code.

Contact Information Tab

Primary and optionally secondary contact information for the person or persons responsible for the data submission is required for IR Web Service Transmissions. The processing server and/or the SUNY IR staff use this contact information to communicate during data submission, locking, and approval.

Only a Primary Contact is required. Any number of Secondary Contacts may be entered, if desired.

Figure 27 The DTI Form – Contact Information Tab

Name of Contact – The first and last name of the Contact. This is a required field.

Job Title – The Title of the contact. This is a required field.

Contact Address – The mailing address of the contact. This field is not required.

E-Mail Address – The E-Mail address of the Contact. Emails are sent in response to processing status changes during the cycle of data transmission, locking, and approval. E-mails are sent to all Contacts listed. No other type of message is routinely sent. This is a required field.

Phone – The telephone number for the contact. IR staff will call if E-mail communication is not working efficiently. This is a required field.

User ID – The Contact's User ID associated with SUNY IR Web Applications. This is usually the same as their Local Campus E-mail user ID and is used when logging into SUNY websites. This is a required field.

Data Review Tab

The Data Review Tab is used to review data collected during processing. The preconfigured data views define the batches available for review. Usually a processing run creates a batch but this is not required. Batches can display any data or aggregation of data.

Figure 28 The DTI Form – Data Review Tab showing the Equivalent Listing Data View (CDS)

Query Block

Select the Interface Type, the Data View, and the Batch of data to review using the Interface Type, Data View, and Batch fields. The Batches available for viewing are determined by the Data View specification. After making the selections, use 'Next Block' to navigate to the Data Review Block. The Batches created by the application are saved until deleted by the user.

Data View – There are preconfigured views of the Extract Target Tables and the Banner Source Tables. These Data Views call the SICAS General Data Extraction Form GPAYEXT, which can be used to further manipulate the data. Refer to the 'Extract Data Button' in the Data Review Block section for more information on GPAYEXT.

Batch – The Batch of data to review using the selected Data View. Available Batch values are calculated based on the view selected. Usually, a Batch contains data from a single processing run.

The preconfigured Data View Options available for Course Data Submission (CDS) are:

CDS Data View Title	Base Table	Description
Course Listing	SOBYCDS	Base Course data with detail block displayed. Calls Data Extract Form without adding any columns.
Course Change Listing	SOBYCDS	List of courses with changes with detail block displayed. Calls Data Extract Form without adding any columns.
Teacher Cert Listing	SOBYCDS	Only displays Teacher Certification courses. Calls Data Extract Form without adding any columns.

General Education Listing	SOBYCDS	Only displays General Education courses. Calls Data Extract Form without adding any columns.
Wide Course Listing	SOBYCDS	Wider format listing of base courses without details. Calls Data Extract Form without adding any columns.
Wide Course Change Listing	SOBYCDS	Wider format listing of courses with changes. Calls Data Extract Form without adding any columns.
Wide SUNY ID Listing	SOBYCDS	Similar to Wide course listing with SUNY ID added. Calls Data Extract Form and adds the SUNY ID column.
Wide Non-CDS Listing	SOBYCDS	Wide format listing of Non-CDS courses. Calls Data Extract form without adding any columns.
Successful Courses List (with Force Resend Option)	SOBYCDS	List of courses successfully sent with checkbox for select courses to resend with the next submission. Calls Data Extract form without adding any columns.
Equivalents Listing	SOBYCDS	Base Course/Equivalent Course IDs and Titles only. Calls Data Extract Form and adds the Equivalent Course ID and Title columns.
Debug Listing	SFRYERR	List of Elements and Debug Messages with detail block displayed.
Profile Listing	SFRYEFF	List of the processing time for each element. This information is useful when debugging performance problems.
Transmission Progress/Results		Displays the progress/results of the transmission with a count and percentage of transmission statuses including: success, failure, errors, unknown, warning, non-CDS. Can be reviewed while process is running.
Error Listing	FFRYERR	List of Processing Errors returned during the run. Calls the Data Extract Form and adds Element, Error Code, and Error Message columns. This Data view is an alternative to the Processing Errors Tab but shows less information.
Warning Listing	SFRYERR	Calls the Data Extract Form and adds the Element, Error Code and Error Message columns.

The preconfigured Data View Options available for Student Data Submission (SDS) are:

SDS Data View Title	Base Table	Description
Student Listing	SOBYSDS	Campus ID, Student Name, Census or Final submission, SUNY ID, and Submission Status with detail block displayed. Calls Data Extract Form without adding any columns.
Student Repeating Data	SOBYSDS SORYSDS	Campus ID, Student Name, with detail block displayed for SUNY Report Type Code Values and Descriptions Calls Data Extract Form adding Census column to identify record as Census or Final Submission.
Student Ethnicity/Race Data	SOBYSDS SORYSDS	Campus ID, Student Name, Ethnicity, Hispanic Origin, Rep Type (SUNY report type code – Admission Criteria codes), Value (SUNY value for repeating data), Description of the repeating value Calls Data Extract Form without adding any columns
Wide Student Listing	SOBYSDS	Campus ID, Student Name, Educational Goal, Total Credits, Higher Ed. History, SUNY Id, Submission Status Calls Data Extract Form adding Census column to identify record as Census or Final Submission.
Primary Curriculum	SOBYSDS SORLCUR	Campus ID, Student Name, Period Code, SUNY ID, Primary Program ID, Program, Level Code and Degree Code
Secondary Curriculum	SOBYSDS SORLCUR	Campus ID, Student Name, Period Code, SUNY ID, Primary Program ID, Level Code, and Degree Code
Students Missing Curriculum Listing	SOBYSDS	Campus ID, Student Name, Period Code, Educational Goal, Total Credits, Higher Ed History, SUNY ID and Submit Status
EOT Student Additions	SOBYSDS	Lists the students who were not reported in the ESS submission and will be reported in the EOT submission. Both ESS and EOT must have been run in at least an audit mode.
Successful Student List (With Force Resent Option)	SOBYSDS	Campus ID, Student Name, Period Code, Educational Code, Total Credits, Higher Ed History, SUNY ID, and Submit Status
Debug Listing	SFRYERR	List of Elements and Debug Messages with detail block displayed
Profile Listing	SFRYERR	List of the processing time for each element. This information is useful when debugging performance problems.
Transmission Progress/Results		Displays the progress/results of the transmission with a count and percentage of transmission statuses including: success, failure, errors, unknown, warning, non-SDS. Can be reviewed while process is running.

Error Listing	SFRYERR	List of Processing Errors returned during the run. Calls the Data Extract Form and adds Element, Error Code, and Error Message columns. This Data View is an alternative to the Processing Errors Tab but shows less information.
Warning Listing	SFRYERR	List of processing warnings returned during the run. Calls the Data Extract Form and adds Element, Error Code and Error Message columns.
Convert Error Listing	SFRYERR	List of Conversion errors (CONVRT) returned during the run. Campus ID, Student Name, Element, and Error Code with detail block displaying error details. Calls Data Extract Form adding Error Message and Error Data columns.

There are eight preconfigured Data View Options available for Term/Section Data Submission (TSDS). The Sections with Students and the Sections with Instructors data views appear similar when viewed on SOAYDTI but when viewed on GPAYEXT, Student information displays from the Sections with Students Data View and Instructor information displays from the Sections with Instructors Data View. For more information on using GPAYEXT, see the Extract Data Button section.

The preconfigured Data View Options available for Term/Section Data Submission (TSDS) are:

TSDS Data View Title	Base Table	Description
Section Listing	SOBYSEC	Base Section data with detail block that includes Base Course information. Calls Data Extract Form without adding any columns.
Sections with Meetings	SOBYSEC	Base Section data with detail block that includes Section details with Meeting and Instructor information. Calls Data Extract Form without adding any columns.
Sections with Students	SOBYSEC	Base Section data with detail block that includes Section details with Students and Instructors associated with the section. Calls Data Extract Form includes Student information.
Sections with Instructors	SOBYSEC	Base Section data with detail block that includes Instructors and Students associated with the section. Calls Data Extract Form includes Instructor information.
Instructor Listing	SOBYINS	Instructor Name and ID with Class Schedule and Percent Effort Detail. Calls Data Extract Form without adding any columns.
Student Listing	SOBYSTU	Student Name and ID with Class Schedule and Grade Details. Calls Data Extract Form without adding any columns.
NO-TSDS Section Listing		Base Section data with detail block that includes Base Course information for courses marked by campus as "Do Not Send" on SSASECT. Calls Data Extract Form without adding any columns.
Successful Section List (with Force Resend Option)		Calls Data Extract Form without adding any columns.
Debug Listing	SFRYERR	List of Elements and Debug Messages with detail block displayed.
Profile Listing	SFRYERR	List of the processing time for each element. This information is useful when debugging performance problems.
Transmission Progress/Results		Displays the progress/results of the transmission with a count and percentage of transmission statuses including: success, failure, errors, unknown, warning, non-TSDS. Can be reviewed while process is running.

TSDS Data View Title	Base Table	Description
Error Listing	SFRYERR	List of Processing Errors returned during the run. Calls the Data Extract Form and adds Element, Error Code, and Error Message columns. This Data View is an alternative to the Processing Errors Tab but shows less information.
Warning Listing	SFRYERR	List of Processing Warnings returned during the run. Calls the Data Extract Form and adds Element, Error Code and Error Message columns.

The preconfigured Data View Options available for Financial Aid Data Submission (FADS) are:

FADS Data View Title	Base Table	Description
Applicant Listing (With Award Details)	RORYAWD	Student awards and resources information (Detail View box). Calls the Data Extract Form and adds Term Code, Fund Code, Award Amount Offered, Award Amount Accepted, and Award Amount Disbursed
Award Listing	RORYAWD	Student awards and resources information. Calls the Data Extract form and adds Period, Fund Resource Indicator, Award Program Code, Award Amount Offered, Award Amount Accepted and Award Amount Disbursed.
Other Award/Resource Listing	RORYFAD	Student resources information Calls the Data Extract Form and adds Period, Fund Resources Indicator, Fund Code, Other Award Name, Award Amount Offered, Award Amount Accepted, and Award Amount Disbursed.
Wide Applicant Listing	ROBYFAD	Student demographic information from the ISIR Calls the Data Extract form and adds PIDM, Local ID, SUNY ID, Date of Birth, Marital Status, Model Code, Stat Code Res, and Zip Code
Applicant Cost of Attendance Listing	ROBYFAD	Student's List of Cost of Attendance Elements Calls the Data Extract Form and adds Tuition, Student Fees, Room Board, Books Supplies, Transportation, Personal Expenses, Other Expenses and COA.
Wide Family Demographic Listing	ROBYFAD	Family demographic information from the ISIR Calls the Data Extract form and adds PIDM, Local ID and Family Member.
Student Demographic Listing	ROBYFAD	Student demographic information from the ISIR Calls the Data Extract Form and adds First FAFSA, Current FAFSA, Born Before, Have Children, Ins House Code, Year in College, Degree Type, and Exp Enroll Status,
Family Financial Listing	ROBYFAD	Family financial information from the ISIR Calls the Data Extract Form and adds Father's Income from Work, Mother's Income from Work, Spouses Income From Work, Income From Work, Parents Income Level, Income Level, EFC, and Second EFC
Debug Listing	SFRYERR	List of Elements and Debug Messages with detail block displayed.
Profile Listing	SFRYERR	List of the processing time for each element. This information is useful when debugging performance problems.

FADS Data View Title	Base Table	Description
Transmission Progress/Results		Displays the progress/results of the transmission with a count and percentage of transmission statuses including: success, failure, errors, unknown, warning. Can be reviewed while process is running.
Error Listing	SFRYERR	List of Processing Errors returned during the run. Calls the Data Extract Form and adds element, Error Code, and Error Message columns. This Data View is an alternative to the Processing Errors Tab but shows less information.
Warning Listing	SFRYERR	List of Processing Warnings returned during the run. Calls the Data Extract Form and adds Element, Error Code and Error Message Columns.
Convert Error Listing	SFRYERR	List of Conversion errors (CONVRT) returned during the run. Campus ID, Student Name, Element, and Error Code with detail block displaying error details. Calls Data Extract form adding Error Message and Error Data Columns.

The preconfigured Data View Options available for Student Revenue Data Submission (SRDS) are:

SRDS Data View Title	Base Table	Description
Student Listing	TOBYSRD	Student Name, Campus ID, and SUNY ID. Calls Data Extract Form without adding any columns.
Wide Student Listing	TOBYSRD	Student Name, Campus ID, SUNY ID, Birth Date and Gender. Calls Data Extract Form without adding any columns.
Wide Student Transaction Listing	TOBYSRD TORYSRD	Student Name, Campus ID, SUNY ID, Count, Revenue Name and Total Amount. Calls Data Extract Form includes Student information.
Student Transaction Detail	TOBYSRD TORYSRD	Student Name, Campus ID, SUNY ID, with detail block that includes detail code, revenue name, academic tern, total amount and transaction details. Calls Data Extract Form includes Instructor information.
Student Multiple Transaction Detail	TOBYSRD TORYSRD	Student Name, Campus ID, SUNY ID with detail block that includes revenue name, academic term, total amount and list of associated transaction. Calls Data Extract Form without adding any columns.
Debug Listing	SFRYERR	List of elements and debug messages with detail block displayed.
Profile Listing	SFRYERR	List of the processing time for each element. This information is useful when debugging performance problems.
Transmission Progress/Results		Displays the progress/results of the transmission with a count and percentage of transmission statuses including: success, failure, errors, unknown, warning, no-SRDS.
Error Listing	SFRYERR	List of Processing Errors returned during the run. Calls the Data Extract Form and adds Element,
Debug Listing	SFRYERR	List of Elements and Debug Messages with detail block displayed.
Profile Listing	SFRYERR	List of the processing time for each element. This information is useful when debugging performance problems.
Transmission Progress/Results		Displays the status of the transmission with a count and percentage of transmission statuses including: success, failure, errors, unknown, warning, no-SRDS. Can be reviewed while process is running.

SRDS Data View Title	Base Table	Description
Error Listing	SFRYERR	List of Processing Errors returned during the run. Calls the Data Extract form and adds Element, Error Code, and Error Message columns. This Data View is an alternative to the Processing Errors Tab but shows less information.
Warning Listing	SFRYERR	List of Processing Warnings returned during the run. Calls the Data Extract Form and adds Element, Error Code and Error Message Columns.
Convert Error Listing	SFRYERR	List of Conversion errors (CONVRT) returned during the run. Campus ID, Student Name, Element, and Error code with detail block displaying error details. Calls Data Extract Form adding Error Message and Error Data columns.

DADS Data View Title	Base Table	Description
Degree Listing	SORYDEG STVYAPE SOBYDAD	Level, Credits, Degree, Awarded, Primary and Major. Calls the Data Extract form and adds Degree Code, Level Code, Primary/Secondary, Program APES, Description, SUNY ID and Submit Status.
Multiple Degree Listing	SORYDEG STVYAPE SOBYDAD	Level, Credits, Degree, Awarded, Primary and Major. Calls the Data Extract Form and adds Degree Code, Level Code, Primary/Secondary, Program APES, Description SUNY ID and Submit Status.
Multiple Major Listing	SORYDEG STVYAPE SOBYDAD	Level, Credits, Degree, Awarded, Primary, Major, Secondary and Major. Calls the Data Extract Form and adds Degree code, Level Code, Primary/Secondary, Program APES, Description, SUNY ID and Submit Status.
Student Ethnicity/Race Data	SOBYDAD SORYDAD	Ethnicity Data and Race Detail. Calls the Data Extract Form and Rep Type, Value, and Description.
Wide Student Listing	SORYDEG STVYAPE SOBYDAD	Student's Name, Campus ID, SUNY ID, Degree, Level, Primary/Secondary, Program APES and Description. Calls the Data Extract Form and adds SUNY ID and Submit Status.
Successful Record Listing (with Force Send Option)	SORYDEG SOBYDAD	Campus ID, Student Name, Degree code, Level Code, SUNY ID and Submit Status.
No Send Listing (With Force Send Option)	SORYDEG SOBYDAD	Campus ID, Student Name, Degree Code, Level Code, SUNY ID and Submit Status.
Summer Students Not In No Send Listing	SORYDEG SOBYDAD	Campus ID, Student Name, Degree Code, Level Code, SUNY ID and Submit Status.
Out of Period Degrees	SORYDEG STVYAPE SOBYDAD	Campus ID, Student Name, Degree Code, Level Code, Primary/Secondary, Program APES, Description, SUNY ID and Submit Status.
Missing APES Code Records	SORYDEG SOBYDAD	Campus ID, Student Name, Degree Code, Level Code, SUNY ID and Submit Status.
Debug Listing	SFRYERR	List of Elements and Debug Messages with detail block displayed.
Profile Listing	SFRYERR	List of the processing time for each element. This information is useful when debugging performance problems.
Transmission Progress/Results		List of the progress/results of the submission with a count and percentage of transmission statuses including: success, failure, errors, unknown, warning, non-DADS. Can be reviewed while process is running.
Error Listing	SFRYERR	List of Processing Errors returned during the run. Calls the Data Extract Form and adds Element,

DADS Data View Title	Base Table	Description
Warning Listing	SFRYERR	List of Processing Warnings returned during the run. Calls the Data Extract Form and adds Element, Error Code and Error Message Columns.
Convert Error Listing	SFRYERR	List of Conversion errors (CONVRT) returned during the run. Campus ID, Student Name, Element, and Error code with detail block displaying error details. Calls Data Extract Form adding Error Message and Error Data columns.

Data Review Block

The Data Review Block dynamically displays data based on the Data View and Batch selections. Depending on the Data View selected, the Data Review Block will either display as one block or it will have a split view with a multiple record browse on the left and a detail display titled 'Additional Columns' on the right. The 'Additional Columns' section contains detailed information about the highlighted record on the left.

The Data Review Block supports standard F7 (Enter Query) and F8 (Execute Query) functionality. Use the wildcard character '%' to create 'like' matching queries in either the 'Browse' or the 'Additional Columns' areas. As an example, to create a subset of courses, Press F7 and enter %Gen Ed Apprv: Yes% into the 'Additional Columns' area then Press F8 to return only records approved for General Education.

Extract Data Button (GPAYEXT)

Picking the 'Extract Data' button at the bottom of the Data Review Tab opens the SICAS General Data Extraction Form GPAYEXT. You cannot navigate to this form directly; it can only be accessed from within an application. When called from within an application it provides additional data selection, manipulation, and extract features for greater customization for reporting and data extraction. Initially, only the Data included in the preconfigured Data View Option selected on the Data Review tab displays. Use the GPAYEXT form to add conditions to limit the number of records displayed, change the sort order, and add columns. The data listings and extracts created on GPAYEXT can be saved for historical reference and for use in other software programs such as Microsoft Excel, Access, or SAS. Refer to the SICAS General Data Extraction Form GPAYEXT Manual for more information on using the GPAYEXT form.

Figure 28 is an example of the Equivalents Listing Data View—a wide Data View with no Additional Columns area. **Error! Reference source not found.** is the transition of this Data View to GPAYEXT when the 'Extract Data' button is selected. This demonstrates the ability to create columns automatically for commonly needed values using GPAYEXT.

Use the 'Column Options' button on the left side of the GPAYEXT form to display the courses in the same order that they appear in the Data View.

Figure 29 The GPAYEXT Form displays when the Extract Data Button on the Data Review Tab is selected

Processing Errors Tab

The Processing Errors Tab (**Error! Reference source not found.**) displays Errors and Warnings returned during data transmission. This tab also contains Debugging information if the 'Debug Messages' box in the Key Block is checked. Audit runs do not generate errors.

All data specific errors generated during data transmission are listed on the Processing Errors Tab and on the System Administration Web Application. Threshold errors are displayed only on the System Administration Web Application. Web Service Transmission errors generated by unsuccessful Web Service Calls are only recorded locally and displayed on the Processing Errors Tab. Any errors that require edits must be corrected in Banner and not on System Administration's website.

Appendix B contains a list of the Transmission Error Codes related to the IR Submission process. There are severity levels associated with the error codes.

Severity 1   Fatal (You must clean up all Fatal Errors before the file can be locked.)
Severity 2   Warning
Severity 3   PSI Warning (The process could not find the building or room in the PSI data. This only applies to State Operated Campuses.)
Severity 4   This is a deferred error and is usually associated with Faculty Information. It indicates that the process could not find an HR record for the faculty member. It is deferred until SA can create a live link into the HR data at System Administration. (This only applies to State Operated Campuses.)

There are several types of error codes:

SA-Dfr   Indicates a System Administration Deferred Error/Non-Fatal. Severity = 4.

SA-Err   Indicates a System Administration Error. The record must be altered for
                acceptance. Severity = 1.

SA-ERR   Indicates a System Administration Error. Unknown Error, Unexpected Error.
                Severity Unknown.

SA-PSI   Indicates a System Administration PSI Warning. Building/Room Codes could not
                be found in PSI lookup. Severity = 3.

SA-WRN   Indicates a System Administration Warning. The record is accepted. If greater
                than 15% of the records have the same type of warning, a threshold error is
                generated. Threshold errors are only displayed on the System Administration
                Web application. Severity = 2.

Convrt   Conversion not completed on the Crosswalk tab of SOAYDTI. The crosswalk
                must be completed.

Dup   Duplicate Warning. Value already exists for this Key, value ignored

WS   Indicates a Web Service Submission error.

DEBUG   Indicates a Debug Message. Debug Messages are displayed only if the 'Debug
                Messages' box (located in the Key Block of the SOAYDTI form) is checked.
                When activated, messages are added to the Processing Errors Tab with a code
                of DEBUG. Debug messages provide additional Status Information from Web
                Services (values returned from the Submission Status Service), Process Markers
                (timestamps of specific processing events), and Summary Timing (some markers
                list total time for loops). The Debug Messages obscure actual errors therefore
                this option should not be used for regular application runs, however the Debug
                option might provide SICAS and campus IT departments with useful information
                during the Pilot and first Production Runs.

Figure 30 The DTI Form - Processing Errors Tab shown for the SCRYCDS process

Process – The name of the process that generated the error.

For CDS, this is the SICAS Course Data Submission Report SCRYCDS.

For SDS, this is the SICAS Student Data Submission Report SGRYSDS.

For TSDS, this is the SICAS Term Section Data Submission Report SSRYTSD.

For FADS, this is the SICAS Financial Aid Data Submission Report RORYFAD.

For SRDS, this is the SICAS Student Revenue Data Submission Report TSRYSRD.

For DADS, this is the SICAS Degree Awarded Data Submission Report SORYDAD.

Term – The Term processed.

Seq – The Sequence number identifying the order errors occurred.

Code – The Oracle, SUNY, or specific application error code depending on the error encountered.

Message – The error message associated with the error code. For processing errors, it is the Oracle Error Message associated with the Code value. For application errors, the message indicates the error encountered during the collection or presentation of data for a specific record in the table. For transmission errors, it is the error text returned by the Web Service call.

Element – The Column Name (Element) on the Column Definitions Tab that created the error (not all errors occur for a specific element).

Target Column – The name of the column in the target table that is updated by the Element.

Error Details Tab

The Error Details Tab is located in the lower portion of the Processing Errors Tab on the SOAYDTI/ROAYDTI/TOAYDTI forms. It contains information about the highlighted Error. Appendix B contains a list of the Transmission Error Codes related to the IR Submission process.
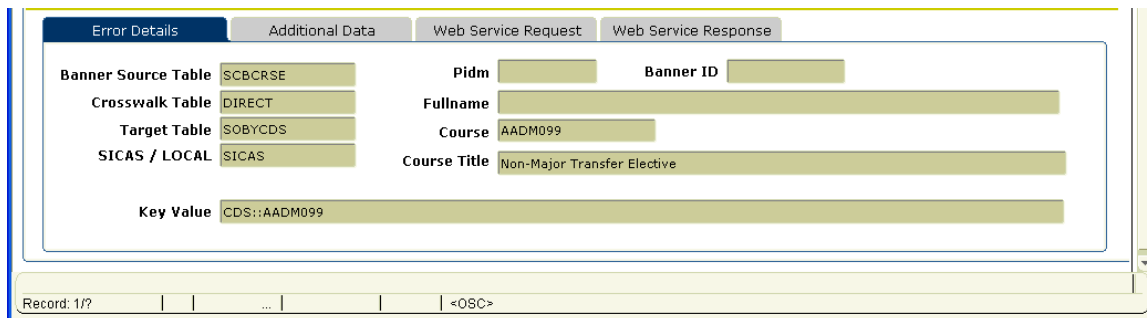
Figure 311 The DTI Form – Error Details Tab in lower portion of the Processing Errors Tab (CDS shown).

Banner Source Table – The name of the Banner Table providing data for the element (Column Name on the Column Definitions Tab).

Crosswalk Table – The table or view used to crosswalk Banner values to required External Codes or a description of the method used to determine the value if it is not cross walked (same as the Crosswalk Source methods on the Column Definitions Tab). Most CDS elements use the Direct method (values are extracted directly from Banner without validation conversion).

Target Table – The name of the extract table the active Target Column updates.

 In CDS, there are two extract tables:

      SICAS Base Course Data Submission Table SOBYCDS – Base Course Data
      SICAS Course Data Submission Table SORYCDS – Equivalent Course Data

 In SDS, there are two extract tables:

      SICAS Student Submission Base Table SOBYSDS – Base Student Data
      SICAS Student Submission Repeating Data Table SORYSDS – Admissions/Placement
        Criteria, Disability Status, and Race Codes

 In TSDS, there are several extract tables:

      SICAS Section Data Submission Table SOBYSEC – Base Section Data
      SICAS Section Data Submission Meeting Table SORYMET – Section Meetings Data
      SICAS Section Data Submission Meeting Table SORYMST – Section Meetings Data for
        Sections meeting at the same time
      SICAS Section Student Base Table SOBYSTU – Student Data
      SICAS Section Student Repeating Table SORYSTU – Student Data
      SICAS Section Faculty Base Table SOBYINS – Instructor Data
      SICAS Section Faculty Repeating Effort Table SORYINS – Instructor Data

In FADS, there are two extract tables.

      SICAS Financial Aid Data Submission Table ROBYFAD – Base Fin Aid Data
      SICAS Financial Aid Awards Repeating Table RORYAWD

In SRDS, there are two extract tables:

      SICAS Revenue Data Submission Student Base Table TOBYSRD – Revenue Data
      SICAS Student Revenue Submission Repeating Data Table TORYSRD – Student Data

In DADS, there are three extract tables:

      SICAS Degree Awarded Submission Student Base Table SOBYDAD – Base Student
        Data
      SICAS Student Submission Repeating Data Table SORYDEG – Degree Data
      SICAS SUNY Program Validation (APES & APIS) STVYAPE – APES/APIS Data

SICAS/Local – Indicates whether the SQL code is SICAS provided or a Local Modification.

Pidm – The Banner PIDM for the individual.

Banner ID – The Banner/Campus ID related to the PIDM.

Fullname – The Full Name of the person related to the PIDM.

Course – The Campus Course Code.

Course Title – The Course Title that corresponds with the Campus Course Code.

Key Value – Special Key Value used to attach error messages back to Element Source and XML storage.

Additional Data Tab

The Additional Data Tab is located in the lower portion of the Processing Errors Tab on the SOAYDTI/ROAYDTI/TOAYDTI forms. It contains additional information about the highlighted Error including any relevant data. The contents depend on the type of error displayed. Appendix B contains a list of the Transmission Error Codes related to the IR Submission process.
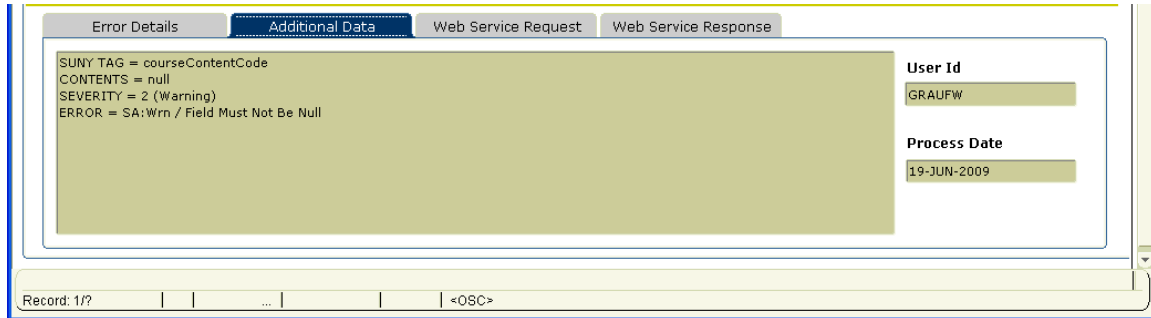
Figure 32 The DTI Form – Additional Data Tab in the lower portion of the Processing Errors (CDS shown).

User ID – The User ID of the person who added or last updated the error record. Usually this is the user who ran the process to build or transmit the data.

Process Date – The date the process ran.

Web Service Request Tab

The Web Service Request Tab is located in the lower portion of the Processing Errors Tab on the SOAYDTI/ROAYDTI/TOAYDTI forms. It is visible only when data exists in the SICAS General Web Service XML Transmission Table GORYXML. This tab displays the first 4,000 characters of the XML transmitted to System Administration (4,000 characters usually contains the entire SOAP Envelope). This is the actual text of the Web Service Request and programmers at System Administration can use this information when debugging transmission errors.
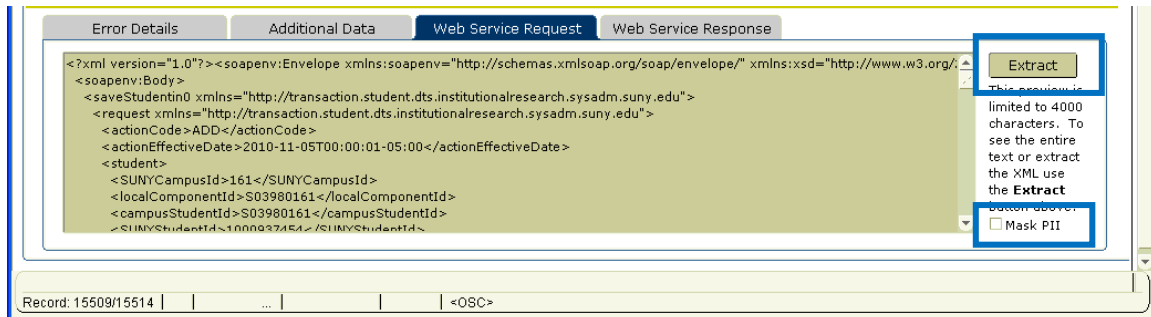
Figure 33 The DTI Form – Web Service Request Tab in lower portion of the Processing Errors Tab (CDS Shown)

To send this information to System Administration, click on the extract button on the right side of the page. Then click on the Help menu and select Extract Data No Key. This will export the XML from Banner and into an excel spreadsheet. You must have pop-up blockers disabled to use this feature. To mask any Personally Identifiable Information (PII), check the Mask PII box before extracting the XML. This will mask the PII in the XML based on the rules established in the PII Masks element on the Column Definitions tab. DO NOT SEND PII VIA EMAIL.
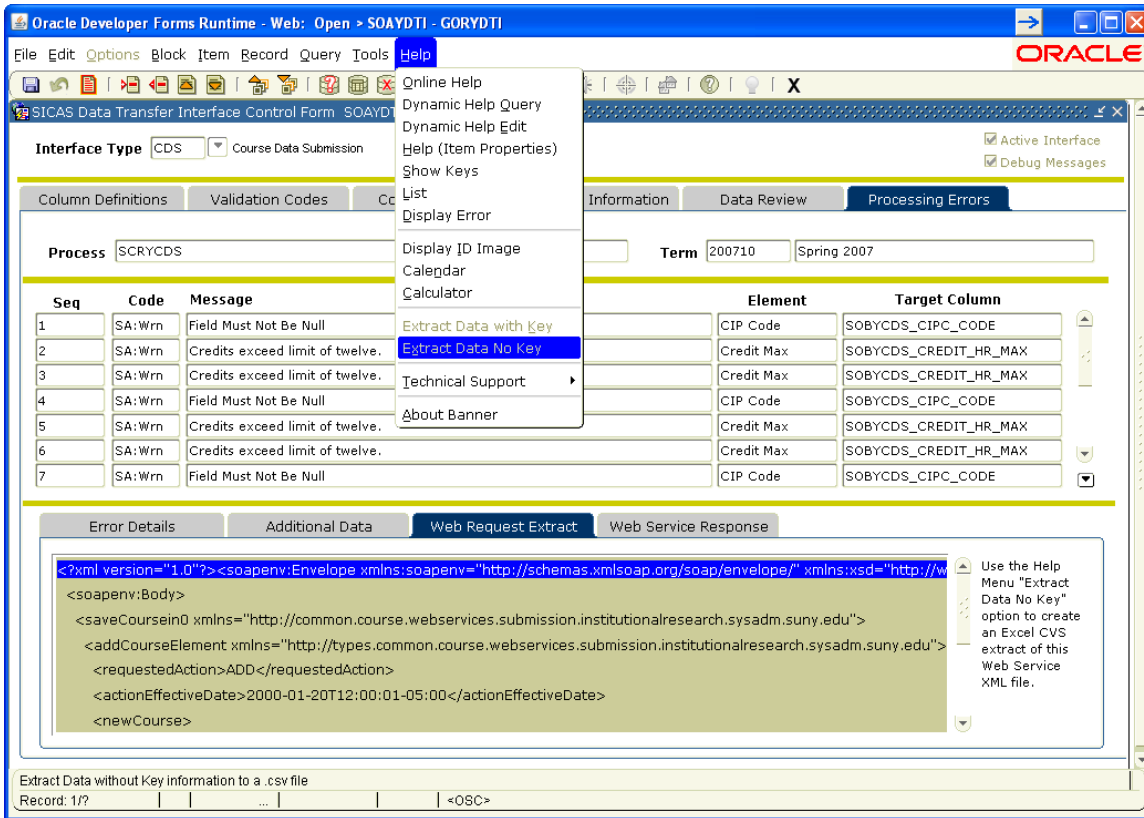
Figure 34 The DTI Form – Web Service Request Tab and the Help Menu extracting data

Web Service Response Tab

The Web Service Response Tab is located in the lower portion of the Processing Errors Tab on the DTI form. It is visible only when data exists in the SICAS General Web Service XML Transmission Table GORYXML. It looks very much like the Web Service Request Tab. This tab displays the first 4,000 characters of the XML received from System Administration in response to the request contained on the Web Service Request Tab (4,000 characters usually contains the entire SOAP Envelope). This is the actual text of the Web Service Response and programmers at System Administration can use this information when debugging transmission errors. To send this information to System Administration, follow the steps outlined above in the Web Service Response section.
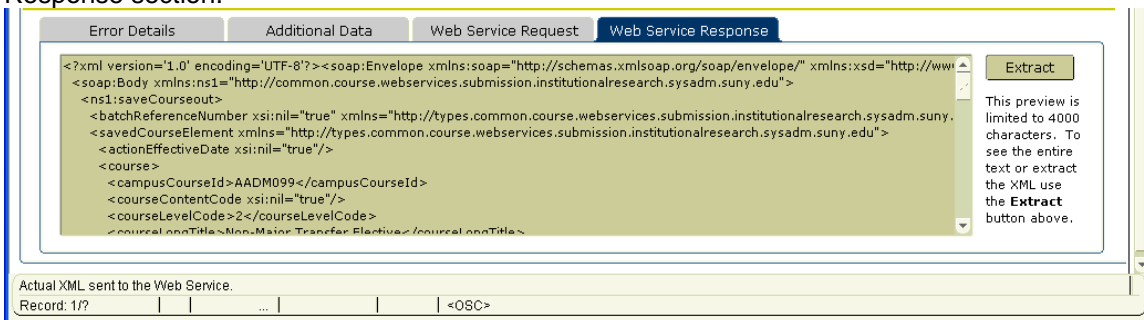
Figure 35 The DTI Form – Web Service Response Tab in lower portion of the Processing Errors Tab (CDS Shown)

SCACRSE   Basic Course Information Form

The Basic Course Information Form SCACRSE contains information on Course Details, SUNY Institutional Research Data, Hours, and Repeat Details. A valid Subject, Course, and Term must be selected before the information in the Course Details block will display. The Course Details information is specific to the course selected in the Key Block.

SICAS modified SCACRSE to include data elements required for the SUNY SIRIS submissions. The SICAS modifications support the Course Data Submission (CDS), the Student Data Submission (SDS), and the Term/Section Data Submission (TSDS). This documentation only covers these modifications and not the general use of the SCACRSE form.

Figure 36 SCACRSE – Basic Course Information Form with SICAS Modifications

SUNY Institutional Research Data

The SUNY Institutional Research Data block contains data required for IR Data Submissions. All fields in this block, except for the SUNY ID field, may be altered by an end-user with Banner Maintenance Access to the SCACRSE form.

The SUNY ID is a Unique Course ID returned by SUNY System Administration the first time a course is sent on a transmission. This ID is unique across all campuses and is a constant for each course. The SUNY ID can be useful when accessing data on System Administration's website. It is a read only field and may not be altered by the end-user.

The default values listed for the SUNY IR Data fields are used only when a new record is created. If the course record is copied to a new term effective record, using the 'Copy' button located in the Course Details block, the existing values for the most recent effective record are used to populate the new record.

Location – The Location used to teach the course. The default for a new course record is 'On Campus'. This field is used for TSDS. All sections created on SSASECT after this field is populated will have the default value from SCACRSE. All variations to this default will need to be manually done.

> Valid Entries:
>> On Campus – The course is taught on your campus. This is the default.
>> Another Campus – The course is taught at another college (section requires IPEDS code for the college).
>> Clinical Facility
>> Student Site (DL) – Distance Learning
>> High School
>> Correctional Facility
>> Community Site
>> Study Abroad (our Prog) – The course section requires an approved SUNY Study Abroad Code.
>> Study Abroad (other) – The course section requires approved SUNY Study Abroad Code.
>> Community/Off-Campus Site
>> Multiple Location (On Campus/Abroad)

Level – The course Level. This field has no default value and must be filled in when a new course record is created. This field is used for CDS and SDS.

> Valid Entries:
>> Remedial/Developmental (Obsolete Fall 2013)
>> Lower Division
>> Upper Division
>> Graduate
>> Continuing Education
>> EOC – Educational Opportunity Center
>> Remedial/Cont. Ed. (Community Colleges only)
>> Remedial – Non-Continuing Ed
>> Developmental

Funding – The funding source of the course. This field is used for TSDS.
> The default value for a new course record is 'State Supported'.

> Valid Entries:
>> State Supported – Fall, Spring and Winter Terms. During the submission, this automatically becomes Self Support/Tuition for sections created in the Summer Term for State Operated Campuses. The value will remain state supported for Community Colleges if the SICAS_TERM_TYPE entries are properly done on GTVSDAX. Please see the GTVSDAX section of this manual for the proper setup.
>> Self Support/Tuition
>> Contract Course
>> State Supported/Fee
>> Self Support/Fee

SUNY ID – This is a read only value from System Administration. SUNY ID is the Unique Course ID returned by SUNY System Administration the first time a course is sent on a transmission. This number is unique across all campuses and is a constant for each course. An existing Campus Course Id can be forced into new course statue (see Action field) when course content changes are made. The SUNY ID may be useful when reviewing course data on the System

Administration website. The SUNY ID is returned by the CDS Web Service and sent as part of the TSDS IR submission.

Credit Course (check box) – Placing a check in this box indicates that the course provides credit towards a degree.

General Ed Approved (check box) – Placing a check in this box indicates that the course is a General Education approved course. The set up for GETA transcript is used to mark courses as General Ed Approved. Courses should be reviewed to ensure they are marked accurately.

Contact Hours- Enter the default contact hours for Faculty and Students in these fields. Values entered in these fields will be used when new sections are created on SSASECT. Changes should be made on SSASECT if an individual section has a different value for contact hours. If the field Student Contact hours is left blank, a default of 15 X the number of credit hours will be transmitted in the TSDS submission. There is no default for Faculty Contact hours so this field must be entered for each section.

Facility – The type of facility most desired for teaching this course. This is not necessarily where the course is taught. This field is intended to show unmet need for facilities. The default for a new course record is 'Classroom'.

> Valid Entries:
> > Classroom – This is the default.
> > Lecture Hall
> > Class Lab
> > Individual Lab
> > Autotutorial Lab (Wet)
> > Autotutorial Lab (Dry)
> > Remote
> > Project Lab/Studio
> > Physical Education
> > Other
> > None/Distance Ed

Action – The Action to take for this course record during the next data transmission. Course records marked 'Do Not Send' in the CDS application create sections marked 'Do Not Send' for TSDS transmissions. To be included in the SDS data transmission, a course must be included in the CDS data transmission, have a Course Level assigned, and have an Action status of 'Ready'.

Valid Entries:

Ready	Ready to transmit record values. Records with this option are transmitted as needed. This is the default value for new course records.
Do Not Send	Do Not Send course record or sections related to the course record. Course records marked 'Do Not Send' in the CDS application create sections marked 'Do Not Send' for TSDS transmissions. Students enrolled in Sections marked 'Do Not Send' are not included in the SDS transmission.
Correction	Indicates a simple edit that does not need to be retained in the course record history table. System Administration does not currently support this option. At this time, the Correction Action functions as Ready. Use Force Update for Corrections.
Force Update	Forces the transmission process to send the record even if no changes appear to have been made. Use this option if the System Administration

records do not match local data but the application indicates the data are current.

Figure 37 SCACRSE – Basic Course Information Form - showing SICAS modification for Course Special Attributes

Special course attributes are displayed on the Basic Course information Form SCACRSE on the Course Special Attributes tab. Course attributes are defined on the Attribute Validation Form STVATTR. Attributes are then added to the Degree Attributes tab on the Course Detail Information Form SCADETL and to the Degree Program Attributes on the Schedule Detail Form SSADETL. These attributes will then need to be cross walked to the System Administration codes on SOAYDTI.

SSASECT   Schedule Form

SICAS modified the Schedule Form SSASECT to include data elements required for the SUNY Term/Section Data Submission (TSDS). This documentation only covers these modifications and not the general use of the SSASECT form. Both versions of the Student Data Submission (SDS), Early Student Submission (ESS) and End of Term (EOT), require that the shadow data be present on IR tab and that the section be marked "Ready" in the action field.

SUNY Institutional Research Data

Pick the SUNY Institutional Research tab to display the data elements required for SUNY IR Term/Section Data Submissions. The SUNY IR Submission Details button will be grayed out if the Term in the Key Block is not greater than or equal to the GTVSDAX entry for the Internal Code FIRST_TERM.

The Campus Course Id, Campus Section Id, Instruction Type, Location Description, Activity Date, and User ID fields cannot be altered by the end-user. The Campus Course ID and Section ID fields are locally generated values used to identify the course and section to System

Administration. Instruction Type is the Schedule Type on the main form and is redisplayed for use in calculating Contact Hours. The Location Description, Activity Date, and User ID fields are filled in automatically. All other fields can be edited when the user has Banner Maintenance Access to the form.

System Administration does not return SUNY IDs for section records. The locally generated Campus Section ID can be used for identifying sections. The SUNY ID for the course displays on the Base Course Information Form SCACRSE.

Figure 38 SSASECT – Schedule Form with the SUNY IR Submission Details Dialog Box displayed

SUNY IR Submission Details

The fields displayed on the SUNY Institutional Research tab box are submitted to System Administration to satisfy the requirements of the SUNY Term/Section Data Submission (TSDS). The Desired Facility is the only entry not submitted directly to System Administration.

This list is an explanation of the fields in the SUNY IR Submission Details dialog box.

Campus Course ID – A locally generated value used to identify the course.

Campus Section ID – A locally generated value used to identify the section. System Administration does not return a SUNY ID for sections. Use the locally generated Campus Section ID displayed here as if it was a SUNY ID.

Funding Source – The Funding Source for this section of the course. This field defaults from the Funding field on SCACRSE.

       Valid Entries:
              State Supported – Fall, Spring, Winter Terms. During the submission, this automatically becomes Self Support/Tuition for sections created in the Summer Term for State Operated Campuses. The value will remain state supported for

Community Colleges if the SICAS_TERM_TYPE entries are properly done on GTVSDAX. Please see the GTVSDAX section of this manual for the proper setup.

Self-Supported/Tuition
Contract Course
Self-Supported/Fee
State Supported/Fee

Insurance – The type of Insurance required for students attending the section.

Valid Entries:
No Insurance Required – This is the default.
Social Work Practicum
Clinical/Medical Practicum

Fac Contact Hours – Enter the total hours spent by faculty in direct contact with students during the Section Period. Fifty minutes of instruction is one faculty contact hour. Preparation time or Office Hours are not included. Example: A typical three-credit lecture yields 45 faculty contact hours; a typical three-credit Independent Study yields zero faculty contact hours. This is a required field. Campuses can enter a value in the Faculty Contact hours field on SCACRSE which will be used as the default for all sections that are created in the future. This data can be changed at the section level on SSASECT. Campuses may also elect to use the local mod point FZ_TSDS_FACULTYHRS to calculate this element. Information on this local mod can be found in the "local Modifications" section of this manual and in Appendix G. Failure to use either option will result in zero (0) being sent for this element.

Student Contact Hours – The total hours spent by the individual student in the scheduled instruction. Total Student Contact Hours are usually the same as Total Faculty Contact Hours, but may differ in those instances when the faculty is not in full/direct supervision of the student (e.g., a typical three credit Lecture will yield the same 45 contact hours for both student and faculty while a typical three-credit Independent Study will yield 45 student contact hours and zero faculty contact hours). If this element is left blank, a default value equal to 15 times the number of credits will be sent.

Instruction Type – The Instruction Type (same as Schedule Type on main form) used in determining the Contact Hours fields.

Other Contact Hours – Enter the number of weekly student contact hours that will generate one credit hour assuming a standard 15 week semester—only required when Crd/Cntct (TT100) is set to 008 – Other.

Desired Facility – The type of facility most desired for teaching this course. This is not necessarily where the course is taught; it is intended to show unmet need for facilities. This field defaults from the Facility field on SCACRSE. This entry is used to select a facility type for meetings established for the section. The default value for new course records is 'Classroom'.

Valid Entries:
Classroom – This is the default.	Remote
Lecture Hall	Project Lab/Studio
Class Lab	Physical Education
Individual Lab	Other
Autotutorial Lab (Wet)	None/Distance Ed
Autotutorial Lab (Dry)	

Location Block – This area contains information about where the section is taught.

Type – The Type of Location where the section is taught. The default value is 'On-Campus'.

Valid Entries:

      On-Campus – The section is taught on your campus. This is the default.

      Another College – The section is taught at another college. Enter the IPEDS code for the college in the Other College field.

      Clinical Facility

      Student Site – Distance Learning

      High School

      Correctional Facility

      Community Site

      Study Abroad (our program) – The section requires an approved SUNY Study Abroad Code. Enter the code in the Study Abroad field.

      Study Abroad (other) – The section requires an approved SUNY Study Abroad Code. Enter the code in the Study Abroad field.

      Community/Off-Campus Site

      Multiple Location (On Campus/Abroad)

Other College – When the section is taught at 'Another College', an IPEDS Code for that campus must be selected from this pull-down list.

Study Abroad – When the section is taught Abroad, a SUNY Study Abroad Code must be selected from this pull-down list.

Description – A description of the Code selected for Other College or Study Abroad.

Action – The Action to take for this course record during the next data transmission. Course records marked 'Do Not Send' in the Course Data Submission (CDS) application create sections marked 'Do Not Send' for the Term/Section Data Submission (TSDS) application. When this occurs, the Action field may not be changed for the TSDS transmission. All Sections included in the Term/Section Data Submission (TSDS) must have an Action status of 'Ready' (Ready to Submit) to be included in the Student Data Submission (SDS). Only students enrolled in Sections that meet SIRIS criteria are included in the Student Data Submission. The default Action status value for new course records is 'Ready'.

Valid Entries:

Ready	Ready to transmit record values. Records with this option are transmitted as needed. This is the default value for new course records.
Correction	Indicates a simple edit that does not need to be retained in the course record history table. System Administration does not currently support this option. At this time, the Correction Action functions as Ready. Use Force Update for Corrections.
Force Update	Forces the transmission process to send the record even if no changes appear to have been made. Use this option if the System Administration records do not match local data but the application indicates the data are current.
Do Not Send	Do Not Send course record or sections related to the course record. This option affects both CDS and TSDS transmissions. Course records marked 'Do Not Send' in the CDS application create sections marked 'Do Not Send' for TSDS transmissions.

Activity Date – The Date the record was created or last updated.

User ID – The User ID of the person who created or last updated the record.

Scheduled Meeting Times Tab

This tab contains information for each meeting created for the course displayed on the SSASECT main form. SICAS modified the Meeting Times Block to include data elements required for the SUNY Term/Section Data Submission (TSDS). Two fields are added for Desired Facility. This documentation only covers these modifications and not the general use of the Scheduled Meeting Times Block.
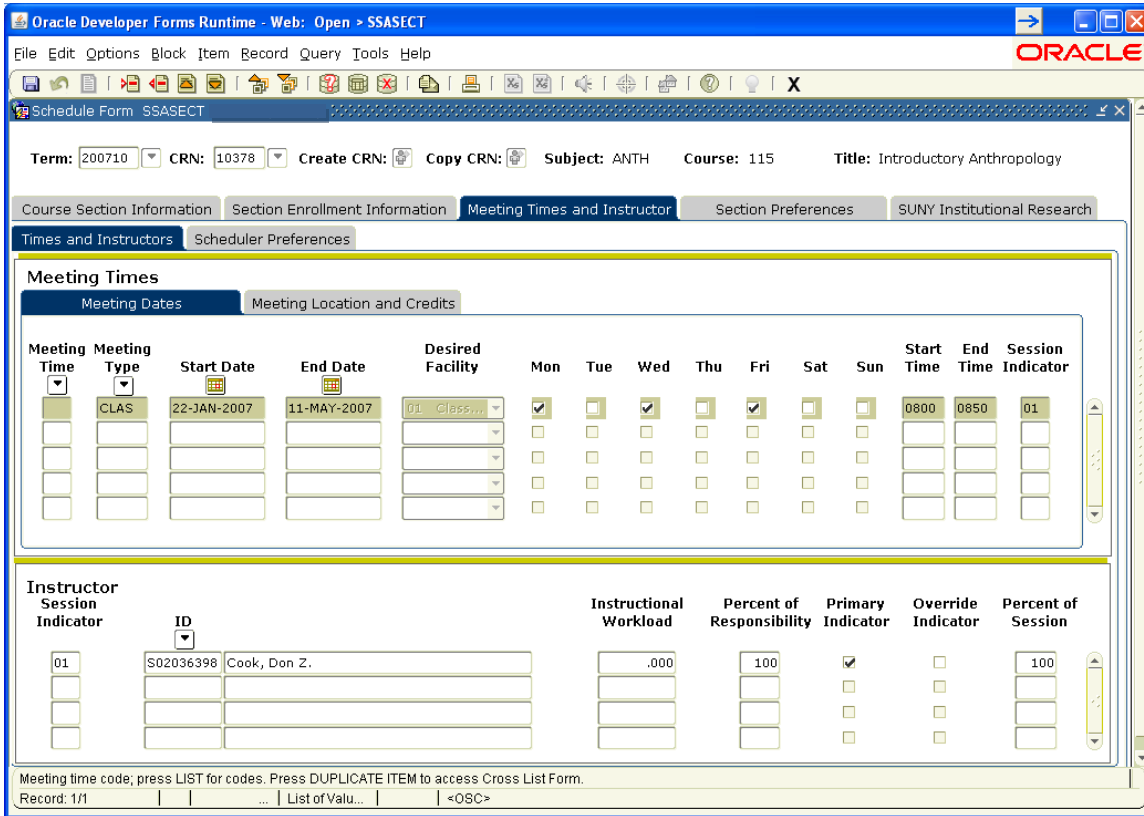
Figure 39 SSASECT Form with the Scheduled Meeting Times Block displayed

Desired Facility – The type of facility most desired for teaching a specific course meeting. This is not necessarily where the course is taught. It is intended to show unmet need for facilities. The value is sent with this meeting record as part of the Term/Section Data IR submission.

Valid Entries for the Desired Facility field for transmission to System Administration:

Classroom	Remote
Lecture Hall	Project Lab/Studio
Class Lab	Physical Education
Individual Lab	Other
Autotutorial Lab (Wet)	None/Distance Ed
Autotutorial Lab (Dry)	

GTVSDAX   Crosswalk Validation Form

Use the Crosswalk Validation Form GTVSDAX to verify External Code values required for the SIRIS applications. This table lists the Codes that must be reviewed and verified before running the IR Submission processes. External Codes may vary by institution. The user must enter Codes specific to their institution.

GTVSDAX Entries Required for the IR Submission Applications

Internal Code	Internal Group	Displays External Code For
PROD_DB	SICAS_IR	Name of Production Banner Database Required to use Web Services
USA	ADDRESS	Permanent address type for U.S. students Required for SDS
FOREIGN	ADDRESS	Permanent address type for Foreign students Required for SDS
PELL_FUND	SICAS_SDS	Pell Grant Fund Code Required for SDS
TAP_FUND	SICAS_SDS	TAP Grant Fund Code Required for SDS
FIRST_TERM	SICAS_SSBYSEC	The First Term to collect Term/Section data Required for TSDS
PSI_ACTIVE	SICAS_SLRRDEF	Report PSI information to System Admin (Y, N) Required for TSDS
FALL	SICAS_TERM_TYPE	Last two digits of the Fall Term Code Required for SDS and TSDS
SPRING	SICAS_TERM_TYPE	Last two digits of the Spring Term Code Required for SDS and TSDS
WINTER	SICAS_TERM_TYPE	Last two digits of the Winter Term Code Required for SDS and TSDS
SUMMER	SICAS_TERM_TYPE	Last two digits of the Summer Term Code Required for SDS and TSDS
STANDARD	SICAS_TERM_TYPE	Identifying 'standard' terms – multiple entries Required for SDS and TSDS
NYS_FUNDED	SICAS_TERM_TYPE	** Community Colleges only ** Multiple entries identifying State Funded Winter and Summer terms Required for SDS and TSDS
CRSELVL_UG	SICAS_SDS	Level code for Undergraduate Students
CRSELVL_GR	SICAS_SDS	Level Code for Graduate Students

Figure 40 Table showing GTVSDAX crosswalks for SIRIS

Web Service Transmission

Internal Code: PROD_DB   Internal Group: SICAS_IR

Use the Crosswalk Validation Form GTVSDAX to enter the name of the institution's PRODUCTION database. This must be done before attempting any Web Services transmissions. The Internal Code is PROD_DB and the Internal Group is SICAS_IR. The External Code is the name of the production Banner database. The default setting for this entry is 'UPDATEME'. The External Code value must be changed to the Instance Name of the PRODUCTION database (i.e. PRODBAN) even on TEST databases. Do not alter this setting unless the name of the production Banner database is changed.

Figure 41 GTVSDAX – Crosswalk Validation Form showing the FIRST_TERM entry

Student Data Submission – SDS

The Student Data Submission (SDS) requires entries on the Crosswalk Validation Form GTVSDAX.

Internal Code: USA   Internal Group: ADDRESS

Use the Internal Code of USA and the Internal Group of ADDRESS to verify the External Code for the permanent address of U.S. students. The default setting for the External Code is 'UPDATEME'. Change the External Code to the Address Type on the Address Type Code Validation Form STVATYP that represents the permanent address for U.S. students.

An institution can create a hierarchy for Address Types by creating another GTVSDAX entry for Internal Code USA, Internal Group ADDRESS and assigning a value in the Sequence box for each Address Type. The Sequence number dictates the order the Address Types are processed.

Internal Code: FOREIGN   Internal Group: ADDRESS

Use the Internal Code of FOREIGN and the Internal Group of ADDRESS to verify the External Code for the permanent address of Foreign students. The default setting for the External Code is 'UPDATEME'. Change the External Code to the Address Type on the Address Type Code Validation Form STVATYP that represents the permanent address of Foreign students. The Foreign student Address Type may be the same as the U. S. student Address Type.

An institution can create a hierarchy for Address Types by creating another GTVSDAX entry for Internal Code FOREIGN, Internal Group ADDRESS and assigning a value in the Sequence box for each Address Type. The Sequence number dictates the order the Address Types are processed.

Internal Code: PELL_FUND   Internal Group: SICAS_SDS

Use the Internal Code PELL_FUND and the Internal Group SICAS_SDS to verify the External Code for the Pell Grant Fund Code. The default setting for the External Code is 'UPDATEME'. Change the External Code to the Fund Code on the Fund Base Data Form RFRBASE that represents the Fund Code for the Pell grant.

Internal Code: TAP_FUND   Internal Group: SICAS_SDS

Use the Internal Code TAP_FUND and the Internal Group SICAS_SDS to verify the External Code for the TAP Grant Fund Code. The default setting for the External Code is 'UPDATEME'. Change the External Code to the Fund Code on the Fund Base Data Form RFRBASE that represents the Fund Code for the TAP grant.

Internal Code: FALL   Internal Group: SICAS_TERM_TYPE
                 SPRING
                 SUMMER
                 WINTER

Use the Internal Code for the Term and the Internal Group SICAS_TERM_TYPE to verify the External Code that is the last two digits of the Term Code (do not include the year). Only define Terms that are normal processing terms—do not define terms used for non-credit course registration. Each Term requires a separate GTVSDAX entry.

Internal Code: STANDARD   Internal Group: SICAS_TERM_TYPE

Use the Internal Code of STANDARD and the Internal Group SICAS_TERM_TYPE to verify the External Code that is the last two digits of the Term Code (do not include the year) for each 'standard' term in use at the institution. This is usually two entries duplicating the FALL and SPRING entries; institution's using trimesters might have three entries. Each Term requires a separate GTVSDAX entry.

Internal Code: NYS_FUNDED   Internal Group: SICAS_TERM_TYPE

Only Community Colleges must define the GTVSDAX entry for Internal Code NYS_FUNDED.

Use the Internal Code of NYS_FUNDED and the Internal Group: SICAS_TERM_TYPE to verify the External Code that is the last two digits of the Term Code (do not include the year) for each Term, other than Fall or Spring, that is State Funded. Only enter Terms that are used at the institution. Each Term requires a separate GTVSDAX entry.

Term/Section Data Submission – TSDS

The Term/Section Data Submission (TSDS) requires entries on the Crosswalk Validation Form GTVSDAX.

GTVSDAX Form

Internal Code: FIRST_TERM   Internal Group: SICAS_SSBYSEC

The Internal Code FIRST_TERM and the Internal Group SICAS_SSBYSEC activates the data collection within the Schedule Form SSASECT. The External Code must be the first term used to populate the IR data submission elements on the SSASECT form. This prevents shadow record creation for historic records on the database. Data is not collected for Sections created before the first term indicated; this will save time and space for any campus with numerous historic records. Once entered, **DO NOT CHANGE** the FIRST_TERM value.

Once the External Code for the GTVSDAX entry Internal Code FIRST_TERM is entered, the ISSBYSEC script can be run to pre-populate the SICAS Section Shadow Table SSBYSEC for all CRNs within the initial term. See the ISSBYSEC.SQL section in this document for more information.

Internal Code: PSI_ACTIVE   Internal Group: SICAS_SLRRDEF

The second GTVSDAX entry for TSDS, Internal Code PSI_ACTIVE and Internal Group SICAS_SLRRDEF, enforces the definition of Banner Room Numbers and their crosswalk to PSI Room Numbers on the SICAS PSI Building and Room Validation Form STVYCRO. The default setting for this entry is 'UPDATEME'. It must be set to 'Y' or 'N' to use the Room Definition Form SLARDEF.

If a campus reports Physical Space Inventory (PSI) information to System Administration, the External Code for this entry should be set to 'Y'. When the External Code is set to 'Y', the Room Definition Form SLARDEF requires that the room number be selected from the Room Validation Form STVYCRO. All new rooms must be setup on STVYCRO before they can be accessed on SLARDEF. Use the CASA Code field on the Building Code Validation Form STVBLDG to crosswalk the PSI Codes for buildings.

Campuses that do not report PSI information to System Administration should set the External Code to 'N'. When the External Code is set to 'N', the SLARDEF form works as Baseline Banner and allows a room to be defined without it existing on STVYCRO.

Internal Code: FALL   Internal Group: SICAS_TERM_TYPE
                 SPRING
                 SUMMER
                 WINTER

Use the Internal Code for the Term and the Internal Group SICAS_TERM_TYPE to verify the External Code that is the last two digits of the Term Code (do not include the year). Only define Terms that are normal processing terms—do not define terms used for non-credit course registration. Each Term requires a separate GTVSDAX entry.

Internal Code: STANDARD   Internal Group: SICAS_TERM_TYPE

Use the Internal Code of STANDARD and the Internal Group SICAS_TERM_TYPE to verify the External Code that is the last two digits of the Term Code (do not include the year) for each 'standard' term in use at the institution. This is usually two entries duplicating the FALL and SPRING entries; institution's using trimesters might have three entries. Each Term requires a separate GTVSDAX entry. Summer and Winter Terms should not be coded as STANDARD.

Internal Code: NYS_FUNDED   Internal Group: SICAS_TERM_TYPE

Only Community Colleges must define the GTVSDAX entry for Internal Code NYS_FUNDED.

Use the Internal Code of NYS_FUNDED and the Internal Group: SICAS_TERM_TYPE to verify the External Code that is the last two digits of the Term Code (do not include the year) for each Term, other than Fall or Spring, that is State Funded. Only enter Terms that are used at the institution. Each Term requires a separate GTVSDAX entry.

SSBYSEC

The SSBYSEC program is used to populate the SIRIS shadow data on SSASECT. The shadow being referred to in this section is the SUNY IR data that exists in the SUNY IR blocks on SCACRSE and SSASECT. This program was originally developed to populate SSASECT prior to the initial run of any of the SIRIS processes, but it has been adapted to shadow data when a campus uses the Ellucian schedule roll program SSRROLL. SSBYSEC needs to be run immediately after the SSRROLL program because SSRROLL does not roll the SIRIS shadow data. Please remember that you must have the External Code for the GTVSDAX entry Internal Code FIRST_TERM and Internal Group SICAS_SSBYSEC entered for SSBYSEC to work.

SSBYSEC can be run from job submission. In order to protect data on existing records, this script will populate the shadow data when no shadow data exists. It is not possible to use SSBYSEC to update individual fields on the shadow table. For example, if you update the contact hours on SCACRSE you cannot use SSBYSEC to update individual sections if they have any shadow data already.

When running SSBYSEC you will receive three prompts:

1. Term to populate: This is the "TO" term. This is a required parameter. Enter the term you wish to add the shadow records to.
2. Roll Term: This is the "FROM" term. This is an optional parameter. Enter the term you wish to copy the shadow records from. The program will copy the shadow data from SSASECT for the term you enter. If the roll term is null, you are in the default term and the process will copy the shadow data from SCACRSE to the shadow table on SSASECT. If shadow data exists on SSASECT, the program will not update any shadow data for that section.
3. Run Type (A/U): Select "A" to run in audit mode or "U" to run in update mode.

Name	Required	Description	Values
Term to Populate	Y	This is the "To" term. Enter the term you wish to add the shadow records to.	Ex: 201210 (spring 2012)
Roll Term	N	This is the "From" term. Enter the term you wish to copy the shadow records from.	
Run Type	Y	This is the run mode.	A = Audit; U = Update

STVYIPD

College Codes:

The STVYIPD for was developed to serve as a crosswalk for College CEEB and FICE to IPEDS codes. To report a student's prior institution that institution must be entered on SOAPCOL and have the Prior Institution (SDS) box checked. The SDS process crosswalks the code from SOAPCOL to an IPEDS code using STVYIPD.

The codes used on SOAPCOL must match the corresponding code on STVYIPD. If your campus uses a six digit CEEB code (001234) on SOAPCOL and there is a four digit CEEB code on STVYIPD (1234), the crosswalk will not be completed and the IPEDS code will not be reported. This will result in a fatal error on the Prior Institution element. You will need to update the CEEB codes on SVTYIPD so they match the format of those on SOAPCOL.

The codes used on SOAPCOL must be present on STVYIPD or the crosswalk will not be completed and no IPEDS code will be reported. This will result in a fatal error. You will need to enter the CEEB code on STVYIPD.

Figure 42 STVYIPD form showing college code crosswalk.

Campuses are required to maintain the data on this form. It is possible to crosswalk many CEEB or FICE codes to one IPEDS code provided that on record is marked as the primary record. That is the recommended method to use when your campus uses a "dummy" CEEB for foreign Institutions. The valid System Administration code for a Foreign Institution is 990050. Figure provides an example of how to crosswalk multiple CEEB codes to one IPEDS code.

Figure 43 STVYIPD illustrating the multiple to one crosswalk

High School Codes:

System Administration requires that campuses report the High School for all new first time students. If the student attended graduated from High School in the United States, the SICAS selects the High School code directly from SOAHSCH and transmits that directly without cross walking. If the student graduated from a foreign High School the code from SOAHSCH must be cross walked on STVYIPD.

To crosswalk a High School code, insert a new record on the form.

1. Enter HS in the Code Field.
2. Enter a description in the Description Field
3. Enter the campus code for the foreign High School in the CEEB Field.
4. Enter the valid System Administration code for International High School in the Substitute CEEB for Reporting Field in the bottom block of the form. (Currently that code is 969999)
5. These steps may be repeated if your campus has multiple codes for foreign High Schools.

Figure provides an example of a setup for foreign High Schools.

Figure 44 STVYIPD Illustrating a setup for foreign High School Codes.

LOCAL CAMPUS MODIFICATION FUNCTIONS

SICAS provides Local Campus Modification Functions to allow an institution to affect the implementation of the CDS, SDS, TSDS, FADS and SRDS applications at their campus. All modifications made to a Local function are preserved during future releases. The local modifications referenced in this section are not the same as the local records that can be created on the DTI form. Please see the DTI form section for more information.

SICAS provides a template for Local Modifications, but technical staff at each institution is responsible for writing the code within the functions. SICAS ships Local Modifications to CREATE but not replace therefore if modifications are made to the function add 'OR REPLACE' after 'CREATE' at the top of the code. Refer to the comments in the local function code for further information.

The FZ_SDS_Admission_Status listed below is required and must be altered to return the student's Admission Status. SDS will not run properly without this local mod being altered.

Any modifications to a local mod point should be thoroughly tested on a campus test database before implementation in production.

CDS – Course Data Submission

FZ_CDS_BuildCourses (optional)

SICAS provides a Local Campus modification function for CDS called FZ_CDS_BuildCourses. A campus can use this mod point to affect the implementation of the routine for specific course records at their institution. Alter code within this function to override the SICAS CDS population routines. Any modifications to a local mod point should be thoroughly tested on a campus test database before implementation in production.

Implementation of this mod point is optional; the SICAS standard routine executes if this mod point is not changed. More information on using FZ_CDS_BuildCourses is available in Appendix E of this manual.

TSDS – Term/Section Data Submission

FZ_TSDS_BuildSection (optional)

SICAS provides a Local Campus modification function for TSDS called FZ_TSDS_BuildSection. A campus can use this mod point to affect the implementation of the routine for specific section records at their institution. Alter code within this function to override the SICAS TSDS population routines. Any modifications to a local mod point should be thoroughly tested on a campus test database before implementation in production.

Implementation of this mod point is optional; the SICAS standard routing executes if this mod point is not changed. More information on using FZ_TSDS_BuildSection is available in Appendix F of this manual.

FZ_TSDS_FACULTYHRS (optional)

SICAS provides a Local Campus modification for TSDS called FZ_TSDS_FACULTYHRS. A campus can use this mod point to calculate the number of hours a faculty member is in contact with students. This data is required for the Contact Hr element (TT110). A campus can elect to use this local mod point or enter the data in the contact hour field on SCACRSE and SSASECT. Failure to use one of these methods will result in the contact hours being submitted as zero (0).

FZ_TSDS_STUDENTHRS

SICAS provides a Local Campus modification for TSDS called FZ_TSDS_FACULTYHRS. A campus can use this mod point to calculate the number of hours a faculty member is in contact with students. This data is required for the Contact Hr element (TT120). A campus can elect to use this local mod point or enter the data in the contact hour field on SCACRSE and SSASECT. Failure to use one of these methods will result in contact hours being calculated using the same algorithm that was used in SDF (multiply weeks in the term, hours in each week and number of credits and divide by 50.)

SDS – Student Data Submission

SICAS provides three Local Campus modification functions for SDS. The FZ_SDS_Admission_Status is required and must be altered to return the student's Admission Status. The FZ_SDS_BuildStudents and FZ_SDS_InculdeStudent functions are optional.

More information on the Local Campus modification functions for SDS is available in Appendix G of this manual.

FZ_SDS_Admission_Status (required)

The SDS data element 'Adm Status' indicates an individual's Admission Status. Each institution determines Admission Status differently; therefore SICAS provides a Local Campus modification function that must be altered to return the correct System Administration Admission Status code.

Implementation of this mod point is required. If the FZ_SDS_Admission_Status. function is not altered locally by campus technical staff, the routine returns a 'Y' and the SICAS standard routine executes. The SICAS standard routine returns an invalid Admission Code of '999'.

The parameters passed in the FZ_SDS_Admission_Status function are:

%param Process_in   Process calling this routine
%param pidm_in   Student PIDM being considered for inclusion
%param term_in   Term Code to check

%return NUMBER   Return a supported System Administration Admission Status Code

FZ_SDS_PriorInst

This local mod point has been included for the Prior Institution element in SDS.   The SDS system was developed with the understanding that most campuses use CEEB or FICE code in the STVSBGI table to identify other Colleges. This local mod point should be used by Campuses that do not use FICE or CEEB in the STVSBGI table. It can also be used when campuses have developed local codes to identify Foreign Institutions. This mod point should be developed to convert the campus code to IPEDS Unit ID or to a System Administration value for Foreign Institutions. A listing of IPEDS Unit ID's is available in the STVYIPD table. Codes for Foreign Institutions can be obtained from the System Administration IR Office.

Any modifications to a local mod point should be thoroughly tested on a campus test database before implementation in production.

The following table lists the System Administration codes for Admission Status. This information is from the Student Data Elements Data Dictionary, February 12, 2008. The SUNY SIRIS Submission Data Dictionaries that define the data elements for each submission are available on SUNY's website at http://www.suny.edu/.

FADS – Financial Aid Data Submission

FZ_FADS_VETERANS_PROGID

The Local Modification function FZ_FADS_VETERANS_PROGID allows an institution to make local changes if they use the same detail code for veterans payments.

<table>
<tr><td colspan="3" align="center">Admission Status Codes</td></tr>
<tr><td align="center">Status</td><td align="center">Code</td><td align="center">Description</td></tr>
<tr><td>Regular Admit</td><td align="center">1</td><td>An individual who has matriculated in an academic program having met the institution's standard academic admissions criteria.</td></tr>
<tr><td>Special Admit Receiving Aid</td><td align="center">2</td><td>An individual receiving benefits from any of the following programs: Educational Opportunity Program (EOP). Search for Education, Evaluation, and Knowledge (SEEK), and The Graduate Educational Opportunity Program (GEOP)</td></tr>
<tr><td>Special Admit Disadvantaged</td><td align="center">3</td><td>An individual eligible for EOP, SEEK, or GEOP but not receiving benefits.</td></tr>
<tr><td>Other Special Admit</td><td align="center">4</td><td>An individual not eligible for EOP, SEEK, or GEOP who is admitted based on admissions criteria that differs from the academic standards applied to most incoming students. This would include risk admissions which are students who fail to meet the normal admissions standards of the college, or students admitted in recognition of a special talent that merits admission to an academic program related to this talent which is offered by the institution.</td></tr>
<tr><td>Non-Degree Seeking</td><td align="center">5</td><td>An individual who has not matriculated at the college and is not a cross-registrant from another college, but is enrolled in at least one credit course (or non-credit course eligible for state aid).</td></tr>
<tr><td>Continuing Education</td><td align="center">6</td><td>An individual enrolling exclusively in continuing education courses.</td></tr>
<tr><td>Cross Registered – Accessory</td><td align="center">7</td><td>A student from the affiliate campus who is attending classes at the reporting SUNY campus under an accessory instruction agreement/contract. Code 7 can only be used if the reporting campus (Element ID SH 010) is Cornell, Alfred Ceramics, or ESF.</td></tr>
<tr><td>Cross Registered-Another SUNY</td><td align="center">8</td><td>A cross-registered student from another SUNY college who is attending classes at the reporting campus under a cross registration agreement.</td></tr>
<tr><td>Cross Registered-Non-SUNY</td><td align="center">9</td><td>A cross-registered student from another Non-SUNY college who is attending classes at the reporting campus under a cross registration agreement.</td></tr>
<tr><td>Exchange Student From Abroad</td><td align="center">10</td><td>A foreign student who is attending classes at a SUNY institution as part of a formal exchange program.</td></tr>
<tr><td>College in the High School</td><td align="center">11</td><td>For state-operated institutions, a high school student taking one or more college level courses taught in his or her high school with curricula oversight and credit transcript provided by the reporting</td></tr>
</table>

Admission Status Codes		
Status	**Code**	**Description**
		SUNY institution. (Community colleges should continue to report these students as Admission Status 1-5.)
Non-Resident On-Line	12	A non-resident student exclusively taking courses that are taught outside of New York State and/or the United States.
Auditor	13	A student enrolled exclusively as an auditor. Use Special Population code (SE025) to specifically identify whether the student is a Special Auditor or State-Aidable Auditor.
EOS	14	A student enrolled exclusively at a recognized SUNY Educational Opportunity Center. SUNY universities and colleges are not to use this code.

If Higher Education History (Element ID SE 020) is Cross-Registered (9), Admission Status must be Cross Registered – Accessory (7). Cross Registered – Another SUNY (8), or Cross Registered – Non-SUNY (9). Admissions Status of 7 can only be used by Cornell, Alfred Ceramics or ESF.

FZ_SDS_BuildStudents (optional)

The Local Modification function FZ_SDS_BuildStudents allows an institution to modify the student population globally or on a student-by-student basis at different times during the SICAS Student Data Submission Process SGRYSDS. This function is called before the SGRYSDS population routine is executed and again afterwards.

> BEFORE PASSES – These are used to implement a campus routine to pick up records to include in the SDS data transmission. Return 'Y' to continue with the SICAS standard population selection; the campus routine will supplement the SICAS selected records. Return 'N' if the campus does not want to run the SICAS standard population selection; the campus routine will replace the SICAS routine.

> AFTER PASSES – These are used to delete SICAS selected records, alter the student data created, or to supplement the SICAS population selections with additional records. The STUDENT_LOOP call provides a single student PIDM for the updates.

Any modifications to a local mod point should be thoroughly tested on a campus test database before implementation in production.

FZ_SDS_IncludeStudent (optional)

The Local Modification function FZ_SDS_IncludeStudent allows an institution to decide if a student should be included or excluded from the population selection based on criteria created by the campus rather than the criteria used by the standard SDS processing.

SRDS Student Revenue Data Submission

Prior Academic Year Reporting

SRDS code assumes that the 2011-12 academic year is encoded as 2011. If a school interprets this as 2012, then a local mod must be written. SRDS logic is therefore sending 2012-13 for current year transactions (and advancing all transactions forward one academic year).

Confirm that the above analysis is correct then process only if it is.

1. Go to TOAYDTI and look up the "Acad Year" element. With that record selected use the Record Menu to "Insert" and then "Duplicate" (both are options on the Record Menu). On the new record, the only change you will need to make is to the "Select Code." The new record you created will be checked as a Local Mod and MAY NOT BE ALTERED.

2. Change the "Select Code" box so that it contains the following code:
    a. select to_char(substr(to_number(stvterm_acyr_code-1),1,4))||'-'||to_char(substr(to_number(stvterm_acyr_code),3,2))
       from stvterm
       where stvterm_code = :torysrd_tran_term_code
3. Commit the change.

4. Re-run the process using TR mode for the period and all of the Academic Year entries will move back one year to their correct value.

Academic Term (RC045) – Summer as First Term in Academic Year Option Format: C(1)

The academic term for which the revenue charge (RC020 – RC070) was made. Enter 1 for the summer in the first calendar year of the Academic Year (RC042), and 5 for the summer in the second calendar year. For example, 1 for 2009-10 means summer 2009 and 5 means summer 2010. Required

IMPORTANT NOTE: This record is used by campuses that have the Summer as the FIRST term in the Academic Year. Just create a Local Mod Record and set the Process Method to "Use Select Code." Other campuses should leave this record INACTIVE since only one of these two items can be active at one time. The Web Service Setting for Alternate XML Code Specification decodes SUMMER to 1 when this record is Active.

Codes:

1   Summer   Summer leads other terms, appears in year one of AcademicYear (RC042).
2   Fall   Fall
3   Winter   Winter
4   Spring   Spring
5   Summer II Summer follows other terms, appears in year two of Academic Year
             (RC042).
6   Unknown   Academic term is unknown

Error Level	**Description**
Fatal	Academic Term is required for each Revenue Charge Amount (RC040) submitted.
Fatal	Academic Term valid values are 1-6.

Academic Term (RC045)   Summer as Last Term in Academic Year Option
                              Format: C(1)

The academic term for which the revenue charge (RC020-Rc070) was made. Enter 1 for the summer in the first calendar year of the Academic Year (RC042), and 5 for the summer in the

second calendar year. For example, 1 for 2009-10 means summer 2009 and 5 means summer 2010. Required.

IMPORTANT NOTE: This record is used by campuses that have the summer as the LAST term in the Academic Year. Just create a Local Mod Record and set the Process Method to "Use Select Code." Other campuses should leave this record INACTIVE since only one of these two items can be active at one time. The Web Service Setting for Alternate XM: Coe specification decodes SUMMER to 5 when this record is active.

Codes:
  1   Summer   Summer leads other terms, appears in year one of Academic Year (RC042).
  2   Fall   Fall
  3   Winter   Winter
  4   Spring   Spring
  5   Summer II   Summer follows other terms, appears in year two of Academic Year (RC042).
  6   Unknown   Academic term is unknown

Error Level   Description
Fatal   Academic Term is required for each Revenue Charge Amount (RC040) submitted.
Fatal   Academic Term valid values are 1-6.

URAS Code (RC032)
*****This is a special record to correct accounting errors at time of feed due to changed GL accounts on page 2 of TSADETC.

TOAYDTI:

Interface Type: SRDS
  • Column Definition Tab
Column Name Column
Go to URAS Code
Record Insert
Record Duplicate
Rename the duplicate record to URAS Code2
Make Process Order 301
Process Method column. Choose "Use Select Code"

  • Element Details Tab: add the following statement in the text box:

  • Select Code Tab: to delete the existing account and replace with a correct one. XXXX=the correct URAS account. Add this in the text box:

    SELECT XXXX FROM DUAL

  • Update Condition Tab: Copy the target column code TORYSRD_URAS_CODE and make it = to the wrong URAS account (XXXX) Add this to the text box:

    TORYSRD_URAS_CODE=XXXX

  • Web Service Settings Tab: In the XML Tag Definitions box DELETE whatever is in that Box, which should be "urasCode". Audit Trail Tab: add the following in the text box with date of your mod:

  • Save and then re-run TSRYSRD.

@WHERE

The @WHERE on TOAYDTI will allow you to ignore transactions in the FEE category if a school has charge transactions for non-students in this category. Reportable students and their charges will all still get reported due to the other population selection criteria built into SRDS. They will all get reported by having their charges fall into the TUI, HOU or MEA categories as well as being reported in the main student repository. Insert No in the FEE section on TOAYDTI.
Save and rerun in TR mode.
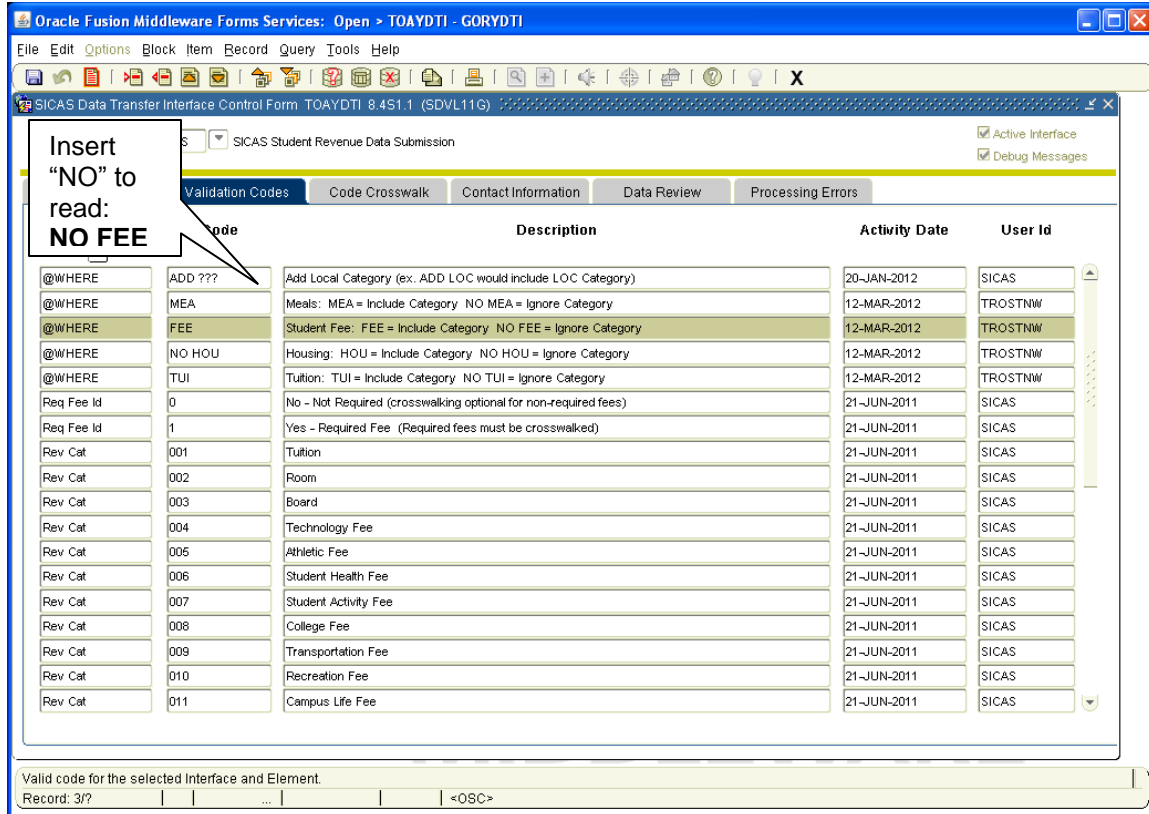
Figure 45 SICAS Data Transfer Interface Control Form TOAYDTI

Any modifications to a local mod point should be thoroughly tested on a campus test database before implementation in production.

PROCESS FLOW

The IR Submission processes (SICAS Course Data Submission Report—SCRYCDS, SICAS Student Data Submission Report—SGRYSDS, SICAS Term Section Data Submission Report—SSRYTSD, and SICAS Financial Aid Data Submission Report- RORYFAD, Student Revenue Data Submission Report - TSRYSRD) collect data for export to System Administration. The data are transmitted using a System Administration Web Service. All Run Modes require the Proxy Client to be configured and up and running so that contact can be made with the System Administration system to establish the status of System Administration's DTS System before executing. Errors and Warnings are returned during data transmission, therefore Audit Modes do not generate error records—Audit Modes (AR and AU) gather data but do not transmit. All data-specific errors generated during transmission are listed on the System Administration Web application and can be viewed on the Processing Errors Tab of the SICAS Data Transfer

Interface Control Form SOAYDTI. Threshold Errors cannot be viewed in Banner; they must be reviewed on the System Administration website. A batch cannot be locked if the Threshold Errors are too high. To maintain synchronization, errors must be corrected in Banner and not with the file correction tools on the System Administration website.

Once all errors are resolved, transmit the corrected data to System Administration, and 'lock' the file using the System Administration website application (refer to the System Administration website for more information on locking a file). Only IR staff at System Administration can unlock a file. Once the file is 'approved' by IR staff, the data are moved from the staging area to the Data Warehouse. Approved batches must be posted in Banner by running the appropriate program in PO mode.   Once a data submission is 'approved', it is final. System Administration does not have a way to 'unapprove' a submission. However, the Course Data Submissions (CDS) can be Unposted locally, corrections made, transmitted, and re-locked for Approval on the System Administration Website. In this case, a CDS must be Re-Locked, Re-Approved, and Re-Posted before Term/Section Data Submission (TSDS) can be transmitted. CDS can only be Unposted before the Term/Section Data Submission (TSDS) is Approved; once TSDS is Approved, all submissions for that term are final.

Terms MUST be processed in order for each submission. For example, if you POST your winter EOT, you will be prevented from submitting an EOT file for the previous fall semester. This is true for all of the submissions except FADS which is not term based.

When an IR submission process is run in Job Submission, only two parameters are required for CDS and TSDS—TERM CODE and RUN MODE. FADS has two parameters, Aid Year Code and Run Mode. SRDS has three parameters, Reporting Year, Reporting Month and Run Mode. SDS has a third parameter Run Period. For CDS, TERM CODE is the Course Catalog Term to collect. For TSDS, TERM CODE is the Class Schedule Term to collect. For SDS, TERM CODE is the term being processed. The TERM CODE must be greater than the last POSTED term. See the Process Parameters section for the SICAS Student Data Submission Report SGRYSDS for an explanation of the Run Period parameter. You must submit your Early Student Submission for the Fall and Spring terms before the End of Term submission.

RUN MODE includes Audit, Transmission, and Batch Codes. Audit Modes AR (Audit Replace) and AU (Audit Update) gather data but do not transmit. The Transmit Modes TR (Transmit Replace) and TU (Transmit Update) gather and transmit data and the Transmit Mode TO only transmits data.

Batch Modes manage the Batch of records as a whole—records are not collected or transmitted.

Delete Unposted Batch (DB)	Deletes an un-posted batch if it is 'Open'
Post Unposted Batch (PO)	Posts an existing batch if it is 'Approved'
Unpost and Replace (UR)	Unpost Approved Batch, Replace All, for SCRYCDS only. Does not transmit
Unpost and Update (UU)	Unpost Approved Batch, Update Batch, for SCRYCDS only. Does not transmit

In Job Submission, RUN MODE is a two-character code entered as a single parameter. Pick the Values pull-down list to display the valid entries. Valid Run Modes are listed in the Parameters section. The RUN MODE and BATCH ACTION parameters are entered separately when the process is run from the Command Line prompt. These two parameters are equivalent to the single RUN MODE parameter used in Job Submission.

If an IR Submission process run in a Transmission Mode terminates before completion, the batch does not finalize. When this happens, you will receive the following error message "SA:INI NO NEW BATCH CREATED- ONLY ONE BATCH CAN BE PROCESSED AT A TIME RESULT=FAILURE." The SICAS DTS Batch Finalization Process SORYDTS will finalize a

'stranded' batch. For more information on using this process, see the SORYDTS SICAS DTS Batch Finalization Process section in this document.

SORYDTS   SICAS DTS Batch Finalization Process

The IR Submission processes (SICAS Course Data Submission Report—SCRYCDS, SICAS Student Data Submission Report—SGRYSDS, and SICAS Term/Section Data Submission Report—SSRYTSD) create batches. If an IR Submission process starts successfully in a Transmission Mode (TR or TU) and initiates a batch, but the process fails to run to completion, the batch will not be finalized.

This can happen if a catastrophic error occurs such as the loss of Internet access, the process is terminated by the user, or there is a problem with the System Administration database or web application. The SICAS DTS Batch Finalization Process SORYDTS will force the finalization of a batch 'stranded' on the System Administration database. A batch finalized by the SORYDTS process is incomplete and must be re-transmitted using either the Transmit Replace Mode (TR) or the Transmit Update Mode (TU) to correct the data at System Administration.

The SORYDTS process creates a Report Control Information page. For an example, see Figure in the Sample Output section.

Refer to the Process Parameter section for a detailed list of parameters for the SICAS DTS Batch Finalization Process SORYDTS.

The SORYDTS process requires the Batch Number of the 'stranded' batch. When this happens, you will receive the following error message "SA:INI NOW NEW BATCH CREATED- ONLY ONE BATCH CAN BE PROCESSED AT A TIME RESULT=FAILURE." This is the last Batch Number on the Submission History Listing found on the SUNY System Administration DTS web application.

Finding the Batch Number at System Administration

The SICAS DTS Batch Finalization Process SORYDTS requires a parameter entry (Batch Number) to identify the batch to finalize. This is the last batch number on the Submission History Listing for the Submission Type that failed to complete successfully.

To find the Batch Number:

1. Log on the SUNY DTS Web Application (for Test Database go to: https://test.suny.edu; for Production Database go to: https://www.suny.edu)

2. Select the Data Transfer System (DTS) link for the Submission Type that failed to complete successfully- Course, Student, or Section (Figure 44)

3, Select the 'View History' option from the 'Select An Action' Drop Down list and Pick 'Go' (Figure 45)

4. Record the last Batch Number in the list – the first column is the Batch Number – enter it as the Batch Number parameter for the SORYDTS process (Figure 46)

Figure 46 SUNY DTS site

Figure 47 SUNY DTS site showing "View History"
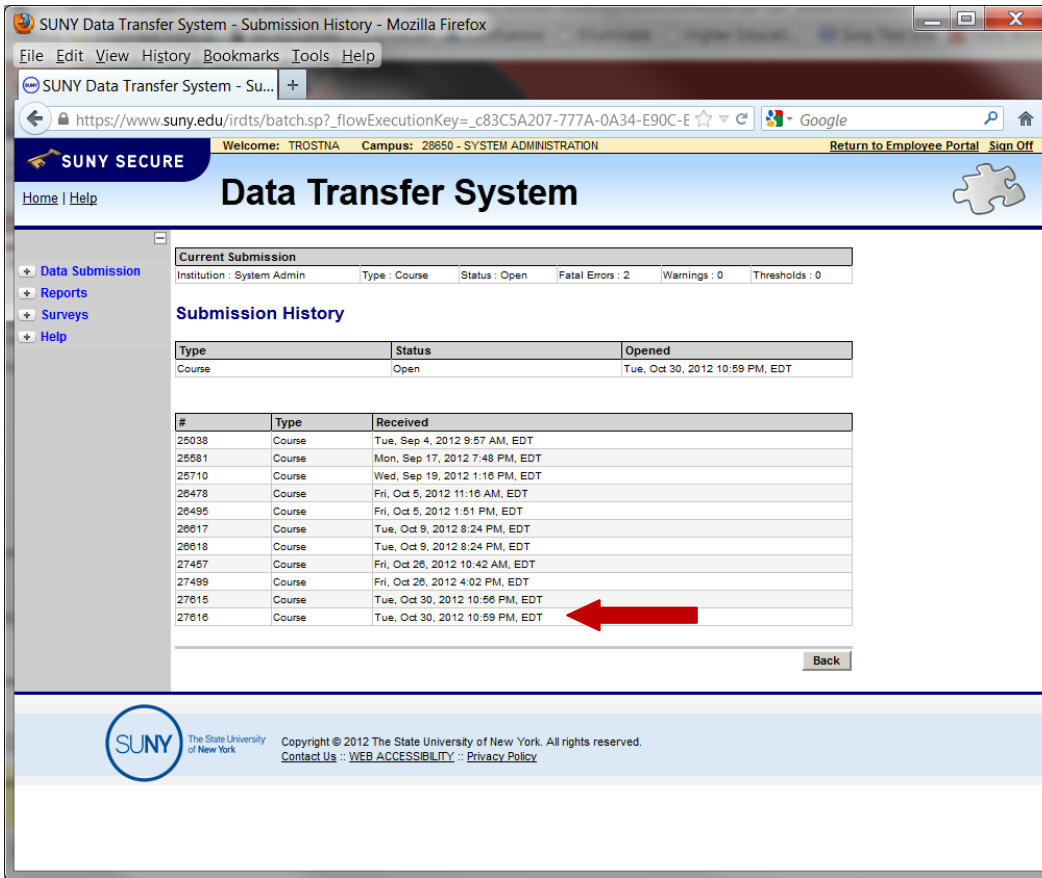
Figure 48 SUNY DTS Site-Record the last Batch

CDS Process Flow

CDS Process Flow Overview

1. Run the SICAS Course Data Submission Report SCRYCDS in Audit Mode

   a. Use the Audit Replace Mode (AR) the first term—this build/replaces all course data
   b. Use the Audit Update Mode (AU) after the first term—only builds changes
   c. Occasionally, it may be desirable to build a full replacement using the Audit Replace Mode

   The Audit Replace Mode collects all course data with an effective term less than or equal to the term being processed

2. Review and Confirm the resulting Course Data using the Data Review tab on the SOAYDTI form. Make corrections as needed.

3. Run SCRYCDS in Transmit Mode to send data to the SUNY Data Transfer System (DTS). For the initial transmission of data, use the same action (Replace or Update) used in Step 1.

4. Review Errors on the Processing Errors tab of the SOAYDTI form. Make corrections as needed. To maintain synchronization, errors must be corrected in Banner and not with the file correction tools on the System Administration website.

5. Use Transmit Update Mode (TU) to send corrected data; repeat until all errors are resolved. Using Update Mode speeds the processing since successful courses are not resent; however, please be aware that new records will also be sent.

6. Use the SUNY System Administration Web page to review and Lock the batch for Approval.

7. If necessary, resolve any problems identified by SUNY IR staff. Re-Transmit and Re-Lock the batch.

8. After SUNY IR staff Approves the batch, run SCRYCDS in Post Batch Mode (PO) to update the Data Posted flag on the SICAS Course Data Submission extract tables.

SCRYCDS   SICAS Course Data Submission Report

The SICAS Course Data Submission Report SCRYCDS collects course data for export to System Administration and generates four reports. The data are transmitted using a System Administration Web Service. If the SCRYCDS process starts successfully in a Transmission Mode and initiates a batch, but the process fails to run to completion, the batch will not be finalized. When this happens, you will receive the following error message "SA:INI NO NEW BATCH CREATED- ONLY ONE BATCH CAN BE PROCESSED AT A TIME RESULT= FAILURE."

If this occurs, use the SICAS DTS Batch Finalization Process SORYDTS to force the finalization of a batch 'stranded' on the System Administration database. A batch Finalized by the SORYDTS process is incomplete and must be retransmitted. For more information on using this process, see the SORYDTS SICAS DTS Batch Finalization Process section in this document.

There are two related tables that store course data: the SICAS Course Data Extract Table SOBYCDS, which contains a single entry for each course, and the SICAS Equivalent Course Extract Table SORYCDS, which is a repeating table containing 0, 1, or multiple Equivalent Course entries.

The SCRYCDS process executes the PY_CDS_PROCESS procedure within the SICAS Data Collection and Crosswalk Interface Package SFKYCVTS. This creates the course population in the SOBYCDS table and then uses the BASE elements to populate additional columns. Once this is completed, the routine populates the SORYCDS table with Equivalent Courses and then uses the RPTG defined elements to populate additional columns in the table. Elements are defined as BASE or RPTG in the Type field on the lower portion of the Column Definitions Tab on the SOAYDTI form. BASE elements are always processed before the RPTG elements.

SDS Process Flow

SDS Process Flow Overview

1. Run the SICAS Data Submission Report SGRYSDS in Audit Mode

     a. Use the Audit Replace Mode (AR) the first time run for any given term
     b. Use the Audit Update Mode (AU) thereafter – only builds changes

   For the fall and spring terms, the SGRYSDS process must be run for an Early Student Submission (ESS) before the End of Term (EOT) submission. SDS is not run in ESS mode for winter and summer terms.

2. Review and Confirm the resulting Student Data using the Data review tab on the SOAYDTI form. Make corrections as needed.

3. Run SGRYSDS in Transmit Mode to send data to the SUNY Data Warehouse staging area. For the initial transmission of data, use the same action (Replace or Update) used in Step 1.

5. Review Errors on the Processing Errors tab of the SOAYDTI form. Make corrections as needed. To maintain synchronization, errors must be corrected in Banner and not with the file corrections tools on the System Administration website

5. Use Transmit Update Mode (TU) to send corrected data; repeat until all errors are resolved. Using Update Mode speeds the processing because only updates are sent; however, please be aware that new records will also be sent – this includes new registrations.

6. Use the SUNY System Administration Web page to review and lock the batch for approval.

7. If necessary, resolve any problems identified by SUNY IR staff. Re-transmit and re-lock the batch.

8. After SUNY IR staff approves the batch, run SGRYSDS in PO (Post Batch) mode to update the Data Posted flag on the SICAS Student Data Submission extract tables.

SGRYSDS   SICAS Student Data Submission Report

The SICAS Student Data Submission Report SGRYSDS collects student data for export to System Administration and generates reports that are accessible on the data review tab of SOAYDTI. The data are transmitted using a System Administration Web Service. If the SGRYSDS process starts successfully in a transmission mode and initiates a batch, but the process fails to run to completion, the batch will not be finalized. If this occurs, use the SICAS DTS Batch Finalization Process SORYDTS to force the finalization of a batch 'stranded' on the System Administration database. A batch Finalized by the SORYDTS process is incomplete and must be retransmitted. For more information on using this process, see the SORYDTS SICAS DTS Batch Finalization Process section in this document.

In fall and spring terms, the SGRYSDS process must be run in ESS before the EOT submission can be sent. It is recommended that the ESS be completed (locked, approved and posted) before the EOT is run.

The student population selected for the Student Data Submission is determined by the student's enrollment. The student must be enrolled in at least one section that meets the following criteria:

- The Section has an Action status of 'Ready' (Ready to Submit) on the SUNY IR Submissions Detail block of the Schedule Form SSASECT

- The Course associated with the Section has an Action status of 'Ready' (Ready to Submit) in the SUNY Institutional Research Data block of the Basic Course Information Form SCACRSE

- The student's Registration Status Code for the Section is marked to count for SDS on the Course Registration Status Code Validation Form STVRSTS

- If the SGRYSDS process is being run for the Early Student Submission, all students who are enrolled in at least one course with a Census date on or before October 15 (Fall) or March 15 (Spring) are included. If the student withdrew from the course after the section census date he/she will also be included. Students who are included in the ESS file will have all of their credit hours reported. For example, a student enrolled in one section with a census date before October 15 and four sections with a census date after October 15 will have the credits from all five sections reported.

- The Section's Census Date must be less than or equal to October 15 (Fall) or March 15 (Spring). The Section Census Date is the Census One Freeze Date field in the Enrollment Data block of the Schedule Form SSASECT. This date can be edited for individual sections on the Section Calendar form SSAACCL. The Section Census Date is used for the FINAL End of Term submission when a student has withdrawn from a section. If the student withdrew after the census date, that section will be counted in the student's enrollment.

There are two tables that store student data for the term being processed for either the ESS or EOT run: the SICAS Student Submission Base Table SOBYSDS which contains the base elements for each student, and the SICAS Student Submission Repeating Data Table SORYSDS

which contains repeating elements (Admissions/Placement Criteria, Disability Status, and Race Codes) for each student.

SICAS provides three Local Campus Modification Functions that can be used to control the student population selected for SDS submission. The FZ_SDS_Admission_Status function is required and must be altered to return the student's Admission Status. Using the FZ_SDS_IncludeStudent or the FZ_SDS_BuildStudent function is optional. For more information on these functions, see the Local Campus Modification Functions section in this document.

See the Process Parameter section for a detailed list of parameters for the SICAS Student Data Submission Report SGRYSDS.

TSDS Process Flow

TSDS Process Flow Overview

1. Complete the CDS data submission. CDS data must be locked, approved, and posted prior to running SSRYTSD because Term/Section Data depends on the existence of Course Catalog and Student data.

2. Run the SDS End of Term (EOT) submission. It will be helpful if SDS is nearly error free before running TSDS. The EOT submission and TSDS submission can be finalized simultaneously. If students are missing from the data warehouse at System Administration, you will need to resubmit your EOT file to include them.

3. Run the SICAS Term/Section Data Submission Report SSRYTSD in Audit Mode

    a. Use the Audit Replace Mode (AR) the first time run for any given term
    b. Use the Audit Update Mode (AU) thereafter – only builds changes

4. Review and confirm the resulting data using the Data Review tab of the SOAYDTI form. Make corrections as needed.

5. If an error in the Course Catalog is discovered, resubmit the CDS file so the missing data is included. The data must be locked, approved and posted before the errors are resolved.

6. Run SSRYTSD in Transmit Mode to send data to the SUNY Data Warehouse staging area. For the initial transmission of data, use the same action (Replace or Update) used in Step 2.

7. Review Errors on the Processing Errors tab of the SOAYDTI form. Make corrections as needed. To maintain synchronization, errors must be corrected in Banner and not with the file correction tools on the System Administration website.

8. Use Transmit Update Mode (TU) to send corrected data; repeat until all errors are resolved. Using Update Mode speeds the processing because only updates are sent; however, please be aware that new records will also be sent. This includes new registrations.

9. Use the System Administration Web page to review and Lock the batch for approval.

10. If necessary, resolve any problems identified by SUNY IR staff. Re-transmit and re-lock the batch.

11. After SUNY IR staff approves the batch, run SSRYTSD in Post Batch Mode (PO) to update the Data Posted flag on the SICAS Term/Section Data Submission extract tables

SSRYTSD   SICAS Term Section Data Submission Report

The SICAS Term Section Data Submission Report SSRYTSD collects student data for export to System Administration and generates reports. The data are transmitted using a System Administration Web Service. If the SSRYTSD process starts successfully in a transmission mode and initiates a batch, but the process fails to run to completion, the batch will not be Finalized. If this occurs, use the SICAS DTS Batch Finalization Process SORYDTS to force the finalization of a batch 'stranded' on the System Administration database. A batch Finalized by the SORYDTS process is incomplete and must be retransmitted. For more information on using this process, see the SORYDTS SICAS DTS Batch Finalization Process section in this document.

The general purpose Web Service library package includes a TSDS specific XML generation package SFKYTSDS. SSRYTSD uses the SICAS Data Collection and Crosswalk Interface Package SFKYCVTS. The SFKYCVIR package adds TSDS specific calls to the toolset. The data are transmitted using a System Administration Web Service.

There are several SICAS tables that store Term Section data:

> SICAS Section Data Submission Table SOBYSEC – Base Section Data
> SICAS Section Data Submission Meeting Table SORYMET – Section Meetings Data
> SICAS Section Data Submission Meeting Table SORYMST – Section Meetings Data for
>   Sections meeting at the same time
> SICAS Section Student Base Table SOBYSTU – Student Data
> SICAS Section Student Repeating Table SORYSTU – Student Data
> SICAS Section Faculty Base Table SOBYINS – Instructor Data
> SICAS Section Faculty Repeating Effort Table SORYINS – Instructor Data

The SSRYTSD process uses six types to separate the elements into groups. Elements are assigned a 'type' in the 'Type' field on the lower portion of the Column Definitions Tab on the SOAYDTI form. They are processed in the order listed:

> SECT   updates SOBYSEC – the SICAS Section Data Submission Table
> MEET   updates SORYMET – the SICAS Section Data Submission Meeting Table
> SMLT   updates SORTMST – the SICAS Section Data Submission Meeting Table for
>   sections meeting at the same time
> BINS   updates SOBYINS – the SICAS Section Faculty Base Table
> RINS   updates SORYINS – the SICAS Section Faculty Repeating Effort Table
> BSTU   updates SOBYSTU – the SICAS Section Student Base Table
> RSTU   updates SORYSTU – the SICAS Section Student Repeating Table

See the Process Parameter section for a detailed list of parameters for the SICAS Term Section Data Submission Report SSRYTSD.

PROCESS PARAMETERS
SICAS Course Data Submission Report SCRYCDS
SICAS Term Section Data Submission Report SSRYTSD
SICAS Student Data Submission SGRYSDS

RUN SEQUENCE NUMBER (Optional) - Number (command line only)
   The job submission sequence number stored in GJBPRUN for the job.
   If a sequence number is entered, the parameters will be selected
   from GJBPRUN, otherwise the user will be prompted for the remaining parameters.

OUTPUT DIRECTORY – Character (Default: SYS$LOGIN) (command line only)
   The directory location where the reports will be created.

TERM CODE (Required) – Character (6 characters required)
   The Term Code must be greater than the last Posted term.
        For CDS, it is the Term Code for the Course Catalog to send. This includes records that
        are effective on or before this term.
        For TSDS, this is the Term Code for the Class Schedule to send.
        For SDS this is the Term Code you wish to send.

RUN PERIOD This is an SDS parameter only.
        For SDS enter ESS for the Early Student Submission and EOT for the End of Term
        Submission.

RUN MODE [A=Audit; T=Transmit; P=Post Batch; D=Delete Batch; R=Report] (Default: A)
   In Job Submission, RUN MODE is a two-character code entered as a single parameter.
   If using Command Line, the two parameters RUN MODE and BATCH ACTION provide similar
   functionality to the single RUN MODE parameter used in Job Submission.

   Valid Run Modes for Job Submission are:
        AU = Audit run to Add Updates to Batch (default mode)
        AR = Audit run to Replace Batch
        TU = Add Updates to Batch and Transmit
        TR = Replace Batch and Transmit
        TO = Transmit Batch Only
        PO = Post Unposted Batch
        DB = Delete Unposted Batch
        UR = Unpost Approved Batch and Replace All– for SCRYCDS only
        UU = Unpost Approved Batch and Update Batch– for SCRYCDS only

BATCH ACTION (command line only)
   If the RUN MODE = A (Audit) or T (Transmit), the prompt is [R=Replace; U=Update;
   T=Transmit Only] – the default is U (Update)

   If the RUN MODE = R (Report), the prompt is [P=Posted; U=Unposted] – the default is U
    (Update)

   If the RUN MODE = D (Delete Batch) or P (Post Batch), there is no prompt for the BATCH
    ACTION parameter

   When the RUN MODE = T (Transmit) and the BATCH ACTION = T (Transmit Only), this
    translates to TO (Transmit Only)

LINES PER PAGE - Number (Default: 60)
   The number of lines to print on each page.

EXECUTE REPORT – Character (Default: Y) (command line only)
   Execute the process – Y = Yes; N = No

SICAS Financial Aid Data Submission RORYFAD
Process Parameters

AID YEAR Code (Required)-
The Financial Aid Year that is being submitted

RUN MODE [A=Audit; T=Transmit; P=Post Batch; D=Delete Batch; R=Report] (Default: A)
   In Job Submission, RUN MODE is a two-character code entered as a single parameter.
   If using Command Line, the two parameters RUN MODE and BATCH ACTION provide similar
   functionality to the single RUN MODE parameter used in Job Submission.

Valid Run Modes for Job Submission are:
        AU = Audit run to Add Updates to Batch (default mode)
        AR = Audit run to Replace Batch
        TU = Add Updates to Batch and Transmit
        TR = Replace Batch and Transmit
        TO = Transmit Batch Only
        PO = Post Unposted Batch
        DB = Delete Unposted Batch

SICAS Student Revenue Data Submission TSRYSRD
Process Parameters

ACADEMIC YEAR (Required) – Character (four characters required) The academic year for the
submission being requested.
REPORTING MONTH (Required) – Character (three characters required. The reporting month
for the academic submission. DEC for period July 1 through December 31; JUN for January 1
through June 30.
RUN MODE [A=Audit; T=Transmit; P=Post Batch; D=Delete Batch; R=Report] (Default: A)
   In Job Submission, RUN MODE is a two-character code entered as a single parameter.
   If using Command Line, the two parameters RUN MODE and BATCH ACTION provide similar
   functionality to the single RUN MODE parameter used in Job Submission.

Valid Run Modes for Job Submission are:
        AU = Audit run to Add Updates to Batch (default mode)
        AR = Audit run to Replace Batch
        TU = Add Updates to Batch and Transmit
        TR = Replace Batch and Transmit
        TO = Transmit Batch Only
        PO = Post Unposted Batch
        DB = Delete Unposted Batch

SICAS DTS Batch Finalization Process SORYDTS
Process Parameters

RUN SEQUENCE NUMBER (Optional) - Number (command line only)
    The job submission sequence number stored in GJBPRUN for the job.
    If a sequence number is entered, the parameters will be selected
    from GJBPRUN, otherwise the user will be prompted for the remaining parameters.

OUTPUT DIRECTORY – Character (Default: SYS$LOGIN) (command line only)
    The directory location where the reports will be created.

IR SUBMISSION CODE (Required) – Character (4 character maximum)
    This is the type of submission that has a stranded batch to finalize.

CDS	Course Data Submission (C is used on the Command Line)
TSDS	Term Section Data Submission (T is used on the Command Line)
SDS	Student Data Submission (S is used on the Command Line)
FADS	Financial Aid Data Submission
SRDS	Student Revenue Data Submission

TERM OF THE BATCH (Required) – Character (6 characters required)
    The Term Code for the Batch being processed when the failure occurred.

BATCH NUMBER (required) – Numeric
    Last Batch number on the SUNY Web App Submission's View History listing.

LINES PER PAGE - Number (Default: 60)
    The number of lines to print on each page.

EXECUTE REPORT – Character (Default: Y) (command line only)
    Execute the process – Y = Yes; N = No

SORYDTS   SICAS DTS Batch Finalization Process

The SICAS DTS Batch Finalization Process SORYDTS only creates a Control Page. If the Finalization Result is FAILURE, query the Processing Errors tab on the SICAS Data Transfer Interface Control Form SOAYDTI. Enter the Interface Type for the failed Batch (CDS, SDS, or TSDS) in the Key Block of SOAYDTI then navigate to the Processing Errors tab and enter the Process Name SORYDTS and the appropriate Term. More complete error information will be displayed.

```
29-AUG-2008 01:43:01 PM              SICAS CENTER              SORYDTS 7.4S2.0
                         SICAS Abandon DTS Batch Utility Process



                         * * * Report Control Information * * *



Interactive/Batch............. I  Interactive
Program Name.................. SORYDTS.SQR
Release....................... 7.4S2.0
Package Version............... 7.4S2.2
Job Submission Number.........
Output Directory.............. DISK$SICAS:[TROSTNW.SQR.CDS]
Submission Code............... SDS
Term Code..................... 200720
Batch Number to Finalize...... 233
Lines Per Page................ 60
Execute Report................ Y  Yes


Finalization Result........... SUCCESS
```

Figure 49 SORYDTS SICAS Abandon DTS Batch Utility Process Report Control Page

Appendix A – SUNY Campus IDs for IR Submissions

		Campus	ID
Doctoral Degree Granting Institutions	Research University Centers		
		Albany	1
		Binghamton	2
		Buffalo Univ	4
		Stony Brook	5
	Other Research and Doctoral Institutions		
		Cornell Stat	165
		Downstate Medical	20
		Upstate Medical	22
		Alfred-Ceramics	28
		Envir Sci & Forestry	23
		Optometry	26
Comprehensive Colleges		Brockport	175
		Buffalo State	7
		Cortland	8
		Empire State	9
		Fredonia	10
		Geneseo	11
		New Paltz	12
		Old Westbury	13
		Oneonta	14
		Oswego	15
		Plattsburgh	16
		Potsdam	17
		Purchase	18
Technology Colleges		Alfred State	30
		Canton	31
		Cobleskill	32
		Delhi	33
		Farmingdale	197
		Maritime	25
		Morrisville	34
		SUNYIT	27

	Campus	ID
Community Colleges	Adirondack	35
	Broome	36
	Cayuga	178
	Clinton	179
	Columbia-Greene	39
	Corning	40
	Dutchess	41
	Erie	166
	Fashion Institute	46
	Finger Lakes	47
	Fulton-Montgomery	48
	Genesee	49
	Herkimer	50
	Hudson Valley	51
	Jamestown	54
	Jefferson	55
	Mohawk Valley	56
	Monroe	57
	North Country	215
	Nassau	58
	Niagara County	59
	Onondaga	61
	Orange County	62
	Rockland	63
	Schenectady	64
	Suffolk	167
	Sullivan	66
	Tompkins Cortland	67
	Ulster County	68
	Westchester	69

Appendix B – Transmission Error Codes

The SICAS Data Transfer Interface Control Form SOAYDTI Processing Errors Tab displays error, debugging, and other types of messages. Some errors are not expected to occur—if encountered, contact SICAS immediately with the details.

This list contains information about the Institutional Research Data Transmission Error Codes. Common errors are listed first.

Code	Description	Severity	Source
SA:Edt	Obsolete Code for System Admin Edits	Errors & Warnings	System Admin
SA:Err	Error	Error; Record Reject	System Admin
SA:Wrn	Warning	Warning; may cause Threshold errors if a large number exist.	System Admin
WS:Ret	Return Value Confirmation Errors Returned Value does not match Sent Value. Return Value received when not allowed.	Warning	SICAS Package
WS: Upd	Return Value could not Update Column Returned value is NULL. Returned value is inappropriate exception.	Error; returned value was not stored back into the staging table.	SICAS Package
SA:ERR	Unknown Error	Unexpected Error; Severity is Unknown.	System Admin
SA:Dfr	Deferred Error	Non-Fatal	System Admin
SA:PSI	PSI Warning – Building/Room Number could not be found in PSI Lookup	Warning	System Admin
SE:Sts	Error Associated with a Batch Status Call (Service Exception)	Non-Fatal; should never occur.	System Admin
SE:INI	Web Service Batch Initialization Error (Service Exception)	Error; Batch Fatal; should never occur.	System Admin
SE:Trn	Web Service Transaction Error (Service Exception)	Error; Record Fatal.	System Admin
SE:FIN	Web Service Finalization Error (Service Exception)	Error; Batch Fatal Batch remains active.	System Admin
WS:FTL	Unexpected Web Service Exception (often follows multiple WS:Exc errors)	Process Failure	SICAS Package

Code	Description	Severity	Source
WS:Exc	Unexpected Web Service Exception – Retry This is usually a timeout on the call; most of the time the next attempt is successful—a WS:FTL is triggered after third failure.	Web Service Failed; will retry the call after a brief delay.	SICAS Package
WS:PCn	Primary Contact Information is Required	Error; Batch Fatal	SICAS Package
WS:Ref	Batch Reference Number not Returned	Error; Batch Fatal; should never occur.	SICAS Package
SA:INI	Batch Initialization Result <> SUCCESS Unexpected call result for Initialization.	Error; Batch Fatal; should never occur.	System Admin Returned Value
WS:XML	XML Creation Failure Usually involves NON-ASCII characters inside ORACLE column values.	Error; Record Fatal.	SICAS Package
WS:ERR	Unknown Error during Batch Initialization or Finalization call. Hard Exception occurred preventing the SE:Ini or SE:Fin handler from reporting the problem.	Error; Fatal Batch on Initialization; Batch remains active on Finalization.	SICAS Package
SA:FIN	Batch Finalization Result <> SUCCESS No exception occurred but System Admin reported that the Batch Finalization was not successful.	Error; Batch remains Active.	System Admin Returned Value
Convrt	Conversion Error	Error, Conversion not completed on SOAYCVT	SICAS Package
Dup	SDS Duplicate Student Detail. Value already exists for this Key.	Warning, Value Ignored	SICAS Package
FATAL	This is a series of Fatal Errors created in response to user selection of parameters for a run. Invalid entries, term conflicts, and inappropriate Run Modes based on System Administration status of batch.	Process Fatal; program cannot run with the parameters/run mode selected by User.	SICAS Package
DEBUG	These are messages (not errors) that are generated by the data collection, processing, and transmission routines when the user has the Debug Messages checkbox checked during the processing run.	Messages	SICAS Package (Optional)

Appendix C – Proxy Client

The Proxy Client is a Java applet developed by System Administration that provides message encryption, digital signing, and HTTPS protocol for IR data submissions using Web Services. Since Course Data Submission (CDS), Student Data Submission (SDS), and Term/Section Data Submission (TSDS) are transmitted using Web Service only, the Proxy Client must be installed and working before these applications can be used.

The Proxy Client software and a text file called README.TXT that contains installation details are located in a ZIP file included in the installation package. This Appendix includes a copy of the README.TXT file found in the installation package.

For additional information on the Proxy Client, email the System Administration Security team at secteam@sysadm.suny.edu.

Alter the code within the Local Campus Modification Function FZ_PROXY_IR to determine the local address of the Proxy Address for IR calls. This Appendix includes a copy of the FZ_PROXY_IR function.

README.TXT

This program is an HTTP proxy that can be used to secure SOAP web service requests made through it by another application. It uses SSL and WS-Security. Details on how these are used are available in the "SUNY Security for SOAP Web Services.pdf" file in the "doc" subdirectory.

Starting the Program

To start the proxy program run the following command from the directory with this file:
java -jar webservice-proxy.jar

This requires Java to be installed and in your path.

Configuring your Application

Your application that makes web service requests will need to be configured to use the proxy. If both the application and the proxy are on the same machine then setting the proxy hostname to "localhost" should work. By default the proxy listens on port 8888. The port can be changed by editing the "server.xml" file in the "conf" subdirectory, shutting down the proxy, and restarting it.

Obtaining a Certificate

A certificate and private key is used to create the WS-Security signature on SOAP requests. A file containing these can be obtained from the web site "https://www.suny.edu/ca/". To use this site your campus security administrator must have assigned you the "SUNY User>Portal Applications>CERTIFICATE_DOWNLOAD" permission. You can look up your campus security administrator using the page at "https://www.suny.edu/security/login/help.do?action=chooseCampus".

An example file for development and testing purposes is provided called "2008-02-29-Institutional_Research-28650.p12". The example secured web services at "https://test.suny.edu/ca/services/" and "http://dev.suny.edu/ca/services/" are configured to accept this certificate as well as any issued from the "https://www.suny.edu/ca/" site.

The proxy can be configured to use a different file by editing the "handler.xml" file in the "conf" subdirectory, shutting down the proxy, and restarting it. The file to use can also be specified on the command line like this: java -jar webservice-proxy.jar -Psign.file=other_file.p12

SSL Configuration

By default the proxy will attempt to contact the web service using HTTPS on port 443. You can change the port the proxy uses to contact the service by editing the "handler.xml" file in the "conf" subdirectory, shutting down the proxy, and restarting it.

Please note that the client application still uses the HTTP protocol when communicating with the proxy. For example, the client application may request to send a web service request to "http://test.suny.edu/ca/services/". To fulfill this request the proxy will access "https://test.suny.edu/ca/services/" using port 443.

The server certificates that the proxy trusts when creating an SSL connection are placed in the "truststore.jks" file. The included version of this file contains the current certificates for "test.suny.edu", "www2.sysadm.suny.edu", and "www.suny.edu".

Contact Information

The System Administration Security team can be contacted at the "secteam@sysadm.suny.edu" email address for any further information.

For campuses that are not running the Proxy Client on their database machine there is an additional change to the "handler.xml" file. Near the top of the file the following block occurs which requires an edit to the bolded line:

```
<handler status="active">
        <name>RestrictClientHandler</name>
         <description>
                Simple access control hander based on source IP addresses.
        </description>
        <removable>true</removable>
        <id>restrictClient</id>
        <files>
                <file />
        </files>
        <params>
                <param name="restrictClient.class"
                        value="sunlabs.brazil.handler.RestrictClientHandler" />

                <param name="restrictClient.prefix" value="" />

                <!-- Restrict connecting IP to the local machine. -->
                <param name="restrictClient.allow" value="127\.0\.0\.1" />
        </params>
</handler>
```

Alter the "value" in this line to contain the IP Address of the database machine that will be calling the Proxy Client. For added security, the Proxy Client will reject all other IP addresses. Note that period characters within the IP address must be escaped (to indicate "." you must enter "\.").

FZ_PROXY_IR

Alter the code within the Local Campus Modification Function FZ_PROXY_IR to determine the local address of the Proxy Address for IR calls.

SICAS provides a template for Local Modifications, but technical staff at each institution is responsible for writing the code within the functions. SICAS ships Local Modifications to CREATE but not replace therefore if modifications are made to the function add 'OR REPLACE' after 'CREATE' at the top of the code. Refer to the comments in the function code for further information.

All modifications made to a Local function are preserved during future releases.

```
CREATE FUNCTION FZ_PROXY_IR
RETURN VARCHAR2 IS
/*

============================================================================

 SICAS CENTER


   Filename:    GZFYPRXY.SQL
   Function:    FZ_PROXY_IR
   Module:      GENERAL
   Created:     11-OCT-2007 NWT
   Release:     7.5S1


   Parameters:  None

   Return Type: VARCHAR2(1)   Proxy Address for IR Calls (i.e.
"http://199.188.10.32:8899");


Description: This is a LOCAL CAMPUS modification function.  Alter code
             within this function to determine the local address of the
             Proxy Address.

             Default behavior raises a descriptive exception.

             SICAS ships this function to CREATE but not replace.  Any
             modifications you make here will be preserved during future
             releases.

Description End


 --------------------------------------------------------------------
   AUDIT TRAIL: 7.5S1            SICAS Center           NWT 04-APR-2008
   FDR # 2002-010
   REL # GENERAL 7.5S1

   1. Made use of new call to support automated proxy client addressing
      changes.  Added it to procedure and documentation.  Requires
      installation of GENERAL 7.5S1 to work.
```

```
    AUDIT TRAIL: 7.4S1.1          SICAS Center              NWT 11-OCT-2007
    FDR #2002-010


  1.  Created Local Mod function to allow each campus to define their
      own proxy address in code to configure Secure IR Web Service
      Calls.
========================================================================
*/


BEGIN
      -------------------------------------
      -- UNCOMMENT AND CORRECT THIS CODE --
      -------------------------------------
      /*
      IF GOKYWSRD.FY_IS_PRODUCTION THEN
        RETURN 'http://999.888.77.66:5555';  -- PROD IR Proxy Address
(your IP and selected Port Number)
      ELSE
        RETURN 'http://999.888.77.66:7777';  -- TEST IR Proxy Address
(your IP and selected Port Number)
      END IF;
      */
      -------------------------------------
      -- COMMENT OUT THIS NEXT CODE LINE --
      -------------------------------------
      --[packit=OFF]
      RETURN 'http://vm-sicasnt1.oneonta.edu:8888';
      --[packit=ON]
      RAISE_APPLICATION_ERROR(-20999,'Required Local Campus Function
"FZ_PROXY_IR" not implemented.');
      RETURN NULL;
END fz_proxy_ir;
```

Appendix D – SUNY Web Service Access Certificates

An X.509 digital certificate, International Telecommunications Union (ITU) X.509 standard, http://www.itu.int/rec/T-REC-X.509-200003-I/en, signed by the SUNY System Administration certificate authority is required to access System Administration's secured web services.

To obtain the certificate, contact your campus Security Administrator and ask to be assigned the "SUNY User>PORTAL>CERTIFICATE_DOWNLOAD" permission.

Security Administrators have been sent an email explaining this permission. A listing of campus Security Administrators is available at:
https://www.suny.edu/security/login/help.do?action=chooseCampus&ignorePreferred=true

Once assigned the security permission, go to the Certificate Download Website at:
https://www.suny.edu/ca/

No permissions are required to download certificates for testing purposes but subscription to SUNYNet is required to access the website. Certificates used for testing are available at:
https://test.suny.edu/ca/

Enter your SUNY System Administration Employee Services portal login information to access these websites. If logged in to the portal when the security permission is assigned, log out and log back in for the permission assignment to take effect.

The Certificate Download for your campus is listed under 'Available credentials'.

Click on a link to download the desired certificate. The link name includes the certificate expiration date and the department that uses the certificate.

Email certificate questions, problems, or comments to the SUNY System Administration Security Team at: secteam@sysadm.suny.edu

Certificate Download Page

Appendix E – Local Campus Modification Function for CDS

SICAS provides a Local Campus modification function for CDS called FZ_CDS_BuildCourses. A local campus can use this mod point to affect the implementation of the specific course record routines at their institution. Alter code within the shipped SICAS Object migrated to the code tree to override the CDS population routines. SICAS ships this function to CREATE but not replace therefore if modifications are made add OR REPLACE at the top of the code. There is an example of a campus modification to the CDS mod point later in this section. Any modifications to a local mod point should be thoroughly tested on a campus test database before implementation in production.

Implementation of this mod point is optional; the SICAS standard routing executes if this mod point is not changed.

```
CREATE FUNCTION fz_CDS_BuildCourses(
  process_in IN SFRYERR.SFRYERR_PROCESS_NAME%TYPE,
  term_in    IN STVTERM.STVTERM_CODE%TYPE,
  pass_in    IN VARCHAR2)
RETURN VARCHAR2 IS
/*
========================================================================
SICAS CENTER

Filename:   SZFYCDS.SQL
Function:   fz_CDS_BuildCourses
Created:    16-AUG-2006 NWT
Module:     STUDENT
Release:    7.3S2


Parameters:    Process_in      Process calling this routine           (required)
               Term_in         Term Code to check                     (required)
               Pass_in         Pass executing                         (required)
               BEFORE_COURSE        -- Before Course Build
               AFTER_COURSE         -- After Course Build
               BEFORE_EQUIV         -- Before Equivalent Course Build
               AFTER_EQUIV          -- After Equivalent Course Build

Return Type: VARCHAR2(1)    Y = Run SICAS routine/run SICAS and then an AFTER PASS
                            N = Do NOT run SICAS routine/no AFTER PASS will happen

               NOTE: The return value only has an effect after the BEFORE passes--it is
                       ignored on AFTER passes.

Description:   This is a LOCAL CAMPUS modification function.  Alter code within this function
               to determine the implementation of the specific course records routine at your
               school.  Implementation of this mod point is completely optional.  The SICAS
               standard routine will execute if this mod point is left as SICAS delivered it.

               This function is called twice--once BEFORE the SICAS standard routine
               executes and again after that process runs.  If your local  campus DOES NOT
               wish to have the standard SICAS code run, then set  this procedure to return 'N'
               on the BEFORE call (there will not be an AFTER PASS if the SICAS routine is
               suppressed.)

                BEFORE PASS—use this to implement your own routine to pick up course
               records to include.  Return 'Y' to continue with the SICAS standard selection
```

(your routine will supplement SICAS selected records) -OR- return 'N' if you do not need to run the SICAS procedure (your routine replaces the SICAS routine).

AFTER PASS—use this to delete SICAS selected records, alter the course data created or to supplement SICAS selections with added records.

Always use the SFKYCVTS.py_Add_CDS_Course() and py_Remove_CDS_Course() calls when creating or removing SOBYCDS records.

SICAS ships this function to CREATE but not replace. Any modifications you make here will be preserved during future releases. Our default function does nothing but return 'Y' causing the SICAS standard routine to execute.

Description End

AUDIT TRAIL: 7.3S2   SICAS Center   NWT 29-JUN-2007
FDR #2002-015

1.   Created Local Mod function to allow each campus to define their own logic for Course record selection.

```
=========================================================================
*/
  vRunSICAS VARCHAR2(1) := 'Y'; -- run SICAS code
BEGIN
 -- ----------------------------------------------------------------
 --  UNCOMMENT AND IMPLEMENT TEST HERE
 -- ----------------------------------------------------------------
  IF pass_in='BEFORE_COURSE' THEN              -- Before Course Build
        --
        -- LOCAL CODE HERE TO SELECT COHORT
        -- to avoid SICAS code use the following line:
        -- vRunSICAS := 'N';
        --
        RETURN vRunSICAS;
  ELSIF pass_in='AFTER_COURSE' THEN            -- After Course Build
        --
        -- LOCAL CODE HERE TO MANIPULATE SICAS SELECTIONS
        --
        RETURN 'Y'; -- ignored but a return is required
  ELSIF pass_in='BEFORE_EQUIV' THEN            -- Before Equivalent Course Build
        --
        -- LOCAL CODE HERE TO SELECT COHORT
        -- to avoid SICAS code use the following line:
        -- vRunSICAS := 'N';
        --
        RETURN vRunSICAS;
  ELSIF pass_in='AFTER_EQUIV' THEN             -- After Equivalent Course Build
        --
        -- LOCAL CODE HERE TO MANIPULATE SICAS SELECTIONS
        --
        RETURN 'Y'; -- ignored but a return is required
  END IF;
END fz_CDS_BuildCourses;
```

EXAMPLE
This copy of the CDS Local Campus modification function FZ_CDS_BuildCourses is modified to prevent sending courses to SUNY System Administration that have a prefix of 'ILR', 'CED', or 'XXX'. The code insertions are highlighted in blue. This code has not been tested. Any modifications to a local mod point should be thoroughly tested on a campus test database before implementation in production.


```
CREATE OR REPLACE FUNCTION fz_CDS_BuildCourses(
   process_in IN SFRYERR.SFRYERR_PROCESS_NAME%TYPE,
   term_in    IN STVTERM.STVTERM_CODE%TYPE,
   pass_in    IN VARCHAR2)
RETURN VARCHAR2 IS
/*
=======================================================================
SICAS CENTER

Filename:   SZFYCDS.SQL
Function:   fz_CDS_BuildCourses
Created:    16-AUG-2006 NWT
Module:     STUDENT
Release:    7.3S2
```

Parameters: Process_in   Process calling this routine   (required)
                Term_in   Term Code to check   (required)
                Pass_in   Pass executing   (required)
                                 BEFORE_COURSE   -- Before Course Build
                                 AFTER_COURSE   -- After Course Build
                                 BEFORE_EQUIV   -- Before Equivalent Course Build
                                 AFTER_EQUIV   -- After Equivalent Course Build

Return Type: VARCHAR2(1)   Y = Run SICAS routine/run SICAS and then an AFTER PASS
                             N = Do NOT run SICAS routine/no AFTER PASS will happen

                  NOTE: The return value only has an effect after the BEFORE passes--it is
                        ignored on AFTER passes.

Description: This is a LOCAL CAMPUS modification function. Alter code within this function
                to determine the implementation of the specific course records routine at your
                school. Implementation of this mod point is completely optional. The SICAS
                standard routine will execute if this mod point is left as SICAS delivered it.

                This function is called twice--once BEFORE the SICAS standard routine
                executes and again after that process runs. If your local campus DOES NOT
                wish to have the standard SICAS code run, then set this procedure to return 'N'
                on the BEFORE call (there will not be an AFTER PASS if the SICAS routine is
                suppressed.

                 BEFORE PASS—use this to implement your own routine to pick up course
                records to include. Return 'Y' to continue with the SICAS standard selection
                (your routine will supplement SICAS selected records) -OR- return 'N' if you do
                not need to run the SICAS procedure (your routine replaces the SICAS routine).

                AFTER PASS—use this to delete SICAS selected records, alter the course data
                created or to supplement SICAS selections with added records.

Always use the SFKYCVTS.py_Add_CDS_Course() and
py_Remove_CDS_Course() calls when creating or removing SOBYCDS records.

SICAS ships this function to CREATE but not replace. Any modifications you
make here will be preserved during future releases. Our default function does
nothing but return 'Y' causing the SICAS standard routine to execute.

Description End

AUDIT TRAIL: 7.3S2   SICAS Center   NWT 29-JUN-2007
FDR #2002-015

1.   Created Local Mod function to allow each campus to define their own logic for Course
        record selection.

```
============================================================================
*/
  vRunSICAS VARCHAR2(1) := 'Y'; -- run SICAS code
BEGIN
 -- ----------------------------------------------------------------
 --  UNCOMMENT AND IMPLEMENT TEST HERE
 -- ----------------------------------------------------------------
  IF pass_in='BEFORE_COURSE' THEN                -- Before Course Build
        --
        -- LOCAL CODE HERE TO SELECT COHORT
        -- to avoid SICAS code use the following line:
        -- vRunSICAS := 'N';
        --
        RETURN vRunSICAS;
  ELSIF pass_in='AFTER_COURSE' THEN              -- After Course Build
        --
        -- LOCAL CODE HERE TO MANIPULATE SICAS SELECTIONS
        --

    DELETE FROM SOBYCDS S
    WHERE S.SOBYCDS_TERM_CODE=term_in
    AND S.SOBYCDS_SUBJ_CODE IN ('IRL','CED','XXX');
    COMMIT;

        RETURN 'Y'; -- ignored but a return is required
  ELSIF pass_in='BEFORE_EQUIV' THEN              -- Before Equivalent Course Build
        --
        -- LOCAL CODE HERE TO SELECT COHORT
        -- to avoid SICAS code use the following line:
        -- vRunSICAS := 'N';
        --
        RETURN vRunSICAS;
  ELSIF pass_in='AFTER_EQUIV' THEN               -- After Equivalent Course Build
        --
        -- LOCAL CODE HERE TO MANIPULATE SICAS SELECTIONS
        --
        RETURN 'Y'; -- ignored but a return is required
  END IF;
END fz_CDS_BuildCourses;
```

Appendix F – Local Campus Modification Function for TSDS

SICAS provides a Local Campus modification function for TSDS called FZ_TSDS_BuildSection. A local campus can use this mod point to affect the implementation of specific section record routines at their institution. Alter code within this function to override the TSDS population routines. SICAS ships this function to CREATE but not replace therefore if modifications are made add OR REPLACE at the top of the code. Any modifications to a local mod point should be thoroughly tested on a campus test database before implementation in production.

Implementation of this mod point is optional; the SICAS standard routing executes if this mod point is not changed.

```
CREATE FUNCTION fz_TSDS_BuildSection(
  process_in IN SFRYERR.SFRYERR_PROCESS_NAME%TYPE,
  term_in    IN STVTERM.STVTERM_CODE%TYPE,
  pass_in    IN VARCHAR2,
  crn_in     IN VARCHAR2 DEFAULT NULL,
  pidm_in    IN VARCHAR2 DEFAULT NULL)
RETURN VARCHAR2 IS
/ *


=========================================================================
SICAS CENTER


Filename:       SZFYTDS.SQL
Function:       fz_TSDS_BuildSections
Created:        06-DEC-2007 NWT
Module:         STUDENT
Release:        BETA_TSDS


Parameters:
        Process_in      Process calling this routine             (required)

        Term_in         Term Code to check                       (required)

        Pass_in         Pass executing                           (required)
                        BEFORE_SECTIONS      -- Before Sections are scanned
                        AFTER_SECTIONS       -- After Sections are scanned
                        AFTER_PROCESSING     -- After ALL processing is done
                        BEFORE_MEETINGS      -- Before Meetings are created for Section
                        AFTER_MEETINGS       -- After Meetings are created for Section

                        All of these are for a single CRN using CRN_in below:

                        AFTER_SECTION        -- After Single Section processed
                        BEFORE_INSTRUCTORS -- Before Instructors are added to Section
                        AFTER_INSTRUCTORS    -- After Instructors are added to Section
                        BEFORE_STUDENTS      -- Before Students are added to Section
                        AFTER_STUDENTS       -- After Students are added to Section
                        AFTER_INSTRUCTOR     -- After each Instructor's base record
                                                    (CRN IS NULL)
                        AFTER_STUDENT    -- After each Student's base record
                                                    (CRN IS NULL)
```

These are for a single CRN and PIDM Combination using both CRN_in and Pidm_in below:

AFTER_INSTRUCTOR   -- After Instructor is added to Section
(CRN NOT NULL)
AFTER_STUDENT   -- After each Student is added to Section
(CRN NOT NULL)

CRN_in   CRN on all passes other than BEFORE_SECTIONS, AFTER_SECTIONS, BEFORE_MEETINGS,AFTER_MEETINGS and AFTER_PROCESSING. When not NULL you only consider items for the single CRN.

Pidm_in   PIDM of the Student or Instructor added on the AFTER_STUDENT and AFTER_INSTRUCTOR calls. When not null, restrict work to single CRN and Pidm combination.

Return Type: VARCHAR2(1)   Y = Run SICAS routine/run SICAS and then an AFTER PASS
N = Do NOT run SICAS routine / no AFTER PASS will happen

NOTES: The return value only has an effect after the BEFORE passes— it is ignored on AFTER passes. Each BEFORE pass return controls AFTER passes of the same type. AFTER single CRN/PIDM calls always occur--they are not controlled by any BEFORE pass.

Description: This is a LOCAL CAMPUS modification function. Alter code within this function to determine the implementation of the specific section records routine at your school. Implementation of this mod point is completely optional. The SICAS standard routine will execute if this mod point is left as SICAS delivered it.

This function is called twice for each type of pass that is needed for the TSDS build process. In most cases the calls are paired with a BEFORE call and an AFTER call and the AFTER call depends on whether or not the SICAS routine is run. There are a few specific CRN/PIDM calls which appear only as after calls and those are always executed. The call stack is as follows:

BEFORE_SECTIONS Call   (called before any sections are added)
IF 'Y' returned THEN
        SICAS SECTIONS ROUTINE
        AFTER_SECTIONS Call (called after optional SICAS routine)
END IF;
SOAYDTI Interface Runs for Sections

BEFORE_MEETINGS Call
IF 'Y' returned THEN
        SICAS MEETINGS ROUTINE
        AFTER_MEETINGS Call (called after optional SICAS routine)
END IF;
SOAYDTI Interface Runs for Meetings

BEFORE_INSTUCTORS Call
IF 'Y' returned THEN
        SICAS INSTRUCTORS ROUTINE
        AFTER_INSTRUCTORS Call (called after optional SICAS routine)
END IF;

SOAYDTI Interface Runs for Instructors
FOR Instructors IN SOBYINS LOOP
        AFTER_INSTRUCTOR Call (chance to alter instructor -- CRN is NULL)
END LOOP;
FOR Instructors IN SORYINS LOOP
        AFTER_INSTRUCTOR Call (chance to alter final Section instructor
        record)
END LOOP;

BEFORE_STUDENTS Call
IF 'Y' returned THEN
        SICAS STUDENTS ROUTINE
        AFTER_STUDENTS Call (called after optional SICAS routine)
END IF;
SOAYDTI Interface Runs for Students
FOR Students IN SOBYSTU LOOP
        AFTER_STUDENT Call (chance to alter student record -- CRN is NULL)
END LOOP;
FOR Students IN SORYSTU LOOP
        AFTER_STUDENT Call (chance to alter final Section student record)
END LOOP;

FOR Sections IN SOBYSEC LOOP
        AFTER_SECTION Call (chance to alter final Section record)
END LOOP;
AFTER_PROCESSSING Call (for whatever purpose you might have)

BEFORE PASSES—use this to implement your own routine to pick up records to include.
Return 'Y' to continue with the SICAS standard selection (your routine will supplement
SICAS selected records) -OR- return 'N' if you do not need to run the SICAS procedure
(your routine replaces the SICAS routine).

AFTER PASSES—use these to delete SICAS selected records, alter the course data
created or to supplement SICAS selections with added records. Some AFTER passes
are specific to a CRN, a PIDM or to both a CRN & PIDM.

Always use the following SFKYCVIR package procedures to add or remove sections or
related records:

py_Add_TSDS_Section()   adds or updates record in SOBYSEC
py_Rem_TSDS_Section()   removes record in SOBYSEC and its children
py_Add_TSDS_Meeting()   adds a SORYMET record
py_Rem_TSDS_Meeting()   removes a record from SORYMET
py_Add_TSDS_Student()   adds a student pidm to SORYSTU and adds
                               a SOBYSTU base student record if none exists
py_Rem_TSDS_Student()   removes a SORYSTU record and related SOBYSTU
                               base record if this was the last SORYSTU record
py_Add_TSDS_Instructor()   adds an instructor pidm to SORYINS and adds
                               a SOBYINS base student record if none exists
py_Rem_TSDS_Instructor()   removes a SORYINS record and related SORYINS
                               base record if this was the last SORYINS record

SICAS ships this function to CREATE but not replace. Any modifications you make here
will be preserved during future releases. Our default function does nothing but return 'Y'
causing the SICAS standard routine to execute.

Description End

```
 --------------------------------------------------------------------------------------
  AUDIT TRAIL: BETA_TSDS          SICAS Center              NWT 20-NOV-2007
  FDR #2002-015

  1. Created Local Mod function to allow each campus to define their own logic for Course
     record selection.


  =========================================================================
*/
 vRunSICAS VARCHAR2(1) := 'Y'; -- run SICAS code
BEGIN
  -- ----------------------------------------------------------------------
  --  UNCOMMENT AND IMPLEMENT TEST HERE
  -- ----------------------------------------------------------------------
  IF pass_in='BEFORE_SECTIONS' THEN              -- Before Section Build
     --
     -- LOCAL CODE HERE TO SELECT COHORT
     -- to avoid SICAS code use the following line:
     -- vRunSICAS := 'N';
     --
     RETURN vRunSICAS;
  ELSIF pass_in='AFTER_SECTIONS' THEN            -- After Section Build
     --
     -- LOCAL CODE HERE TO MANIPULATE SICAS SELECTIONS
     --
     RETURN 'Y'; -- ignored but a return is required
  ELSIF pass_in='AFTER_SECTION' THEN             -- After Section (CRN is not NULL)
     --
     -- LOCAL CODE HERE TO MANIPULATE SINGLE CRN
     --
     RETURN 'Y'; -- ignored but a return is required
  ELSIF pass_in='BEFORE_MEETINGS' THEN           -- Before Section Meetings Build
     --
     -- LOCAL CODE HERE TO SELECT COHORT
     -- to avoid SICAS code use the following line:
     -- vRunSICAS := 'N';
     --
     RETURN vRunSICAS;
  ELSIF pass_in='AFTER_MEETINGS' THEN             -- After Section Meetings Build
     --
     -- LOCAL CODE HERE TO MANIPULATE SICAS SELECTIONS
     --
     RETURN 'Y'; -- ignored but a return is required
  ELSIF pass_in='BEFORE_INSTRUCTORS' THEN        -- Before Section Instructors Build
     --
     -- LOCAL CODE HERE TO SELECT COHORT
     -- to avoid SICAS code use the following line:
     -- vRunSICAS := 'N';
     --
     RETURN vRunSICAS;
  ELSIF pass_in='AFTER_INSTRUCTORS' THEN         -- After Section Instructors Build
     --
     -- LOCAL CODE HERE TO MANIPULATE SICAS SELECTIONS
```

```
   --
   RETURN 'Y'; -- ignored but a return is required
 ELSIF (pass_in='AFTER_INSTRUCTOR') AND (CRN_in IS NULL) THEN
                                                 -- After Instructor Base (CRN IS NULL)
   --
   -- LOCAL CODE HERE TO MANIPULATE SINGLE INSTRUCTOR PIDM
   --
   RETURN 'Y'; -- ignored but a return is required
 ELSIF (pass_in='AFTER_INSTRUCTOR') AND (CRN_in IS NOT NULL) THEN
                                                 -- Add Instructor to Section (CRN NOT NULL)
   --
   -- LOCAL CODE HERE TO MANIPULATE SINGLE INSTRUCTOR PIDM & CRN
   --
   RETURN 'Y'; -- ignored but a return is required
 ELSIF pass_in='BEFORE_STUDENTS' THEN          -- Before Section Students Build
   --
   -- LOCAL CODE HERE TO SELECT COHORT
   -- to avoid SICAS code use the following line:
   -- vRunSICAS := 'N';
   --
   RETURN vRunSICAS;
 ELSIF pass_in='AFTER_STUDENTS' THEN           -- After Section Instructors Build
   --
   -- LOCAL CODE HERE TO MANIPULATE SICAS SELECTIONS
   --
   RETURN 'Y'; -- ignored but a return is required
 ELSIF (pass_in='AFTER_STUDENT') AND (CRN_in IS NULL) THEN
                                                 -- After Student Base (CRN IS NULL)
   --
   -- LOCAL CODE HERE TO MANIPULATE SINGLE STUDENT PIDM
   --
   RETURN 'Y'; -- ignored but a return is required
 ELSIF (pass_in='AFTER_STUDENT') AND (CRN_in IS NOT NULL) THEN
                                                 -- Add Student to Section (CRN NOT NULL)
   --
   -- LOCAL CODE HERE TO MANIPULATE SINGLE INSTRUCTOR PIDM & CRN
   --
   RETURN 'Y'; -- ignored but a return is required
 ELSIF pass_in='AFTER_PROCESSING' THEN         -- After all processing complete
   --
   -- LOCAL CODE HERE TO ALTER ANY TSDS RELATED DATA
   --
   RETURN 'Y'; -- ignored but a return is required
 END IF;
END fz_TSDS_BuildSection;
```

Appendix G – Local Campus Modification Functions for SDS

SICAS provides three Local Campus modification functions for SDS. The FZ_SDS_Admission_Status is required and must be altered to return the student's Admission Status. The FZ_SDS_BuildStudents and FZ_SDS_InculdeStudent functions are optional.

FZ_SDS_Admission_Status (required)

The SDS data element 'Adm Status' indicates an individual's Admission Status. Each institution determines Admission Status differently, therefore SICAS provides a Local Campus modification function that must be altered to return the correct System Administration Admission Status code.

Implementation of this mod point is required. If the FZ_SDS_Admission_Status. function is not altered locally by campus technical staff, the routine returns a 'Y' and the SICAS standard routine executes. The SICAS standard routine returns an invalid Admission Code of '999'.

Any modifications to a local mod point should be thoroughly tested on a campus test database before implementation in production.

```
CREATE OR REPLACE FUNCTION fz_SDS_Admission_Status(
   process_in IN SFRYERR.SFRYERR_PROCESS_NAME%TYPE,
   pidm_in    IN SPRIDEN.SPRIDEN_PIDM%TYPE,
   term_in    IN STVTERM.STVTERM_CODE%TYPE)
RETURN NUMBER IS
```

Description
```
This is a LOCAL CAMPUS modification function. Alter code within this
function to determine the implementation of the specific logic used
locally to determine a Student's Admission Status.  '999' is returned
by this routine as SICAS delivered it -- which is an invalid code.

SICAS ships this function to CREATE but not replace. Any modifications
you make here will be preserved during future releases. Our default
function does nothing but return 'Y' causing the SICAS standard routine
to execute.

 %param Process_in     Process calling this routine
 %param pidm_in        Student PIDM being considered for inclusion
 %param term_in        Term Code to check

 %return NUMBER        Supported System Administration Admission Status
Codes:
                        1 = Regular Admit
                        2 = Special Admit Receiving Aid
                        3 = Special Admit Disadvantaged
                        4 = Other Special Admit
                        5 = Non-Degree Seeking
                        6 = Continuing Education
                        7 = Cross Registered - Accessory (Cornell,
                            Alfred Ceramics, or ESF campuses only)
                        8 = Cross Registered - Another SUNY
                        9 = Cross Registered - Non-SUNY
                       10 = Exchange Student from Abroad
```
Description End

```
=======================================================================
    SICAS CENTER

    Filename:      SZFYSDS.SQL
    Function:      fz_SDS_Admission_Status
    Created:       23-JUN-2008 NWT
    Module:        STUDENT
    Release:       7.4S2


    ---------------------------------------------------------------------
    AUDIT TRAIL: 7.4S2        SICAS Center                NWT 18-JUN-2008
    FDR #2002-015

    1.  Created Local Mod function to allow each campus to define their
own logic for the admission status of the student.
================================================================
*/

  vRESULT NUMBER := '999'; -- an invalid code
BEGIN
  -- ----------------------------------
  --  ADD YOUR CUSTOM CAMPUS LOGIC HERE
  -- ----------------------------------
  -- 1 = Regular Admit
  -- 2 = Special Admit Receiving Aid
  -- 3 = Special Admit Disadvantaged
  -- 4 = Other Special Admit
  -- 5 = Non-Degree Seeking
  -- 6 = Continuing Education
  -- 7 = Cross Registered - Accessory (Cornell, Alfred Ceramics, or ESF
campuses only)
  -- 8 = Cross Registered - Another SUNY
  -- 9 = Cross Registered - Non-SUNY
  --10 = Exchange Student from Abroad
  -- ----------------------------------
  RETURN vRESULT;
END fz_SDS_Admission_Status;
```

FZ_SDS_BuildStudents (optional)

The Local Modification function FZ_SDS_BuildStudents allows an institution to modify the student population globally or on a student-by-student basis at different times during the SICAS Student Data Submission Process SGRYSDS. This function is called before the SGRYSDS population routine is executed and again afterwards.

Any modifications to a local mod point should be thoroughly tested on a campus test database before implementation in production.

```
CREATE OR REPLACE FUNCTION fz_SDS_BuildStudents(
   process_in IN SFRYERR.SFRYERR_PROCESS_NAME%TYPE,
   term_in    IN STVTERM.STVTERM_CODE%TYPE,
   census_in  IN DATE,
   RunFor_in  IN VARCHAR2,
   pass_in    IN VARCHAR2,
   pidm_in    IN VARCHAR2 DEFAULT NULL,
   type_in    IN VARCHAR2 DEFAULT NULL,
   element_in IN VARCHAR2 DEFAULT NULL)
RETURN VARCHAR2 IS
/**
Description
This is a LOCAL CAMPUS modification function. Alter code within this
function to determine the implementation of the specific section
records routine at your school.  Implementation of this mod point is
completely optional.  The SICAS standard routine will execute if this
mod point is left as SICAS delivered it.

This function is called twice for each type of pass that is needed for
the SDS build process.  In most cases the calls are paired with a
BEFORE call and an AFTER call and the AFTER call depends on whether or
not the SICAS routine is run.  There are a few specific CRN/PIDM calls
which appear only as after calls and those are always executed. The
call stack is as follows:

        BEFORE_STUDENTS Call   (called before any sections are added)
        IF 'Y' returned THEN
           SICAS SECTIONS ROUTINE
           AFTER_STUDENTS Call (called after optional SICAS routine)
        END IF;
        SOAYDTI Interface Runs for Sections

BEFORE PASSES -- use this to implement your own routine to pick up
records to include.  Return 'Y' to continue with the SICAS standard
selection (your routine will supplement SICAS selected records) -OR-
return 'N' if you do not need to run the SICAS procedure (your routine
replaces the SICAS routine).

AFTER PASSES -- use these to delete SICAS selected records, alter the
student data created or to supplement SICAS selections with added
records.  The STUDENT_LOOP call provides a single student PIDM for the
updates.

Always use the following SFKYCVIR package procedures to add or remove
sections or related records:

 py_Add_SDS_Student()     adds a student pidm to SOBYSDS if none exists
```

```
 py_Rem_SDS_Student()     removes a SOBYSDS record and related SORYSDS
                          records that exist


 py_Add_SDS_StudentDetail()    adds a student pidm to SORYSDS and adds
                               a SOBYSDS base student record if none exists


py_Rem_SDS_StudentDetail()    removes a SORYSDS record


SICAS ships this function to CREATE but not replace.  Any modifications
you make here will be preserved during future releases. Our default
function does nothing but return 'Y' causing the SICAS standard routine
to execute.

 %param    Process_in    Process calling this routine
 %param    Term_in       Term Code to check
 %param    Census_in     Census Date for the Run
 %param    RunFor_in     CENSUS or FINAL
 %param    Pass_in       Pass executing:
           BEFORE_STUDENTS    -- Before Students are scanned
           AFTER_STUDENTS     -- After Students are scanned
           BEFORE_REPEAT      -- Before Student Repeating Items
                                 (Element_in is set)
           AFTER_REPEAT       -- After Student Repeating Items
                                 (Element_in is set)
           STUDENT_LOOP       -- Loops through students for student by
                                  student changes
           AFTER_PROCESSING   -- called after all processing is complete
                      but before transmission of data to System Admin.

 %param    Pidm_in       PIDM of the Student (NULL on all calls except
                         STUDENT_LOOP
 %param    type_in       Element Type from SOAYDTI for Repeating
                         Elements (NULL on STUDENT calls)
 %param    element_in    Element Name from SOAYDTI for Repeating
                         Elements (NULL on STUDENT calls)


 %return   VARCHAR2(1)   Y = Run SICAS routine / run SICAS and then an
                             AFTER PASS
                         N = Do NOT run SICAS routine / no AFTER PASS
                             will happen

 NOTES: The return value only has an effect after the BEFORE passes--it
is ignored on AFTER passes. Each BEFORE pass return controls AFTER
passes of the same type.  AFTER single CRN / PIDM calls always
occur--they are not controlled by any BEFORE pass.
*/
```

Description End

```
/*
=======================================================================
   SICAS CENTER
   FILENAME:    SZFYSDS.SQL
   FUNCTION:    fz_SDS_BuildStudents
   CREATED:     06-MAY-2008 NWT
   MODULE:      STUDENT
   RELEASE:     7.4S2
```

```
    USAGE:        Local Mod Point allowing Campus Control over Student
Submission Population
--SICAS.HEND
--SICAS.ABEGIN
   ----------------------------------------------------------------------
   AUDIT TRAIL: 7.4S2      SICAS Center                  NWT 06-MAY-2008
   FDR#2004-026
   1.  Created Local Mod function to allow each campus to define their
own logic for Student record selection.
=========================================================================
--SICAS.AEND
*/
  vRunSICAS VARCHAR2(1) := 'Y'; -- run SICAS code
BEGIN
  -- ----------------------------------
  --  UNCOMMENT AND IMPLEMENT TEST HERE
  -- ----------------------------------
  IF pass_in='BEFORE_STUDENTS' THEN                              --
Before Students Build
       --
       -- LOCAL CODE HERE TO SELECT COHORT
       -- to avoid SICAS code use the following line:
       -- vRunSICAS := 'N';
       --
       RETURN vRunSICAS;
  ELSIF pass_in='AFTER_STUDENTS' THEN                            --
After Section Build
       --
       -- LOCAL CODE TO DELETE RECORDS WE ADDED
       -- OR ALTER THE DATA
       --
       RETURN 'Y'; -- ignored but a return is required
  ELSIF pass_in='BEFORE_REPEAT' THEN                             --
Before each repeating type
       --
       -- LOCAL CODE HERE TO TAKE OVER DATA COLLECTION
       --
       RETURN 'Y'; -- ignored but a return is required
  ELSIF pass_in='AFTER_REPEAT' THEN                              --
After each repeating type
       --
       -- LOCAL CODE HERE TO DELETE UNWANTED DATA
       -- OR ALTER THE DATA
       --
       RETURN 'Y'; -- ignored but a return is required
  ELSIF pass_in='STUDENT_LOOP' THEN                              -- One
Call per student
       --
       -- LOCAL CODE HERE TO DELETE UNWANTED DATA
       -- OR ALTER THE DATA
       --
       RETURN 'Y'; -- ignored but a return is required
  ELSIF pass_in='AFTER_PROCESSING' THEN                          --
After all processing complete
       --
       -- LOCAL CODE HERE TO ALTER ANY SDS RELATED DATA
       --
```

CONFIDENTIAL

```
      RETURN 'Y'; -- ignored but a return is required
   END IF;
END fz_SDS_BuildStudents;
```

FZ_SDS_IncludeStudent (optional)

The Local Modification function FZ_SDS_IncludeStudent allows an institution to decide if a student should be included or excluded from the population selection based on criteria created by the campus rather than the criteria used by the standard SDS processing.

Any modifications to a local mod point should be thoroughly tested on a campus test database before implementation in production.

```
CREATE OR REPLACE FUNCTION fz_SDS_IncludeStudent(
   process_in IN SFRYERR.SFRYERR_PROCESS_NAME%TYPE,
   pidm_in    IN SPRIDEN.SPRIDEN_PIDM%TYPE,
   term_in    IN STVTERM.STVTERM_CODE%TYPE,
   crn_in     IN SSBSECT.SSBSECT_CRN%TYPE,
   sicas_in   IN VARCHAR2)
RETURN VARCHAR2 IS
/**
```

Description
```
 This is a LOCAL CAMPUS modification function. Alter code within this
function to determine the implementation of the specific logic used
locally to determine that a Student is required in the SDS extract
based on the specific SECTION and STUDENT being reviewed. The sicas_in
is returned by this routine as SICAS delivered it.

 SICAS ships this function to CREATE but not replace. Any modifications
you make here will be preserved during future releases. Our default
function does nothing but return 'Y' causing the SICAS standard routine
to execute.

 %param Process_in     Process calling this routine
 %param pidm_in        Student PIDM being considered for inclusion
 %param term_in        Term Code to check
 %param crn_in         CRN being considered as cause for including
 %param sicas_in       SICAS determined Inclusion for the Student
                       Y = SICAS routine decided the student should be
                           included
                       N = SICAS routine decided the student is not
                           needed
 %return VARCHAR2      Y = Campus routine decided to Include this PIDM
                       N = Campus routine decided not to include PIDM
```
Description End
```
*/
/*
```

```
--SICAS.HBEGIN

===========================================================================
    SICAS CENTER


    Filename:    SZFYSDS.SQL
    Function:    fz_SDS_IncludeStudent
    Created:     16-JUN-2008 NWT
    Module:      STUDENT
    Release:     7.4S2


    ----------------------------------------------------------------------
--SICAS.HEND
--SICAS.ABEGIN
    AUDIT TRAIL: 7.4S2     SICAS Center                    NWT 18-JUN-2008
    FDR #2002-015

    1.  Created Local Mod function to allow each campus to define their
own logic for the including of Students in SDS based on Registration
for a term.


===========================================================================
--SICAS.AEND
*/
  vRESULT VARCHAR2(1) := sicas_in; -- run SICAS code
BEGIN
  -- ----------------------------------
  --  ADD YOUR CUSTOM CAMPUS LOGIC HERE
  -- ----------------------------------
  --  Alter the contents of vRESULT to
  --  Y or N indicating your routine's
  --  decision about including the PIDM.
  -- ----------------------------------
  RETURN vRESULT;
END fz_SDS_IncludeStudent;
```

Appendix H – Local Campus Modification Functions for Multi-Term Processing

   SICAS has developed the Multi-term package to assist campuses who use a separate term for Non-Credit courses. These non-credit terms do contain data that is required by System Administration and must be sent with the SDS, CDS and TSDS submissions.
  This package is NOT shipped in a regular release by SICAS and will NOT be maintained by SICAS over time. Campuses are encouraged to modify this package in anyway the need to for their specific data situations. Please do not submit RFs related to this package.

   This package WILL NOT be included in any SICAS Releases or Patches. Campuses that wish to implement this solution should request the package by sending an email to sicascen@oneonta.edu.

   This Code is supplied to SICAS Clients to assist in developing Local Mod Points supporting Multi-Term SIRIS Submissions. This package was not built to act as a final solution for any campus but rather was built as a good start for any campuses specific implementation. Code samples have been embedded in the documentation below which show these routines being called from the appropriate locations within the FZ mod points for CDS,TSDS & SDS.

   CURSOR objects included in this package attempts to duplicate selection criteria of the baseline SIRIS processing while attempting to provide more data sources to provide the campus programmer with more information that might be useful when deciding whether or not to include the selected records.

```
   -------------------------------------------------------------------
--------------- */
--SICAS.HEND
--SICAS.ABEGIN
/*
  AUDIT_TRAIL: 7.5S1.2              SICAS Center      NWT 04-NOV-2009
  RF#5848
  1. Released with Package Body

  AUDIT_TRAIL: 7.5S1.1              SICAS Center      NWT 14-OCT-2009
  RF#5787
  1. Released with Package Body

  AUDIT_TRAIL: 7.5S1.0              SICAS Center      NWT 26-JAN-2009
  1. Created package of alternate term loading routines to act as the
     basis for Local Mods to collect these records for clients that
     use special Non-Credit and High School Credit terms.

--SICAS.AEND
*/
-------------------------------------------------------------------
----------
-- Sample CDS Alternate Term Call
-------------------------------------------------------------------
----------
/** Procedure to Add Alternate CDS Term records.
    %param  process_in  Name of the Process which called the FZ calling
this routine.
    %param  term_in     Term of the SUBMISSION (not the Alternate Term
being added).
```

```
    %param  alt_term_in Term of the records being added to the
Submission for Term_in.

Suggested Use in FUNCTION fz_CDS_BuildCourses:

  ELSIF pass_in='AFTER_COURSE' THEN                          -- After
Course Build
    -- LOCAL CODE BEGINS --
    DECLARE
       alt_term STVTERM.STVTERM_CODE%TYPE;  -- variable to calculate
alt term(s)
     BEGIN
       -- process each valid Submission Term adding additional term(s)
needed

       IF term_in LIKE '%10' THEN        -- Fall Term / One Alternate
Term

          alt_term := SUBSTR(term_in,1,4)||'05';  -- example: 200010 &
200005
          SFKYCVIR_LOCAL.pz_cds_alt_term(process_in, term_in,
alt_term);

       ELSIF term_in LIKE '%20' THEN     -- Spring / Two Alternate
Terms

          alt_term := SUBSTR(term_in,1,4)||'15';  -- example: 200020 &
200015
          SFKYCVIR_LOCAL.pz_cds_alt_term(process_in, term_in,
alt_term);

          alt_term := SUBSTR(term_in,1,4)||'25';  -- example: 200020 &
200025
          SFKYCVIR_LOCAL.pz_cds_alt_term(process_in, term_in,
alt_term);

       ELSIF ...term tests as needed...
             ...more code as needed...
       ELSE
          -- invalid term should STOP process
          RAISE_APPLICATION_ERROR(-20000,'LOCAL=Entered invalid
Submission Term')
       END IF;
     END;
    -- LOCAL CODE ENDS --
    RETURN 'Y'; -- ignored but a return is required
*/
--------------------------------------------------------------------------
----------
PROCEDURE pz_cds_alt_term(
   process_in  IN SFRYERR.SFRYERR_PROCESS_NAME%TYPE,
   term_in     IN STVTERM.STVTERM_CODE%TYPE,
   alt_term_in IN SOBYCDS.SOBYCDS_SOURCE_TERM%TYPE
);
--------------------------------------------------------------------------
----------
-- Sample TSDS Alternate Term Call
```

```
------------------------------------------------------------------------
----------
/** Procedure to Add Alternate TSDS Term records.
    %param  process_in  Name of the Process which called the FZ calling
this routine.
    %param  term_in     Term of the SUBMISSION (not the Alternate Term
being added).
    %param  alt term in Term of the records being added to the
Submission for Term_in.

Suggested Use in FUNCTION fz_TSDS_BuildSection:

  ELSIF pass in='AFTER_SECTIONS' THEN                           --
After Section Build
    -- LOCAL CODE BEGINS --
    DECLARE
        alt_term STVTERM.STVTERM_CODE%TYPE;  -- variable to calculate
alt term(s)
     BEGIN
        -- process each valid Submission Term adding additional term(s)
needed

        IF term_in LIKE '%10' THEN         -- Fall Term / One Alternate
Term

          alt_term := SUBSTR(term_in,1,4)||'05';  -- example: 200010 &
200005
          SFKYCVIR_LOCAL.pz_tsds_alt_term(process_in, term_in,
alt_term);

        ELSIF term_in LIKE '%20' THEN      -- Spring / Two Alternate
Terms

          alt_term := SUBSTR(term_in,1,4)||'15';  -- example: 200020 &
200015
          SFKYCVIR_LOCAL.pz_tsds_alt_term(process_in, term_in,
alt_term);

          alt_term := SUBSTR(term_in,1,4)||'25';  -- example: 200020 &
200025
          SFKYCVIR_LOCAL.pz_tsds_alt_term(process_in, term_in,
alt_term);

        ELSIF ...term tests as needed...
             ...more code as needed...
        ELSE
           -- invalid term should STOP process
           RAISE_APPLICATION_ERROR(-20000,'LOCAL=Entered invalid
Submission Term')
        END IF;
     END;
    -- LOCAL CODE ENDS --
     RETURN 'Y'; -- ignored but a return is required
*/
------------------------------------------------------------------------
----------
PROCEDURE pz_tsds_alt_term(
```

```
    process_in  IN SFRYERR.SFRYERR_PROCESS_NAME%TYPE,
    term_in     IN STVTERM.STVTERM_CODE%TYPE,
    alt_term_in IN SOBYCDS.SOBYCDS_SOURCE_TERM%TYPE
);
--------------------------------------------------------------------------
----------
-- Sample SDS Alternate Term Call
--------------------------------------------------------------------------
----------
/** Procedure to Add Alternate SDS Term records.
    %param  process_in  Name of the Process which called the FZ calling
this routine.
    %param  term_in     Term of the SUBMISSION (not the Alternate Term
being added).
    %param  census_in   Census Date (or NULL if Final Submission is
processing).
    %param  alt_term_in Term of the records being added to the
Submission for Term_in.

Suggested Use in FUNCTION fz_SDS_BuildStudents:

  ELSIF pass_in='AFTER_STUDENTS' THEN                              --
After Section Build
    -- LOCAL CODE BEGINS --
    DECLARE
        alt_term STVTERM.STVTERM_CODE%TYPE;  -- variable to calculate
alt term(s)
     BEGIN
        -- process each valid Submission Term adding additional term(s)
needed

        IF term_in LIKE '%10' THEN         -- Fall Term / One Alternate
Term

           alt_term := SUBSTR(term_in,1,4)||'05';  -- example: 200010 &
200005
           SFKYCVIR LOCAL.pz_sds_alt_term(process_in, term_in,
census_in, alt_term);

        ELSIF term_in LIKE '%20' THEN      -- Spring / Two Alternate
Terms

           alt_term := SUBSTR(term_in,1,4)||'15';  -- example: 200020 &
200015
           SFKYCVIR LOCAL.pz_sds_alt_term(process_in, term_in,
census_in, alt_term);

           alt_term := SUBSTR(term_in,1,4)||'25';  -- example: 200020 &
200025
           SFKYCVIR LOCAL.pz_sds_alt_term(process_in, term_in,
census_in, alt_term);

        ELSIF ...term tests as needed...
             ...more code as needed...
        ELSE
            -- invalid term should STOP process
```

```
            RAISE_APPLICATION_ERROR(-20000,'LOCAL=Entered invalid
Submission Term')
        END IF;
      END;
    -- LOCAL CODE ENDS --
    RETURN 'Y'; -- ignored but a return is required
*/
------------------------------------------------------------------------
----------
PROCEDURE pz_sds_alt_term(
   process_in  IN SFRYERR.SFRYERR_PROCESS_NAME%TYPE,
   term_in     IN STVTERM.STVTERM_CODE%TYPE,
   census_in   IN DATE,
   alt_term_in IN SOBYCDS.SOBYCDS_SOURCE_TERM%TYPE
);
```

Appendix I – Add records to the Degree Awarded Data Submission (DADS)

There might be the need to add students to the population being sent to System Administration, perhaps due to a degree awarded after the file has been submitted or if there was a change to a degree that would not be reflected in a current submission.

To add students to the population, use the Population Selection Extract Data form GLAEXTR.

1. The entry for the Application Field is: STUDENT
2. The entry for the Selection ID field is: SICAS_SHRYDAD_ADDTL_DEGREES
3. The entry for the Creator ID is: SATURN. IMPORTANT NOTE: Do not create your own Creator ID.
4. Perform a Next Block
5. Enter the ID number for the student(s) to be added to the population
6. Save the entries made on this form
7. Run the SHRYDAD process
8. Lock, Approve and Post the submission
9. Remove the ID numbers from GLAEXTR

Important Note: Be sure to remove the ID numbers from the Population Selection Extract Data GLAEXTR after successfully submitting, locking, approving and posting the submission. If the ID numbers are not removed from GLAEXTR, data for these ID's will be sent with the next submission and will generate fatal errors.