
Jenzabar CX

Transcript



JENZABAR

Technical Manual

Copyright (c) 2001 Jenzabar, Inc. All rights reserved.

You may print any part or the whole of this documentation to support installations of Jenzabar software. Where the documentation is available in an electronic format such as PDF or online help, you may store copies with your Jenzabar software. You may also modify the documentation to reflect your institution's usage and standards. Permission to print, store, or modify copies in no way affects ownership of the documentation; however, Jenzabar, Inc. assumes no responsibility for any changes you make.

Filename: tmtrans

Distribution date: 06/21/2001

Contact us at www.jenzabar.com

Jenzabar CX and QuickMate are trademarks of Jenzabar, Inc.

INFORMIX, PERFORM, and ACE are registered trademarks of the IBM Corporation

Impromptu, PowerPlay, Scenario, and Cognos are registered trademarks of the Cognos Corporation

UNIX is a registered trademark in the USA and other countries, licensed exclusively through X/Open Company Limited

Windows is a registered trademark of the Microsoft Corporation

All other brand and product names are trademarks of their respective companies

JENZABAR, INC.
TRANSCRIPT TECHNICAL MANUAL

TABLE OF CONTENTS

SECTION 1 – USING THIS MANUAL	1
Overview.....	1
Purpose of This Manual.....	1
Intended Audience.....	1
How to Use This Manual	1
Product Differences	1
Structure of This Manual	1
Related Documents and Help	2
Conventions Used in This Manual.....	3
Introduction.....	3
Style Conventions.....	3
Flowchart Conventions.....	4
Jenzabar CX-Specific Terms.....	5
Keystrokes.....	6
SECTION 2 – TRANSCRIPT PROCESSES	7
Overview.....	7
Introduction.....	7
Purpose of Product.....	7
Background Knowledge.....	7
Process Flow	9
Diagram	9
Process Description.....	10
Application Relationships	12
Related Jenzabar CX Applications.....	12
SECTION 3 – TRANSCRIPT TABLES AND RECORDS	13
Overview.....	13
Introduction.....	13
Alphabetical Organization.....	13
What Is an SQL Table?	13
What Is a Jenzabar CX Table?	13
What Is a Jenzabar CX Record?.....	13
Summary List of Tables and Records Used.....	14
Introduction.....	14
Impact of Changes to Tables and Records.....	14
Common Tables and Records.....	14
Shared Tables and Records.....	14
Transcript Tables and Records	16
Table and Record Relationships	17
Key to Entity Relationship Diagrams	17
Entity Relationship Diagram 1	17
Entity Relationship Diagram 2	18
Entity Relationship Diagram 3	20
Entity Relationship Diagram 4	22
Schemas.....	23
Introduction.....	23
File Naming Conventions	23
Field Descriptions	23
Schema File Reports.....	24

Introduction	24
Report Locations	24
Report Descriptions	24
Tables and Records	25
Table and Record Information	25
SECTION 4 – TRANSCRIPT MACROS AND INCLUDES	27
Overview	27
Introduction	27
Relationship Among Macros, Includes, and C Programs	27
General Installation Procedures	27
Macros	28
Introduction	28
Definition and Function	28
How to Locate Macros	28
Aplocate Program	28
Enable Macros	29
Periodic Macros	30
Table Macros	30
Includes	31
Introduction	31
Purpose	31
Macro Dependency	31
How to Locate Includes	31
Associated Includes	31
SECTION 5 – Jenzabar CX PROGRAM FILES	33
Overview	33
Introduction	33
Program Files Detailed	33
Definition File	33
Example of a def.c File	34
mac.h Files	35
Example of a mac.h File	35
SECTION 6 – TRANSCRIPT	37
Overview	37
Introduction	37
Prerequisite	37
Program Features Detailed	37
Process Flow	38
Diagram	38
Data Flow Description	38
Program Relationships	38
Tables and Records Used	38
Parameters	40
Introduction	40
Parameter Syntax	40
Parameters	40
Program Screens and Windows	43
Introduction	43
Access	43
Screen Files and Table/Record Usage	43
SECTION 7 – TRANSFER ADD	47
Overview	47

Introduction	47
Program Features Detailed	47
Process Flow	48
Diagram	48
Data Flow Description	48
Program Relationships	48
Tables and Records Used	48
Parameters	50
Introduction	50
Parameter Syntax	50
Parameters	50
Program Screens and Windows	51
Introduction	51
SECTION 8 – TRANSFER ENTRY	53
Overview	53
Introduction	53
Program Features Detailed	53
Process Flow	54
Diagram	54
Data Flow Description	54
Program Relationships	55
Tables and Records Used	55
Parameters	56
Introduction	56
Parameter Syntax	56
Parameters	56
Program Screens and Windows	57
Introduction	57
Access	57
Screen Files and Table/Record Usage	57
SECTION 9 – MENUS, SCREENS, SCRIPTS, AND REPORTS	59
Overview	59
Introduction	59
Directory Locations	59
Menu Structure	60
Introduction	60
Directories and Files that Define Menu Structures	60
Menu Source Directories	60
The Menu Description File: Example 1	60
Interpreting the menudesc File in Example 1	61
Subdirectories in the menusrc Directory	62
Contents of menusrc Subdirectories	62
Navigating the menusrc Subdirectories	62
The Menu Description File: Example 2	63
Interpreting the menudesc File in Example 2	63
Menu Option Files	63
Interpreting the Menu Option File	64
Determining the Type of Executed Process	65
Summary of Menu Source, Menu Description, and Menu Option Information	66
Menu Options	67
Programs, Screens, and Scripts	67
PERFORM (Table Maintenance) Screens	72
Introduction	72
PERFORM Screens	72

Csh Scripts	73
Introduction	73
Csh Scripts	73
ACE Reports	74
Introduction	74
ACE Reports	74
SECTION 10 – CUSTOMIZING THE TRANSCRIPT PROCESSES.....	75
Overview	75
Introduction	75
Basic Information	75
Implementation Process Checksheet	75
Cross-Functional Issues	76
Introduction	76
List Of Cross-Functional Issues	76
Assessing the Transcript Setup	81
Introduction	81
Transcript Features	81
Reviewing and Modifying Data in Tables and Records	82
Introduction	82
Procedure to Review Values	82
Table and Record Information	82
Important Transcript Tables and Records	82
Order of Table Information in This Section	82
Table Setup Sequence	82
Building the Transcript Tables	83
Introduction	83
Access	83
Transcript Selection Table	83
Transcript Exam Table	83
Transcript Form Table	83
Setting Up the Transcript Program (<i>trans</i>)	88
Introduction	88
Setting Includes for Transcript Processing	88
Variations in the Transcript Program	95
Modifying Transcripts	97
Introduction	97
How to Adapt the Transcript to Institution Policy - Reviewing the Options	97
How to Adapt the Transcript to Institution Policy - Changing the File	98
Transcript Formats - Defined	98
Transcript Format - Example	98
Transcript Format - Line Types	99
Other Transcript Form Table Fields	106
Available Tables	107
How to Modify the Format of the Transcript	107
How to Locate the Student Statistics Information	108
How to List Student Statistic Database Field Names	108
How to Locate Other Files Containing Transcript Information	108
How to Modify the Transcript Program Screen	109
How to Add a New Transcript Form	112
How to Update the Macro File and the Menu Options	112
Checking Transcript Format Modifications	113
How to Change the Default Form	113
How to Change the System Default Form	113
How to Test New Forms	114
Resolving Transcript Problems	114

How to Add Page Numbers to the Transcript.....	114
Screen Usage to Prevent Data from Printing	115
Default Attribute	116
Screen Usage to Print Blank Lines: a Non-Working Example	116
Screen Usage to Print Blank Lines: A Working Example.....	116
Screen Usage to Print Inconspicuous Character	117
Inclusion of Course Work in Progress.....	117
Transcript Sort Keys	118
Introduction	118
Types of Sorting	118
Session Sorting	118
Item Sorting	118
Group Sorting	119
Preparing Hold Codes	121
Introduction	121
How to Set up Hold Codes	121
Helpful Hints Using Hold Codes	122
Maintaining Historical Information on Session Records.....	123
Introduction	123
Setup	123
Automatic Update	124
Cut-Off Date for Updates.....	124
Update Process	125
Testing the Processes	125
Updating the Student's Classification	125
Printing Academic Statuses.....	125
Setting Up and Using Course Equivalencies	126
Introduction	126
Example of Equivalent Courses	126
Equivalent Courses in Jenzabar CX.....	126
How to Locate Course Equivalencies.....	126
Setup Process - Course Equivalencies.....	127
Example of Equivalencies Within Catalog Window.....	127
Effects of Using Course Equivalencies	127
Example of Repeat History Window	128
Tracking Course Renumbering	128
Historical Course Numbering in Jenzabar CX.....	128
Course History Setup Process	128
Example of Course History Window	129
Effects of Using Course History	129
Transfer Equivalencies in Jenzabar CX	129
Example of Transfer Equivalencies.....	129
Transfer Equivalency Setup Process	130
Example of Transfer Equivalency Window.....	130
Effects of Using Transfer Equivalencies.....	130
Setting Up and Using Repeat Logic	131
Introduction	131
Setup Process - Repeat Logic.....	131
Testing Process - Repeat Logic	132
Troubleshooting Issues - Repeat Logic.....	133
Troubleshooting Process - Repeat Logic	133
Includes - Repeat Logic.....	134
Course Record - Repeat Logic.....	135
Grade Table - Repeat Logic	137
Repeat Table - Repeat Logic	138
Repeat Table Fields	139

Repeat Table Factors in Applying Repeat Logic.....	141
Course Work Record - Repeat Logic	142
Transcript Program - Repeat Logic	142
Repeat Logic Test Cases	143
SECTION 11 – PROGRAM ERRORS AND CRASH RECOVERY.....	151
Overview.....	151
Introduction	151
Fatal Errors	151
Other Issues	151
Fatal Errors.....	152
Messages Received	152
Error and Crash Recovery Procedures	153
Introduction.....	153
Core Dump Recovery	153
INDEX	155

SECTION 1 – USING THIS MANUAL

Overview

Purpose of This Manual

This manual provides technical information required to install, customize, and maintain the Transcript product of Jenzabar CX.

Intended Audience

This manual is for use by those individuals responsible for the installation, customization, and maintenance of CX.

How to Use This Manual

If you are not familiar with the processes and features of the Transcript product, read the manual for:

- Detailed reference information about how the product works
- Procedures for customizing and maintaining the product

If you are familiar with the processes and features of the Transcript product, and just need specific reference information or a procedure, look through the table of contents or index and refer to the pages you need.

Product Differences

This manual contains information for using all features developed for the Transcript product. Your institution may or may not have all the features documented in this manual.

Structure of This Manual

This manual contains both general reference information and procedures for installing, customizing, and maintaining the Transcript product. The organization of the manual is as follows:

Overview information

- Section 1 - Information about using this manual
- Section 2 - Overview information about the product

Product reference information

- Section 3 - Tables and records used in the product
- Section 4 - Macros and Includes
- Section 5 - CX program files
- Section 6 - Transcript program (*trans*)
- Section 7 - Transfer Add program (*trnsadd*)
- Section 8 - Transfer Entry program (*trnsent*)
- Section 9 - Menus, screens, scripts, and reports

Product procedures

- Section 10 - Procedures to install and customize your processes

Error reference/Recovery procedures

- Section 11 - A reference of fatal and serious errors and recovery procedures

Reference information

Index

Related Documents and Help

The following resources are also available to assist you in installing, customizing, maintaining, and using the Transcript product.

Form Entry program (*forment*)

CX System Reference Technical Manual

Form Order table (*formord_table*)

CX System Reference Technical Manual

QuickMate online help

QuickMate Installation Guide

Getting Started User Guide

Technical manuals

CX Implementation and Maintenance Technical Manual

CX System Reference Technical Manual

Terminology

Master Glossary

UNIX-based help

Help command (<Ctrl-w>) in screens and menus

User guides

Transcript User Guide

Getting Started User Guide

Conventions Used in This Manual

Introduction

Jenzabar, Inc. has established a set of conventions to help you use this manual. The list of conventions presented below is not exhaustive, but it includes the more frequently used styles and terms.

Style Conventions

CX technical manuals observe the following style conventions.

Boldface type

Represents text that you type into the system (e.g., Type **UNDG**), command names (e.g., **Finish**), or keys you use to execute a command or function (e.g., **<Enter>**).

Bulleted lists

Show items not ranked or without a sequential performance.

CAUTION:

Indicates a caution or warning of a potential risk or condition.

<Enter>

Represents the Enter, Return, Line Feed, or ↵ key on your keyboard.

Italic type

Is used in any of these ways:

- To represent a new or key term
- To add emphasis to a word
- To cross-reference a section of text
- To represent a variable for which you substitute another variable (e.g., substitute *filename* with an appropriate filename)

<Key name>

Represents a key that you must press.

Note:

Indicates a note, tip, hint, or additional information.

Numbered lists

Show ranking of items or sequence of performance.

Parentheses

When used around a field name, indicate the field is unlabeled. The field description includes the location of the field.

Quotation marks

Represent information written in this guide exactly as it appears on the screen.

Example: The message, "Now Running..." appears.

Flowchart Conventions

Flowcharts representing a general overview of a particular application or process are included in this manual. Symbols are used in flowcharts as follows:



Process

Represents a processing function the system performs or the user must do.



Predefined Process

Represents a subroutine or module that can be called by an application; i.e., generally programming code.

Note: Used only for exception situations. The Process symbol is the standard symbol.



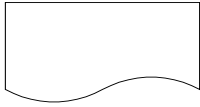
Terminator

Represents the very first or last symbol in a flowchart. Also used for references to another manual.



Display

Represents data displayed on a screen or window. Used for all menus including the master menu.

**Document**

Represents an output report, forms, or any readable data, either hardcopy or displayed on a screen.

**Stored data**

Represents data maintained in the database including tables and records.

**Manual operation**

Represents any process performed manually by the user.

**Manual input**

Represents data entered by any manual method, such as data entry, scanning, or reading bar codes.

**Connector**

Connects one section of the same flowchart to another. Usually, a connector contains a number where it exits a flowchart, and the same number in the connector showing re-entry.

Jenzabar CX-Specific Terms

Some terms used in this manual may be unfamiliar to you, either because they are terms you have not used before or because Jenzabar, Inc. has assigned a slightly different meaning to a familiar term. The following list identifies and explains the most common Jenzabar-specific terms:

Application

One or more software programs that enable you to perform a particular procedure, such as entering student information.

Data

Specific information you enter into fields on a particular data entry screen.

Enter

To type information on a keyboard and execute by any of the following actions:

- Pressing the **<Enter>** key
- Clicking on the **OK** button
- Selecting **Finish**

F key

Any of the function keys located on your keyboard (e.g., **<F1>**).

Hot key

The capitalized and underlined (or highlighted) letter of a command on a menu.

ID

The number assigned to each student or organization associated with your institution (e.g., 12345).

Parameter

A variable in the system that is given a constant value for a specific application (e.g., a date can be a parameter for producing a report).

Select

To execute a command by any of the following actions:

- Performing the keystrokes
- Pressing the hot key
- Highlighting the command or option and pressing **<Enter>**
- Clicking on the icon or button with the mouse

System

The Jenzabar, Inc. product, CX.

Keystrokes

When you see two keys separated by a dash (e.g., **<Ctrl-c>**), hold down the first key (**<Ctrl>**) while pressing the second (**<c>**).

SECTION 2 – TRANSCRIPT PROCESSES

Overview

Introduction

This section provides information on the purpose and process flow of Transcript.

Purpose of Product

The primary purpose of Transcript is to enable an institution to create and print transcripts, update course statistics, and change transcript information.

Background Knowledge

The following list describes the necessary background information that you should know to implement and support the Transcript product.

UNIX

Know the following about the UNIX operating system:

- Csh environment and commands
- Editor commands (e.g., vi)

INFORMIX-SQL

Know about the following INFORMIX tools:

- SQL database
- PERFORM screens
- ACE reports

Jenzabar CX database tools and utilities

Know how to use the following database tools:

- MAKE processor
- Schemas
- Macros
- Includes
- Program screens and windows

Jenzabar CX

Know the following about the Jenzabar CX standard product:

- CX directory structure
- The menu processor
- The CX database engine
- The product update process

QuickMate features

Know the following about the CX Graphical Server:

- Client/Server processing
- Telnet settings
- Keyboard settings
- Mouse settings
- GUI mode commands

C Programming

If you want to modify any CX programs to meet unique needs at your institution, you must know how to use the C programming language and have an in-depth knowledge of the CX code.

Transcript policies and procedures

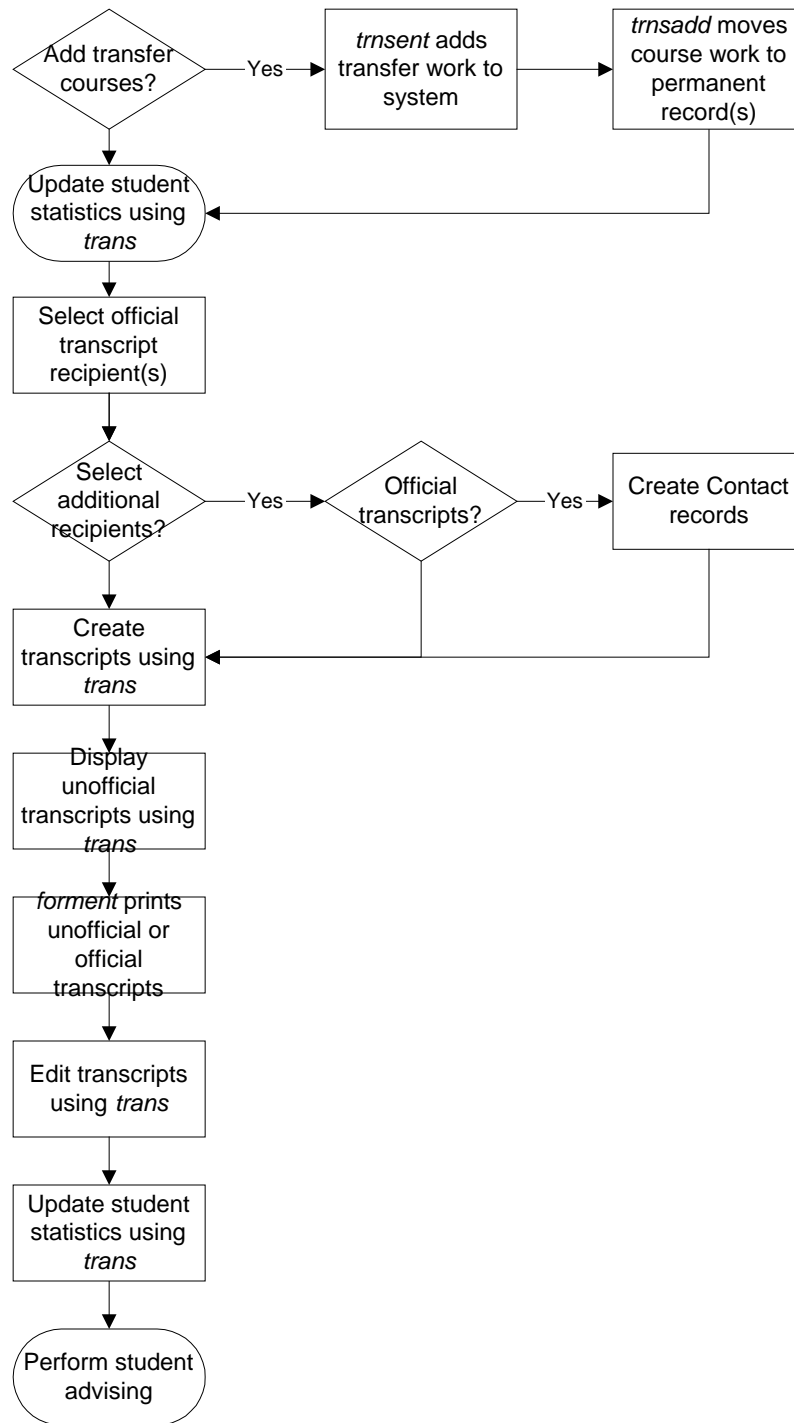
Know answers to the following questions:

Process Flow

Diagram

The following diagram shows the process flow of the Transcript product.

Note: For more information about program interrelationships and detailed data flow diagrams, see the program sections in this manual.



Process Description

The Transcript product involves the following processes.

Note: The following process description is an example and is not intended as policy guidance. The process may vary among institutions.

1. The Registrar's office uses *trnsent* to begin adding transfer work to the system either as pre-transfer courses or as transfer courses.

2. The Registrar's office uses *trnsadd* to move conditionally accepted course work to a student's permanent record so the transfer courses are available to appear on the student's transcript.
3. Update student statistics using *trans*.
4. Select one or more students to receive official transcripts.
Note: The process of selecting a student to receive an official transcript includes creating a Contact record for each recipient of an official transcript. It is not necessary to create Contact records for recipients of unofficial transcripts.
5. If processing transcripts individually, select additional recipients of the official transcript.
Note: The process of selecting an additional recipient to receive an official transcript includes creating a Contact record for each additional recipient of the official transcript(s). It is not necessary to create Contact records for additional recipients of unofficial transcripts.
6. Use *trans* to create official or unofficial transcripts either including or excluding the current term's course statistics.
7. Display unofficial transcripts using *trans*.
8. Print official or unofficial transcripts onto a form using *forment*.
9. Edit transcripts using *trans*.
10. Update student statistics using *trans*.
11. Perform student advising.

Application Relationships

Related Jenzabar CX Applications

The Transcript product interacts with several other applications and products in CX. The following list describes the interrelationships.

Grading

Users of the Grading application can interact with Transcript when using Grade Audit.

SECTION 3 – TRANSCRIPT TABLES AND RECORDS

Overview

Introduction

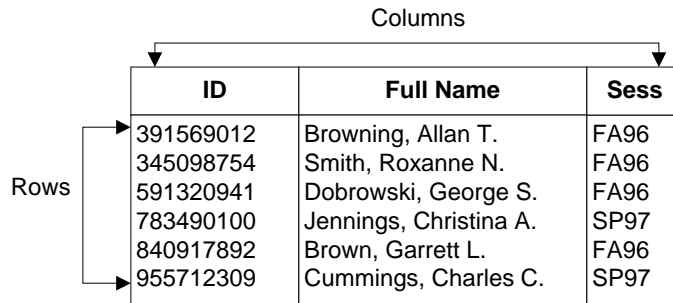
This section provides reference information about each table and record associated with the Transcript product. It also provides definitions of SQL and CX table and record terminology and differentiates among common, shared, and product-specific tables and records.

Alphabetical Organization

The tables and records appear in alphabetical order in this section.

What Is an SQL Table?

In a relational SQL database, a table is an organized set of any kind of data, regardless of its purpose for validation or information maintenance. The basic unit of organization of a table is a column, that is, a category of data. A table can have multiple columns and multiple rows of data.



The diagram shows a table with three columns and seven rows. A horizontal double-headed arrow above the table is labeled "Columns". A vertical double-headed arrow to the left of the table is labeled "Rows".

ID	Full Name	Sess
391569012	Browning, Allan T.	FA96
345098754	Smith, Roxanne N.	FA96
591320941	Dobrowski, George S.	FA96
783490100	Jennings, Christina A.	SP97
840917892	Brown, Garrett L.	FA96
955712309	Cummings, Charles C.	SP97

What Is a Jenzabar CX Table?

A *table* in CX contains information that remains static and is denoted with the *_table* extension. For example, the State table, named *st_table*, contains the list of the states in the United States of America. On the CX menu, you can access most tables in Table Maintenance menus.

What Is a Jenzabar CX Record?

A *record* in CX is a table containing information that changes on a regular basis and is denoted with the *_rec* extension. For example, the Alternate Address record, named *aa_rec*, contains any other addresses where students can be contacted, such as a summer address. You access records in CX program screens and windows, scroll screens, and PERFORM screens. SQL makes no distinction between tables and records; all sets of data are tables.

Summary List of Tables and Records Used

Introduction

Tables and records used in Transcript can be divided into the following categories:

- Common
- Shared
- Product-specific

Among these categories, some tables and records are required, while others are optional.

Impact of Changes to Tables and Records

If you make changes to schemas for any tables or records, you must reinstall each associated product or module.

Common Tables and Records

Modules in the Transcript product use several tables and records that are common throughout CX. These tables and records are:

Note: For additional information about these common tables and records see the *CX Technical Manual - Volume II (System Reference)* unless otherwise noted.

- Accomplishment record (accomp_rec)
- Accomplishment table (accomp_table)
- Addressee record (addree_rec)
- Alternate Address record (aa_rec)
- Alternate Address table (aa_table)
- Contact record (ctc_rec) (See *Communications Management Technical Manual*)
- Degree table (deg_table)
- Exam record (exam_rec)
- Exam table (exam_table)
- ID record (id_rec)
- Title table (title_table)

Shared Tables and Records

Some tables and records used in Transcript originate, or are more frequently used, within other CX product areas. These tables and records, and their originating product areas, are:

Note: For additional information, see the technical manual for the primary product area. If multiple products are shown, the primary product is listed first.

- Academic Calendar record (acad_cal_rec)
 - Course/Class Schedule
 - Registration
 - Financial Aid
- Academic record (stu_acad_rec)
 - Registration
- Academic Status table (acad_stat_table)
 - Registration
- Audit Criteria record (audcrit_rec)
 - Degree Audit
- Audit Input record (audin_rec)
 - Degree Audit

- Audit Permissions table (audperm_table)
 - Degree Audit
- Audit Requirement table (audreq_table)
 - Degree Audit
- Audit Sequence record (audseq_rec)
 - Degree Audit
- Certificate table (cert_table)
 - Registration
- Classification table (cl_table)
 - Registration
- Concentration table (conc_table)
 - Admissions
- Counting table (cntg_table)
 - Course/Class Schedule
- Course record (crs_rec)
 - Course/Class Schedule
 - Degree Audit
 - Recruiting/Admissions
- Course Work record (cw_rec)
 - Registration
 - Degree Audit
- Course Work Substitution record (cwsbst_rec)
 - Registration
 - Degree Audit
- Education record (ed_rec)
 - Financial Aid
 - Registration
 - Student Services
- Grade table (grd_table)
 - Grading
- Grading table (grdg_table)
 - Grading
- Major table (major_table)
 - Registration
- Minor table (minor_table)
 - Registration
- Noncatalog record (non_cat_rec)
 - Degree Audit
- Noncatalog Equivalency record (noncateq_rec)
 - Degree Audit
- Pre-Noncatalog record (prencat_rec)
 - Degree Audit
- Pre-Transfer Course Work record (precw_rec)
 - Degree Audit
- Profile record (profile_rec)
 - All products
- Program Enrollment record (prog_enr_rec)
 - Registration
 - Financial Aid
 - Student Services
- Program table (prog_table)
 - All products

- Registration record (reg_rec)
 - Registration
- Repeat table (rep_table)
 - Course/Class Schedule
 - Grading
 - Registration
- School record (sch_rec)
 - Admissions
 - Financial Aid
 - Registration
 - Student Services
- Section record (sec_rec)
 - Course/Class Schedule
- Session table (sess_table)
 - Course/Class Schedule
- Student Services record (stu_serv_rec)
 - Admissions
 - Financial Aid
 - Student Affairs
 - Student Billing
- Student Statistics record (stu_stat_rec)
 - Registration
 - Student Services
- Subaudit table (subaud_table)
 - Degree Audit
- Subcriteria record (subcrit_rec)
 - Degree Audit
- Subsession table (subsess_table)
 - Course/Class Schedule

Transcript Tables and Records

Programs in the Transcript product use the following Transcript tables and records. File information about these tables and records is in this section.

- Block Transfer record (blktw_rec)
- Transcript Comment record (trans_comm_rec)
- Transcript Exam table (transexam_table)
- Transcript Form table (transfrm_table)
- Transcript Selection table (transsel_table)

Table and Record Relationships

Key to Entity Relationship Diagrams

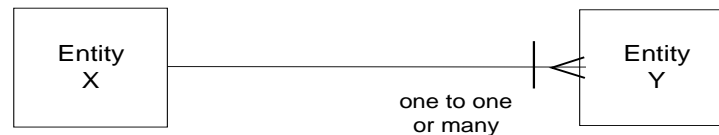
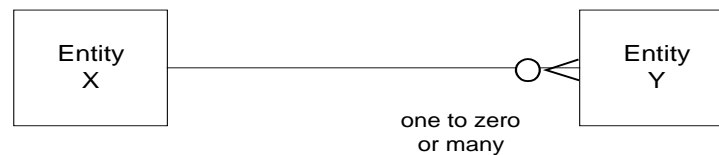
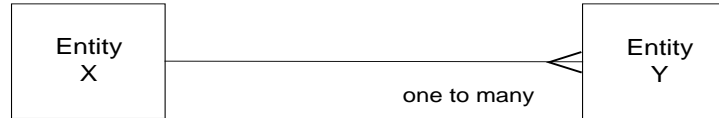
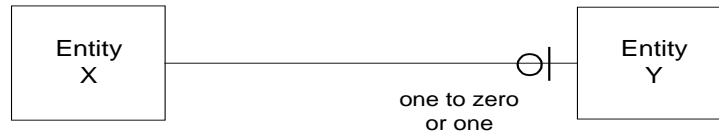
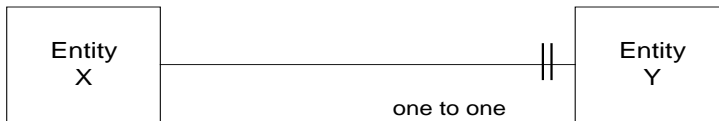
This diagram provides a key to interpreting the entity relationships in these diagrams.

Interpret the diagrams as follows:

- Entity X has a one to (a) one or many relationship to Entity Y.
- Entity Y has a one to (b) one relationship to Entity X.



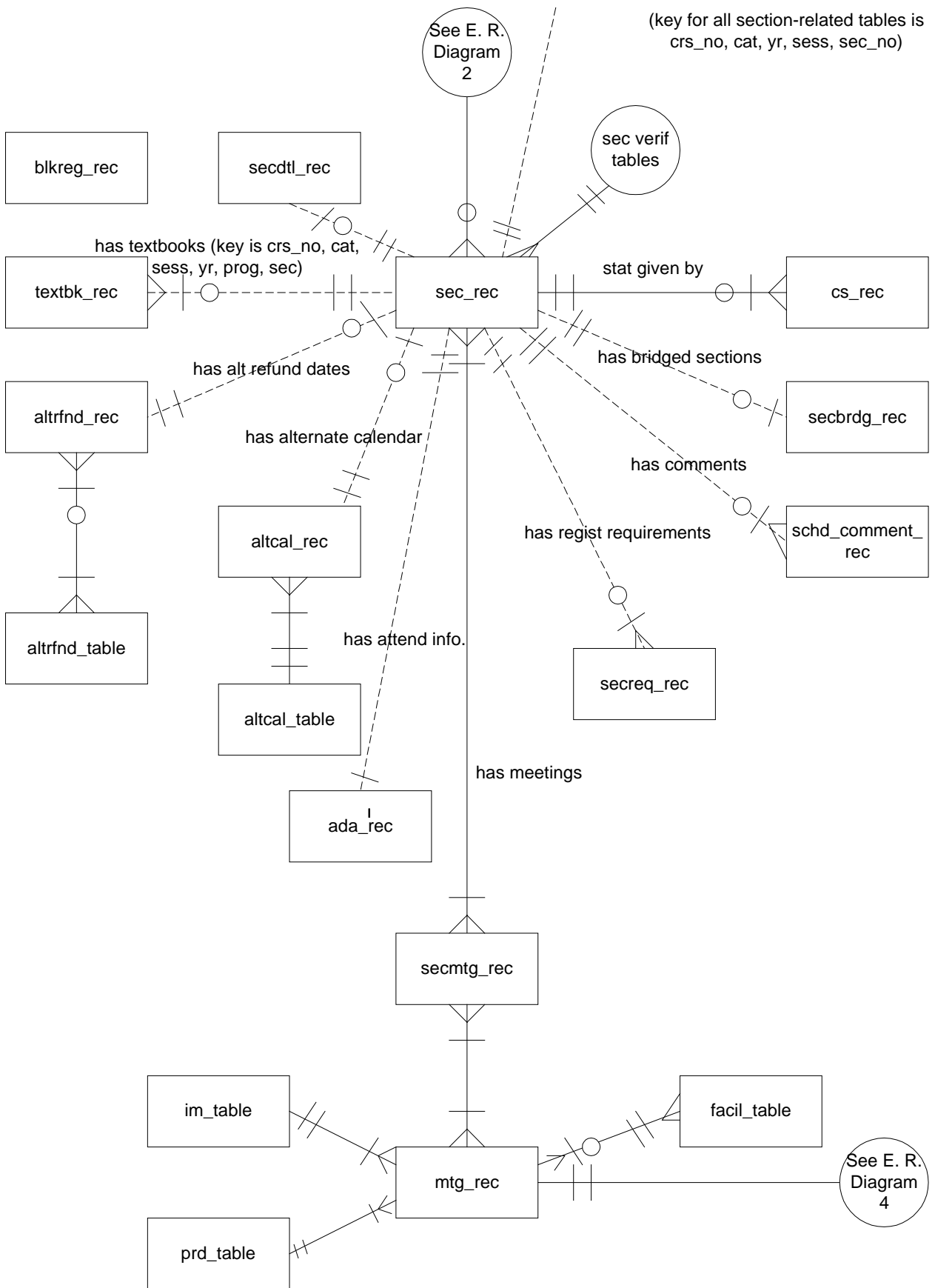
Relationship symbols and their meanings:



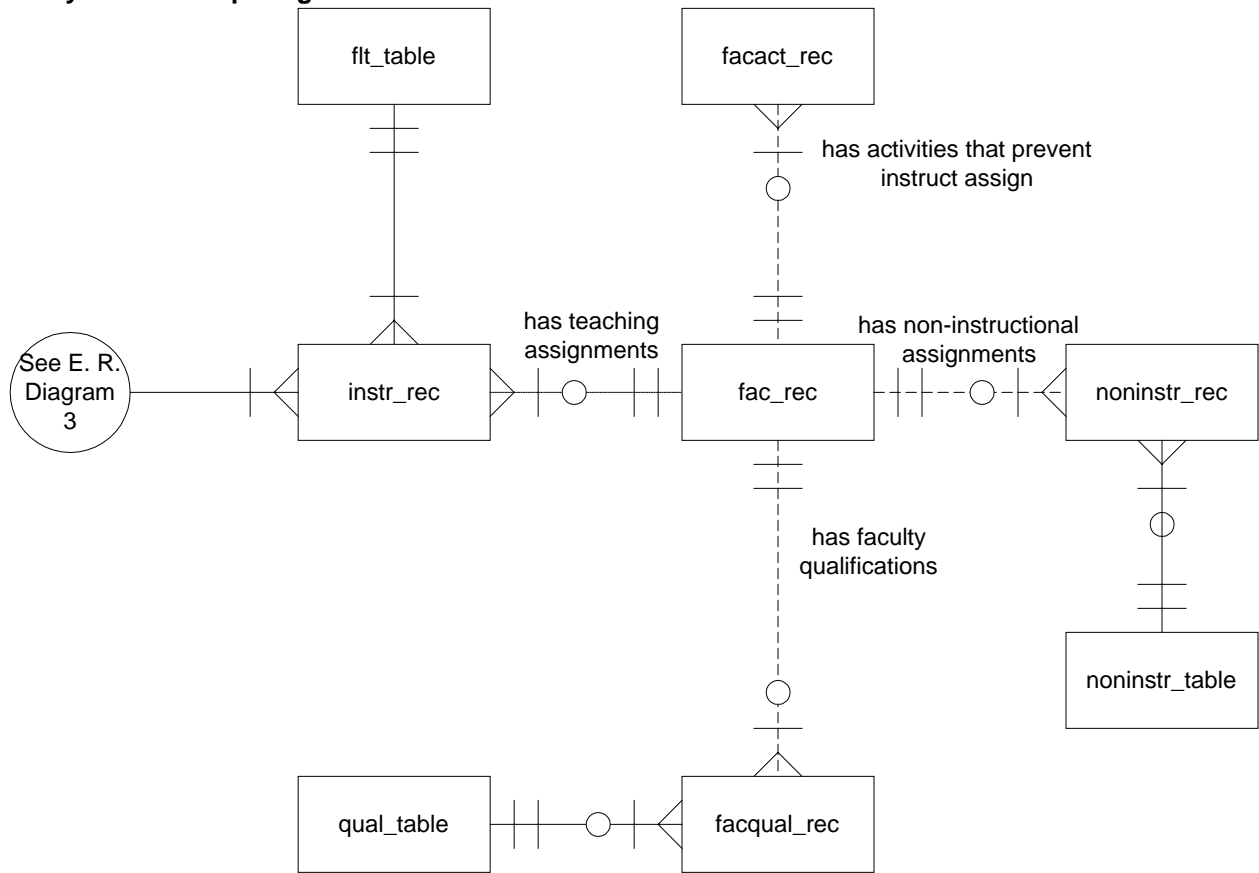
Entity Relationship Diagram 1

The following diagrams show the relationship between the tables and records the Transcript product uses.

Entity Relationship Diagram 3



Entity Relationship Diagram 4



Schemas

Introduction

Schema files define the structure of database files and associated fields in the CX data dictionary. You can access schema files associated with the Transcript product in the following directory path: `$CARSPATH/schema/student`.

File Naming Conventions

CX makes distinctions in the naming of schemas. For schema files containing definitions of CX tables, the UNIX filename begins with the letter *t* followed by characters describing the English name of the table (e.g., *tst* for the State table). For schema files containing definitions of CX records, the UNIX filename describes the English name of the record (e.g., *id* for the ID record).

The first line in a schema file, after revision information, specifies the INFORMIX database table that the schema defines. For example, *st_table* (State table) is specified in the *tst* schema file.

Field Descriptions

Schema files contain descriptions of each field defined in a table or record. You can view descriptions of fields in Transcript tables and records by accessing the schema files.

Schema File Reports

Introduction

The standard CX product includes three reports that provide information about the tables and records used in each product area. When table implementation begins, you can run the reports to provide the installation team with the most current information about the tables and records, and the columns in each.

Report Locations

Access the reports from the System Management: Data Dictionary menu.

Report Descriptions

The following three reports provide information that is useful during the table setup phase of implementation:

Database Fields for the Jenzabar CX Database Dictionary

Report filename: *dbefield*

Menu option: Fields by File Report

Description: This report lists each column in the specified range of files (e.g., *id_rec* to *profile_rec*), including its name, short and long descriptions, field type, and size.

Database Files for the Jenzabar CX Database Dictionary

Report filename: *dbefile*

Menu option: Fields by Track Report

Description: This report lists each table that relates to the specified range of product areas (e.g., A for Admissions to F for Financial), including the table name, description, and purpose.

Database Files/Fields for the Jenzabar CX Database Dictionary by Track

Report filename: *dbetrack*

Menu option: File/Field by Track Report

Description: This report combines the contents of *dbefield* and *dbefile*, displaying the tables for the specified product areas (e.g., A for Admissions to F for Financial) and the columns in each table.

Tables and Records

Table and Record Information

The following list identifies the tables and records that originate from the Transcript product. The list includes the filenames, location, purpose, and association of each table and record with programs, products, and other tables and records.

Note: The *Program interrelationships* in the list are included in the Transcript product. The *Product interrelationships* in the list are not included in the Transcript product.

Block Transfer record

This record is used in Grade Audit (GA) conversion for storage of block transfer courses. These are groups of transfer courses that can be calculated in the overall GPA. It tracks a student's transfer course information as a block of credit (e.g., adding to the transcript 12 credit hours for 4 courses of 3 credits each), instead of adding course work course-by-course (e.g., adding to the transcript 4 courses of 3 credits each course-by-course).

UNIX filename: blktw

Informix filename: blktw_rec

Schema location: \$CARSPATH/schema/student

Program interrelationships: trans

Product interrelationships: Admissions, Registration

Table/record interrelationships: Transcript Form table

Transcript Comment record

Contains transcript comments to be printed on official and unofficial transcripts.

UNIX filename: transcomm

Informix filename: trans_comm_rec

Schema location: \$CARSPATH/schema/student

Program interrelationships: trans

Product interrelationships: Grading

Table/record interrelationships: Associated with the Transcript Form table.

Transcript Exam table

Used to identify exam types to be included on the header of a transcript with a given selection value.

UNIX filename: transexam

Informix filename: transexam_table

Schema location: \$CARSPATH/schema/student

Program interrelationships: trans

Product interrelationships: None

Table/record interrelationships: None

Transcript Form table

Stores the screen form names for transcript output.

Notes:

- The pathname for the screen description for this table is \$CARSPATH/modules/regist/screens/ttransfrm.
- Before using this table, all forms that are used must be present in the following path: \$CARSPATH/modules/regist/forms/trans
- This table must have at least one unofficial transcript for display to your terminal screen, one unofficial transcript to print interactively (two column) or from Contact records, and one official transcript (two column) to print only from Contact records.
- Normally, this table is modified only if the forms are renamed or if additional forms are added. This usually occurs when an institution has two or more academic programs and there is a requirement to have a different transcript form for undergraduate and graduate programs.
- The printed copies of the transcript (both official and unofficial) are always two columns. It is possible to change the width of the columns and the space between the two columns in this table.
- It is possible to remove certain screens (lines) from a transcript by removing them from the appropriate Transcript Form table entry.
- You can produce grade change mailers from this table.

CAUTION: Test any changes carefully because there may be undesirable side effects.

UNIX filename: ttransfrm

Informix filename: transfrm_table

Schema location: \$CARSPATH/schema/student

Program interrelationships: trans, forment

Product interrelationships: Admissions, Grading

Table/record interrelationships: Associated with the Academic Calender record, Academic record, Accomplishment record, Block Transfer record, Course record, Course Work record, Education record, Exam record, Section record, Student Statistics record, and Transcript Comment record.

Transcript Selection table

Contains valid transcript selection values.

UNIX filename: transsel

Informix filename: transsel_table

Schema location: \$CARSPATH/schema/student

Program interrelationships: trans

Product interrelationships: Admissions, Registration

Table/record interrelationships: None

SECTION 4 – TRANSCRIPT MACROS AND INCLUDES

Overview

Introduction

This section provides reference information about macros and includes used to set up the Transcript product.

Relationship Among Macros, Includes, and C Programs

For all elements of the product other than C programs, m4 macros contain the definitions of features that have been designed to be modified for each institution. These macros, located under \$CARSPATH/macros, are passed through the m4 processor.

Jenzabar CX contains an alternative method for the setting of features in C programs. Macros in the source code of C programs are not passed through the m4 processor because the C compiler has its own preprocessor, the cc.

To provide the same apparent functionality to C programs, a section in the include source directory has been allocated for a type of include file that acts as an intermediary between the m4 processor and the cc preprocessor. In operation, m4 macros are defined whose output is a valid cc macro. These m4 macros are placed in the include files. Then the include files are translated and the appropriate cc macro is placed in the include file. The C compiler includes the installed include file during compilation so that the m4 macro influences the result of the compilation.

General Installation Procedures

See the *Implementation & Maintenance Technical Manual*, for general procedures on setting and installing changes to macros and includes.

Macros

Introduction

CX contains macros that define specific values used throughout the Transcript product. The macros and includes enable you to change the available options and functionality of the Transcript product without having to modify C code. By modifying macros, you can customize your implementation of the Transcript product and make the product easier to maintain.

Definition and Function

A macro is an instruction that causes the execution of a pre-defined sequence of instructions in the same source language. A macro consists of uppercase letters and underscores, and is used in place of a text string within source files. CX expands the macro to the longer text during the installation process for a file. CX uses the following kinds of macros:

- Enable - Allows you to enable a feature of CX
- DBS_COMMON - Allows you to define database values in screens
- Periodic - Allows you to make changes on a periodic basis

Macros can perform one of the following functions:

- Define defaults on a screen (_DEF)
- Define valid values in a field (_VALID or _INCL)
- Enable system modules (ENABLE_MOD)
- Enable system features (ENABLE_FEAT)
- Establish a valid value for an include

How to Locate Macros

To modify Transcript macros, access the \$CARSPATH/macros/custom/student file. To access Common macros, access the \$CARSPATH/macros/custom/common file.

Applocate Program

You also can locate macros using the *Applocate* program. *Applocate* checks the descriptions of macro files for the product you specify and lists each file that it locates in a separate file in your home directory.

Note: To locate the macros used in Transcript, using *Applocate*, you must specify the product name.

Follow these steps to run the *Applocate* program.

1. Select Utility Menu from the CX menu. The Utilities: Main menu appears.
2. Select File Options. The Utilities: File Options menu appears.
3. Select Locate Macro Values. The Locate Macro Values screen appears.
4. Select **Table Lookup** in the Macro Category field. A list of module names appears in a Table Lookup box.
5. Select a module name (e.g., ACADEMIC/HISTORY), and click **OK**. The Table Lookup box disappears.
6. Select **Finish**. The Output Parameters and Scheduling window appears.
7. Enter the following:
 - In the Time field, enter **NOW**.
 - In the Background field, enter **Y**.

8. Select **Finish**.

The system creates the file, *allocate.out*, and sends it to your home directory.

Enable Macros

The following lists the Transcript enable macros located in the *student* macro file.

'ACAD_TRANS_VALID','F,P,N,W'

'ACAD_TRANS_INCL','include=(ACAD_TRANS_VALID),upshift'

'ACAD_TRANS_EX','(F)inal, (P)artial, (N)one, (W)aived.'

Define the transcript status for the ed_rec:

- F (Final)
- P (Partial)
- N (None)
- W (Waived)

Note: Normally, no change is required. Admissions also uses this macro. Update this macro only if there are additional transcript types they wish to record on Education (ed_rec) records.

'CTC_REG_GRDRPT_DEF','GRDRPT'

'CTC_REG_GRDRPT_INCL','include=(GRDRPT,"GRDRPT/M"),upshift'

'CTC_REG_GRDRPT_EX','GRDRPT or GRDRPT/M'

'CTC_REG_TRANS_DEF','TRANS'

'CTC_REG_TRANS_INCL','include=(WIDE,MORE,TRANS)'

Define the contact resource, transcript, and grade report defaults and includes.

Notes:

- Update these macros as necessary according to the forms used by the registrar. Generally, they do not require changes unless new forms are introduced.
- Especially for new transcript forms, ensure they are consistent with the TRANS_ in \$CARSPATH/macros/custom/table and the transfrm_table.
- In \$CARSPATH/menuopt/regist/programs/reg.o there is a PA=PPCTC_RESRC line. Be aware that the corresponding PA line allows for only the valid codes of SDS. If SDS80 is to be used, make the appropriate change. The same approach is necessary in reg.ob for SDSBILL.

'ENABLE_FEAT_BLK_RESIDENT','N'

Defines whether to display the menu option allowing input of resident work on the Registrar: Transcripts menu. The menu option enables one to put all previous transcript work in as a summary consisting only of totals. The macro's default setting, N, specifies that this menu option does not appear on the Registrar: Transcripts menu.

Note: If enabled, this macro allows the registrar to enter block resident course work from a PERFORM screen. You can enter block transfer work using the Transcript Edit function of Transcript. You normally accomplish entering course work by this method during the conversion process.

The following lists the Form Entry enable macros located in the *common* macro file.

'FORMENT_ALT_ID_NUMBER','92394'

Defines which ID (id_rec) record to use to temporarily store ID information on an individual who is to receive forms, but who does not have an ID record already in the database. Form Entry allows you to specify an alternate ID (an ID other than the ID for whom the form is

generated) without that individual actually having an `id_rec` in the database. To provide that functionality, however, Form Entry requires an ID record that is used as temporary storage space for this alternate recipient name and address data. This macro specifies the ID number of that record. Make sure that the ID number you assign here is specially created only for Form Entry.

'FORMENT_STU_EXIT_PASSWORD','CARS'

Indicates the password you must use to run *forment* in the Student mode. To exit *forment* from the Student mode, press **<Ctrl-n>**, then enter this password.

'FORMENT_PERM_CATEGORY','FORMENT'

'FORMENT_PERM_CODE','OVERRIDE'

Defines the Permission table (`perm_table`) category and permission code for allowing the operator to override the charges for transcript orders.

Periodic Macros

Transcript has no periodic macros located in the *periodic* macro file.

Table Macros

The following lists the Transcript table macros located in the *table* macro file.

'TRANSFRM_DEF', 'WIDE'

'UNOFF_TRANSFRM_DEF', 'UNOF'

Define the default for the transcript form code.

Note: Update as necessary. The form must be consistent with an entry in the Transcript Form table (`transfrm_table`). `WIDE` is the accepted form for the official transcript in the standard product. Use `transfrm_def` for official transcripts, and use `unoff_transfrm_def` for unofficial transcripts.

Includes

Introduction

The Transcript product contains includes, which determine the features that are enabled in the product. An include can be either a compile option that enables or disables a feature, or a default value for a feature.

To enable a feature in the Transcript product, you must define an include in `$CARSPATH/include/custom`. To disable an include, comment out the include in the same file. See the *CX System Reference Technical Manual*, for more information on enabling and disabling includes. By modifying includes, you can customize your implementation of the Transcript product and make the product easier to maintain.

Purpose

An include allows you to activate or deactivate features in C programs without changing the C code. You can also specify compilation values for an entry program in the Transcript product.

Macro Dependency

Includes have a dependency on macros. Normally, you do not directly modify includes for the product. You must modify a corresponding macro value and then reinstall the include.

How to Locate Includes

To locate a Transcript include, access the `$CARSPATH/include/custom` directory.

Note: For more information about using the MAKE processor and modifying includes, see the *CX System Reference Technical Manual*.

Associated Includes

The following list describes the modifications that may be required for the Transcript includes.

trans

Contains includes that affect the format and content of the transcript, including (for example):

- Printing course and/or section titles
- High school code values used in the `sch_type` field
- Formatting when using 1/6 hour courses

trans.out

Contains includes that define reason codes put in the `reg_recs` and added when a `cw_rec` is added or voided in Transcript Edit.

SECTION 5 – Jenzabar CX PROGRAM FILES

Overview

Introduction

This section provides reference information about the files that relate to most CX programs. By understanding the file structure and the contents of the files, you can locate most of the information you need about any program.

Program Files Detailed

This section contains details about the following files:

Note: All other files for each CX program are standard C programming files with standard components and structure.

def.c

The `def.c` file contains the declaration of external variables (including structures) that must be available to all source files in the program. These variables can also be initialized in this file. As with other C source files, the files also contain comments. The **makedec** command uses the `def.c` file to create the `dec.h` file.

mac.h

The `mac.h` file contains preprocessor include and define statements, typedef statements, and structure template definition statements. The file also contains macro substitution defines and declarations of structures. This file is included in all source files during compilation through use of the `dec.h` file.

Definition File

Every program uses a definition (`def.c`) file, located in the following paths:

- `$CARSPATH/src/regist/trans`
- `$CARSPATH/src/regist/trnsadd`
- `$CARSPATH/src/regist/trnsent`
- `$CARSPATH/src/common/forment`

The `def.c` file for a screen-oriented program can contain the following information:

- Includes for a `mac.h` file
- Declaration of global variables and structures used throughout the program
- Structure and non-structure screen binds (i.e., program buffer to screen buffer binds)
- Ring menu definitions
- Prompt line information
- Program parameters
- Declarations of dynamic memory (`dmms`, `dmls`, and `dmlts`) in relation to functionality within `libdmm` (the dynamic memory management package)
- Screen pointers

The `def.c` file for a non-screen-oriented program can contain the following information:

- Includes for a `mac.h` file
- Global program variables
- Includes for schema files `def.c` files
- Form pointers that provide the location for forms
- `Sqllda` pointers that bind the file structure to the form
- `dmm`, `dml`, and `dmlt` definitions
- Program parameters
- Declarations of functions so the compiler can handle a call of that function

Example of a def.c File

The following is an edited excerpt from the def.c file for the Transcript program (*trans*). It illustrates the common components of a standard CX def.c file.

<pre>#include "mac.h"</pre>		1
<pre>#include <schema/common/aadef.c> #include <schema/common/accompdef.c></pre>		2
<pre>/* Screen label fields - initialize only if they are defined */ #if LABEL_FIELDS_NO > 0 char *label_flds[LABEL_FIELDS_NO][3] = LABEL_FIELDS; #else /* LABEL_FIELDS_NO */ char ***label_flds; #endif /* LABEL_FIELDS_NO */</pre>		3
<pre>FILE *fpt; /* FILE pointer for temp file */ FILE *errfpt; /* FILE pointer for error file */</pre>		4
<pre>struct tstx_type tst; /* Transcript line structure */ struct sst_type sst; /* Transcript session structure */ struct sst_type *psst; /* Transcript session pointer */</pre>		5
<pre>struct param_type param_array[] = { { 'L', sitecode, PRM_CHAR, sizeof(sitecode) -1, PRM_REQUIRED, "site", "Site value"}, { 'P', progcode, PRM_CHAR, sizeof(progcode) -1, NULL, "Program", "Academic program code" }, }</pre>		6
<pre>/* ----- ----- */ define the run menu SCR_MENUSTART(run_menu) SCR_MENUOPT2(0,"Print",'P',SCR_GMENABLE, "Print a transcript.", "Print"), SCR_MENUOPT2(0,"eDit",'D',SCR_GMENABLE, "Edit a session of a student's trans cript.", "Crscat"), SCR_MENUOPT2(0,"Update",'U',SCR_GMENABLE, "Update student academic and stude nt status records.", "Update"), SCR_MENUOPT2(0,"Exit",'E',SCR_GMENABLE, "Exit program.", "Exit"), SCR_MENUEND;</pre>		7

Note: The legend for the file contents directly follows this example.

Legend for the def.c file:

1. mac.h include
2. schema file def.c's
3. screen label fields
4. pointers
5. structure definitions
6. parameters
7. menu definitions

mac.h Files

Every program uses a macro header (mac.h) file, located in the following paths:

- \$CARSPATH/src/regist/trans
- \$CARSPATH/src/regist/trnsadd
- \$CARSPATH/src/regist/trnsent
- \$CARSPATH/src/common/forment

The *mac.h* file for a screen-oriented program can contain the following information:

- Includes related to system header files
- Includes related to CX library and other application processes
- Includes for schema files mac.h files
- Program constant definitions (i.e., *#define* statements)
- Structure definitions

Example of a mac.h File

The following is an edited excerpt from the mac.h file for *trans*. It illustrates the common components of a standard CX mac.h file.

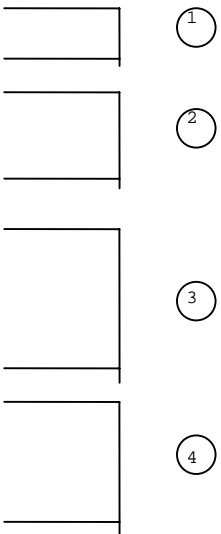
```
#include <ctype.h>
#include <memory.h>
#include <stdio.h>

#include <util/trn.h>
#include <applic/common.h>

#include <schema/common/accompmac.h>
#include <schema/common/ctcmac.h>

----- */
#define CW_NO_SECTITLE_FMT(cwno) "%012dX ", cwno
#define CRS_MIN_ID          ' '
#define CRS_LINE_ID         'U'
#define CRS_TITLE2_ID       'V'
#define CRS_TITLE3_ID       'W'
#define CRS_SECTITLE_ID     'X'

struct score_type
{
    long    subscore;           /* exam subscore */
    char    score_date[5];     /* date exam was taken */
    char    ctgry[5];          /* exam type */
};
~
```



Note: The legend for the file contents directly follows this example.

Legend for the mac.h file:

1. includes for header files
2. includes for schema files
3. program constant definitions
4. structure definitions

SECTION 6 – TRANSCRIPT

Overview

Introduction

This section provides reference information about the Transcript program (*trans*). The Transcript product uses the Transcript program to produce transcripts for an institution. CX recomputes a student's academic history each time it produces a transcript, or when it runs the process in batch mode.

The Transcript program uses the Transcript Form table (*transfrm_table*) to define the type of form that a particular transcript should use. The screens used in the Transcript Form table to produce the transcript are located in the directory path `$CARSPATH/modules/regist/forms/trans`.

Prerequisite

CX applies repeat logic in various ways based on the decisions of an institution. Jenzabar, Inc. recommends that an institution review the repeat logic before using *trans*.

Note: For more information about repeat logic, see *Setting Up and Using Repeat Logic* in the section *Customizing the Transcript Processes* in this technical manual.

Program Features Detailed

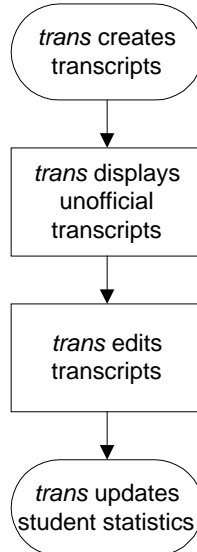
This section contains details about the following features of the Transcript program:

- Process flow
- Parameters
- Program screens and windows

Process Flow

Diagram

The following diagram shows the flow of data in the Transcript program.



Data Flow Description

The following describes the data flow in the Transcript program.

1. The Transcript program accesses the Form Entry program to create either official or unofficial transcripts for a student based on the student's `stu_acad_rec`.

Note: The Transcript program creates official transcripts only for students who have a Contact record (`ctc_rec`) for a transcript.

2. The Transcript program displays unofficial transcript(s) (`stu_acad_rec`).
3. The Edit function of the Transcript program edits both official and unofficial transcript(s) (`stu_acad_rec`).
4. The Transcript program updates student statistics (`stu_stat_rec`).

Program Relationships

The following programs use Transcript.

- *Libreg*
- *libsql*
- *libtrn*
- *sysbuff*
- *sysline*

Tables and Records Used

The Transcript program uses the following Common and Transcript tables and records.

Note: For information about the Common tables and records, see the *CX System Reference Technical Manual*. For information about the Transcript tables and records, see the section *Transcript Tables and Records* in this manual.

Common tables and records

- aa_rec
- aa_table
- accomp_rec
- accomp_table
- addree_rec
- ctc_rec
- ed_rec
- exam_rec
- exam_table
- id_rec
- profile_rec
- sch_rec
- title_table

Transcript tables and records

- acad_cal_rec
- acad_stat_table
- audperm_table
- blktw_rec
- cert_table
- cl_table
- cntg_table
- conc_table
- crs_rec
- cw_rec
- deg_table
- grd_table
- grdg_table
- major_table
- minor_table
- non_cat_rec
- noncateq_rec
- prog_enr_rec
- prog_table
- reg_rec
- rep_table
- sec_rec
- sess_table
- stu_acad_rec
- stu_serv_rec
- stu_stat_rec
- subsess_table
- trans_comm_rec
- transfrm_table
- transsel_table
- transexam_table

Parameters

Introduction

CX contains parameters and compilation values for executing the Transcript program (*trans*). You can specify parameters to compile *trans* in a specified manner at the time of execution.

Note: You can also specify compilation values with the includes for the Transcript product that affect the Transcript program.

Parameter Syntax

You can display Transcript parameters by entering the following: **trans -,**

The following is the correct usage for running the Transcript program from the UNIX shell:

```
trans -L site [-p Program] [-s Session] [-y Year] [-a Runcode] [-C] [-c] [-d] [-D Due Date] [-e] [-f Form] [-h] [-i ID] [-I ADR ID] [-o] [-P Device] [-r Rcpt ID] [-R resrc] [-t Trans Select] [-S] [-u] [-Y Academic Year] [-U Course Number] [-T Catalog] [-N Section Number]
```

Parameters that appear in brackets are optional. Parameters that do not appear in brackets are required.

Parameters

The following lists the parameters for running *trans*.

-L site

Required - Specifies the site value.

Example: trans -L CARS

-p Program

Optional - Specifies the academic program code.

Example: trans -p UNDG

-s Session

Optional - Specifies the academic session.

Example: trans -s FA

-y Year

Optional - Specifies the year associated with the specified session.

Example: trans -y 2000

-a Runcode

Optional - Specifies the alternate address runcode.

Example: trans -a TRANS

Note: Using this option requires that an entry exists in the ADR Table (*adr_table*) with the specified alternate address runcode.

-C

Optional - Specifies to include the current session on the transcript.

-c

Optional - Specifies to produce transcripts from Contact records.

-d

Optional - Specifies to access *trans* in display-only mode.

-D Due Date

Optional - Specifies the Contact record due date used to select transcripts.

Example: trans -D 12/05/2000

Note: The “-D” parameter is optional in that you can use the Contact record in CX to generate transcripts.

-e

Optional - Specifies to print Education records (ed_rec) on transcripts.

-f Form

Optional - Specifies the transcript form code.

Example: trans -f UNOF

Note: The form code must exist in the Transcript Form table (transfrm_table).

-h

Optional - Specifies that the system must ignore student holds.

-i ID

Optional - Specifies the student ID number.

Example: trans -i 12345

Note: The “i” parameter is optional. Use it to produce a single transcript by ID number.

-I ADR ID

Optional - Specifies the alternate address ID by number to use on the transcript.

Example: trans -I 12345

Notes:

- The value determines which addree_rec to use (e.g., -I 12345) in which the value is the ID number of the individual running the request to print transcripts. This value is the ID number associated with the id_used_by value in the addrec_rec.
- The parameter is optional and is used to select an alternate name to be used on the transcript.

-o

Optional - Specifies to print official transcripts.

-P Device

Optional - Specifies the printer device code.

Example: trans -P lpr

-r Rcpt ID

Optional - Specifies the recipient ID number.

-R resrc

Optional - Specifies the contact resource code for printing transcripts.

Example: trans -R TRANS

Note: This option is used by ADR processing in areas outside of transcripts (e.g., admissions) and is not recommended for use with transcript requests.

-t Trans Select

Optional - Specifies the selection code the system uses to select courses to include.

-S

Optional - Specifies to restrict printing to one student at a time.

Note: The “-S” parameter is optional; however, the Form Entry program uses it to produce transcripts.

-u

Optional - Specifies to update student statistics.

-Y Academic Year

Optional - Specifies the academic year to use for year-to-date statistics.

-U Course Number

Optional - Specifies the course number of the course for students whose statistics the system is to update.

-T Catalog

Optional - Specifies the catalog of the course for students whose statistics the system is to update.

-N Section Number

Optional - Specifies the section number of the course for students whose statistics the system is to update.

Program Screens and Windows

Introduction

The Transcript program (*trans*) has screens and windows for performing the following interactive functions:

- Edit transcripts
- Print transcripts
- Update student statistics

Access

The screen and window files for *trans* are located in the following directory paths:

- \$CARSPATH/src/regist/trans/SCR

Notes:

- You can access windows from each program screen in *trans*.
- See the *CX System Reference Technical Manual* for information about common windows that appear in Transcript.

Screen Files and Table/Record Usage

The Transcript screens and windows appear in the following files and use the indicated tables and records.

editaccomp

Contains the Accomplishment panel.

Access: \$CARSPATH/src/regist/trans/SCR

Tables/Records:

- accomp_rec
- accomp_table

editblktw

Contains the Block Transfer panel.

Access: \$CARSPATH/src/regist/trans/SCR

Tables/Records:

- blktw_rec

editcrs

Contains the Resident, Transfer, and Waiver panel.

Access: \$CARSPATH/src/regist/trans/SCR

Tables/Records:

- cntg_table
- crs_rec
- cw_rec
- grd_table
- grdg_table
- rep_table

editexam

Contains the Exam panel

Access: \$CARSPATH/src/regist/trans/SCR

Tables/Records:

- exam_rec
- exam_table

editline

Contains the

Access: \$CARSPATH/src/regist/trans/SCR

Tables/Records:

- sess_table

editncat

Contains the Noncatalog panel.

Access: \$CARSPATH/src/regist/trans/SCR

Tables/Records:

- cntg_table
- crs_rec
- crsctgry_table
- cw_rec
- degapply_table
- grd_table
- grdg_table
- non_cat_rec
- rep_table

editncat-new

Contains the Noncatalog panel.

Access: \$CARSPATH/src/regist/trans/SCR

Tables/Records:

- cntg_table
- crs_rec
- crsctgry_table
- cw_rec
- degapply_table
- grd_table
- grdg_table
- non_cat_rec
- rep_table

editsec

Contains the Section Title panel.

Access: \$CARSPATH/src/regist/trans/SCR

Tables/Records: None

edittc

Contains the Comment panel.

Access: \$CARSPATH/src/regist/trans/SCR

Tables/Records:

- trans_comm_rec

help

Contains the student registration commands (e.g., Edit, Print, Update).

Access: \$CARSPATH/src/regist/trans/SCR

Tables/Records: None

progenr

Contains the Program Enrollment Records portion of the Create/Edit Transcripts screen and the Display Transcripts screen.

Access: \$CARSPATH/src/regist/trans/SCR

Tables/Records:

- prog_enr_rec

prompt

Contains the Create/Edit Transcripts screen and the Display Transcripts screen minus the Program Enrollment Records portion.

Access: \$CARSPATH/src/regist/trans/SCR

Tables/Records:

- id_rec
- prog_table
- sess_table
- transfrm_table
- transsel_table

SECTION 7 – TRANSFER ADD

Overview

Introduction

This section provides reference information about the Transfer Add (*trnsadd*) program . The Transcript product uses Transfer Add to move conditionally accepted course work to a student's permanent record and thus make it available for the transcript.

Program Features Detailed

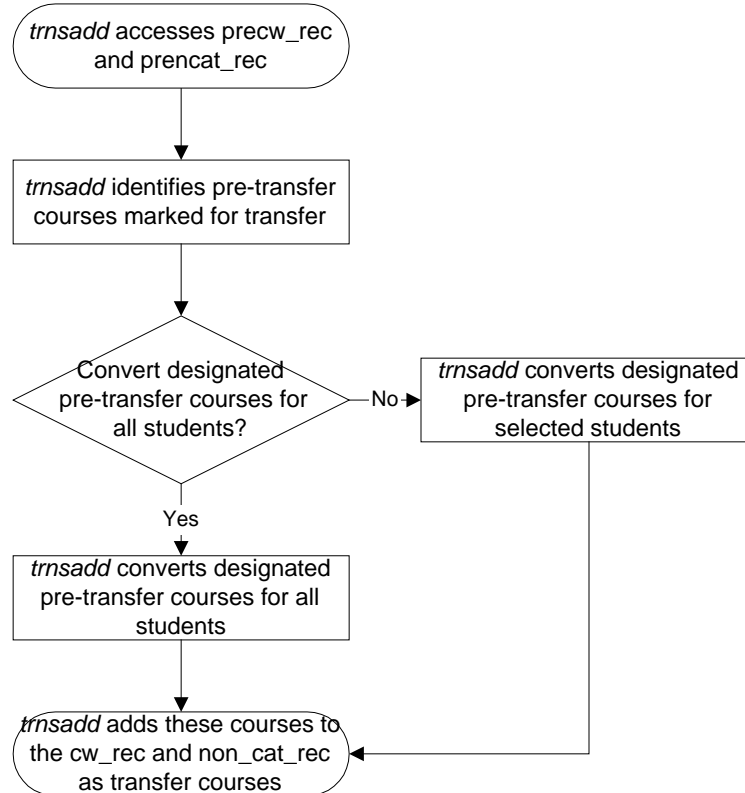
This section contains details about the following features of the Transfer Add program:

- Process flow
- Parameters
- Program screens and windows

Process Flow

Diagram

The following diagram shows the flow of data in the Transfer Add program.



Data Flow Description

The following describes the data flow in the Transfer Add program.

1. The Transfer Add program accesses the *precw_rec* and *prencat_rec* of a student
2. The Transfer Add program identifies pre-transfer courses marked for transfer for the selected student(s) where the Move Date field is blank.
3. The Transfer Add program converts these designated pre-transfer courses either for all students or for selected students and adds them to the *cw_rec* and *non_cat_rec* of the selected student(s) as transfer courses.

Program Relationships

The following programs use the Transfer Add program.

- *Libreg*

Tables and Records Used

The Transfer Add program uses the following Common and Transcript tables and records.

Note: For information about the Common tables and records, see the *CX System Reference Technical Manual*. For information about the Transcript tables and records, see the section *Transcript Tables and Records* in this manual.

Common tables and records

Note: The Transfer Add program uses no common tables and records.

Transcript tables and records

- cw_rec
- non_cat_rec
- precw_rec
- prencat_rec

Parameters

Introduction

CX contains parameters and compilation values for executing the Transfer Add program. You can specify parameters to compile Transfer Add in a specified manner at the time of execution.

Note: You can also specify compilation values with the includes for the Transcript product that affect the Transfer Add program.

Parameter Syntax

You can display Transfer Add parameters by entering the following: **trnsadd -**,

The following is the correct usage for running the Transfer Add program from the UNIX shell:

trnsadd [-i List of ID Number(s)]

Parameters that appear in brackets are optional. Parameters that do not appear in brackets are required.

Parameters

The following lists the parameters for running *trnsadd*.

-i List of ID Number(s)

Optional - Specifies the identification numbers of students.

Program Screens and Windows

Introduction

The Transfer Add program (*trnsadd*) has no screens and windows because it is a batch program.

SECTION 8 – TRANSFER ENTRY

Overview

Introduction

This section provides reference information about the Transfer Entry program (*trnsent*). The Transcript product uses Transfer Entry to enter a student's transfer work and create Transfer Equivalency records.

Program Features Detailed

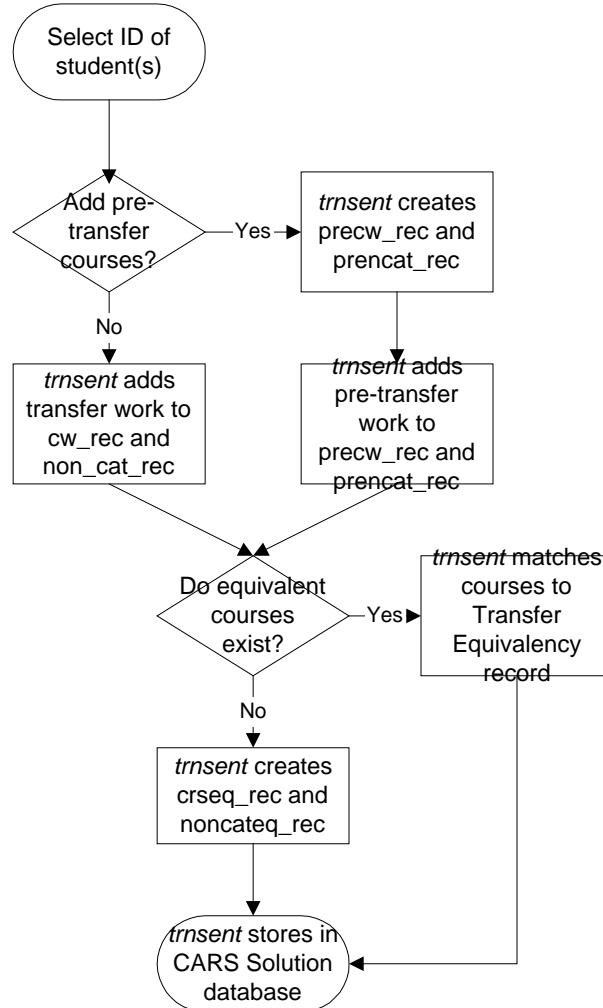
This section contains details about the following features of the Transfer Entry program:

- Process flow
- Parameters
- Program screens and windows

Process Flow

Diagram

The following diagram shows the flow of data in the Transfer Entry program.



Data Flow Description

The following describes the data flow in the Transfer Entry program.

1. The user selects the ID of the student(s) and decides whether to add transfer work as pre-transfer courses.
2. If the user adds transfer work as pre-transfer courses, *trnsent* adds the transfer work to a *precw_rec* and *prencat_rec* it creates for the student.
3. If the user adds transfer work as transfer courses, *trnsent* adds the transfer work directly to the student's *cw_rec* and *non_cat_rec*.
4. If equivalent courses for the transfer work do not exist in a Transfer Equivalency record, the Transfer Entry program creates a *crseq_rec* and *noncateq_rec*.
5. If equivalent courses for the transfer work do exist in a Transfer Equivalency record, the

Transfer Entry program matches the courses to the Transfer Equivalency record.

6. The Transfer Entry program stores the transfer work in the CX database.

Program Relationships

The following programs use Transfer Entry.

- *Libaud*
- *Libreg*

Tables and Records Used

The Transfer Entry program uses the following Common and Transcript tables and records.

Note: For information about the Common tables and records, see the *CX System Reference Technical Manual*. For information about the Transcript tables and records, see the section *Transcript Tables and Records* in this manual.

Common tables and records

- exam_rec
- exam_table
- id_rec
- profile_rec

Transcript tables and records

- audcrit_rec
- audin_rec
- audperm_table
- audreq_rec
- audseq_rec
- crs_rec
- cw_rec
- cwsubst_rec
- non_cat_rec
- noncateq_rec
- precw_rec
- prencat_rec
- prog_enr_rec
- stu_acad_rec
- stu_stat_rec
- subaud_table
- subcrit_rec

Parameters

Introduction

CX contains parameters and compilation values for executing the Transfer Entry program. You can specify parameters to compile Transfer Entry in a specified manner at the time of execution.

Note: You can also specify compilation values with the includes for the Transcript product that affect the Transfer Entry program.

Parameter Syntax

You can display Transfer Entry parameters by entering the following: **trnsent -**,

The following is the correct usage for running the Transfer Entry program from the UNIX shell:

trnsent [-d] -p prog -L site [-c] [-m] [-a]

Parameters that appear in brackets are optional. Parameters that do not appear in brackets are required.

Parameters

The following lists the parameters for running *trnsent*.

-d

Optional - Specifies to access *trnsent* in display-only mode.

-p prog

Required - Specifies the code of the academic program to which the system enters transfer courses.

-L site

Required - Specifies the site of the transfer student.

-c

Optional - Specifies entry of actual transfer work.

-m

Optional - Specifies to update the move flag for pre-transfer work.

-a

Optional - Specifies to add Transfer Equivalency records.

Program Screens and Windows

Introduction

The Transfer Entry program has screens and windows for performing the following interactive functions:

- Adding pre-transfer courses to a Pre-Transfer Course Work record (precw_rec)
- Adding transfer courses to a student's transcript

Access

The screen and window files for Transfer Entry are located in the following directory paths:

- \$CARSPATH/modules/regist/progscr/trnsent

Notes:

- You can access windows from each program screen in Transfer Entry.
- See the *CX System Reference Technical Manual* for information about common windows that appear in Transcript.

Screen Files and Table/Record Usage

The Transfer Entry program screens and windows appear in the following files and use the indicated tables and records.

trnstu

Contains the Add Pre-Transfer Courses - Pre-Transfer Course Entry screen and the Add Transfer Courses - Transfer Course Entry screen.

Access: \$CARSPATH/modules/regist/progscr/trnsent

Tables/Records:

- id_rec

SECTION 9 – MENUS, SCREENS, SCRIPTS, AND REPORTS

Overview

Introduction

This section provides reference information on the following features of the Transcript product:

- Menu source files
- Menu option files
- PERFORM screens
- SQL scripts
- Csh scripts
- ACE reports
- Letters

Directory Locations

The features detailed in this section are located in the following directory paths:

Menu source files

\$CARSPATH/menusrc/student/regist/transcr
\$CARSPATH/menusrc/student/regist/transcr/transfer

Menu option files

\$CARSPATH/menuopt/common/programs
\$CARSPATH/menuopt/regist/programs
\$CARSPATH/menuopt/regist/reports
\$CARSPATH/menuopt/regist/screens
\$CARSPATH/menuopt/regist/scripts
\$CARSPATH/menuopt/utilities/programs

PERFORM screens

\$CARSPATH/modules/regist/screens

SQL scripts

None

Csh scripts

\$CARSPATH/modules/regist/scripts

ACE reports

\$CARSPATH/modules/regist/reports

Letters

None

Menu Structure

Introduction

CX menus provide access to a functionally related group of menu options. For example, a CX user in the Admissions office can access all the options necessary for processing prospects, recruits, and applicants from his/her menu, but typically cannot access options for processing accounting information. Depending on the work responsibilities of the CX users at your institution, you can customize their menus to offer access to as many or as few menu options as desired.

The selections that appear on the CX menus at your institution are controlled by a variety of interrelated directories and files. This section explains how the directories and files define CX menus, and it also provides information about the standard CX menu options.

Directories and Files that Define Menu Structures

The directories and files that control the standard CX menus are:

- Menu source directories
- Menu description files
- Menu option files

Menu Source Directories

Menu source (menusr) directories define the branches of menus and submenus that offer access to CX features. The highest level of the menu source directory, located at \$CARSPATH/menusr, contains files and subdirectories similar to the following:

admit/	student/
fiscal/	system/
instdev/	utility/
menudesc	

Note: In this example, six subdirectories (designated with a slash [/] after their names) and one file appear in the menusr directory.

The Menu Description File: Example 1

Menusr directories always contain a menu description (menudesc) file. The menudesc file defines the options available at the current menu level. To continue the example above, the menudesc file in the \$CARSPATH/menusr directory contains the following type of information:

```

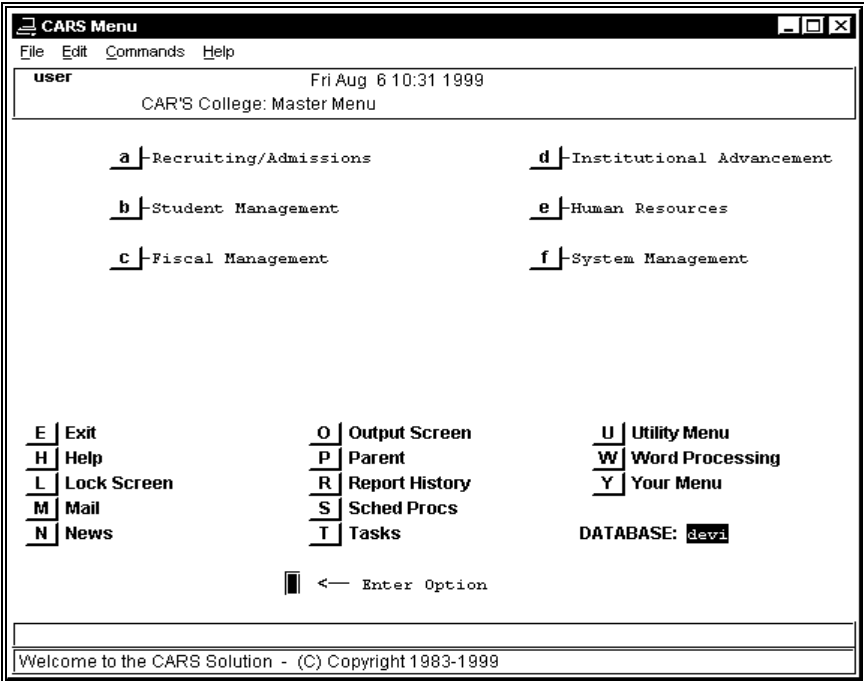
#
#Revision Information (Automatically maintained by 'make' - DON'T CHANGE)
#-----
#Header:
#Log:
#
#-----
#
#
TI=INST_NAME: Master Menu
SD=CARS Solution Master Menu
LD=Master Menu For INST_NAME
m4_keepif(ENABLE_MOD_ADMISSIONS, `Y')
MNU_SUB(admit)
m4_keepend
MNU_SUB(student)
MNU_SUB(fiscal)
m4_keepif(ENABLE_MOD_DEVELOP, `Y')
MNU_SUB(instdev)
m4_keepend
MNU_SUB(system)

```

Interpreting the menudesc File in Example 1

- In this example, the menudesc file has the following information:
- Comment lines (lines beginning with a pound sign [#])
 - Title and descriptive information (lines beginning with TI [title], SD [short description], or LD [long description])
 - Keepif lines (lines indicating that a particular submenu is available if a corresponding ENABLE macro is set to Y)
 - Keepend lines (lines indicating the end of a macro-controlled option)
 - MNU_SUB lines (lines showing the name of the submenu)

Specifically, this example defines a menu called <institution name>: Master Menu. Assuming your institution has purchased the Admissions and Development applications and has enabled the ENABLE_MOD_ADMISSIONS and ENABLE_MOD_DEVELOP macros, your Master Menu would contain the following menu options:



Note: The names used for each of the submenus are controlled in their respective submenus.

Subdirectories in the menusrc Directory

Typically, the menusrc directory for a particular menu also contains subdirectories that relate to that menu's submenu options. In the example of the Master Menu, the following subdirectories are in the main \$CARSPATH/menusrc directory, and they relate directly to the menu options that appear on the preceding menu screen example.

admit/

Recruiting/Admissions submenu

fiscal/

Fiscal Management submenu

instdev/

Institutional Advancement submenu

student/

Student Management submenu

system/

System Management submenu

utility/

Utility submenu (accessed from the lower part of the CX menu screen)

Contents of menusrc Subdirectories

The menusrc subdirectories have the same components as the main menusrc directory used in the preceding examples. Each contains subdirectories for more submenus, as well as a menudesc file that defines the appearance and contents of the submenu itself.

Navigating the menusrc Subdirectories

By using UNIX commands to move into any of the menusrc subdirectories and viewing its contents (i.e., its own subdirectories, if any, and its related menudesc file), you can view and map the menus, submenus, and menu options available at your institution. To continue the preceding example, if you enter **cd utility** at the UNIX prompt while in the \$CARSPATH/menusrc directory, you can then enter **lsf** to display the contents of the utility subdirectory, as in the following:

Makefile	data/	iq/	ltrblrep/	spooler/
RCS/	document/	login/	menudesc	
adr/	files/	ltrlbl/	sbscr/	

If you then enter **cd data** at the UNIX prompt and execute the **lsf** command again, you display the contents of the data subdirectory, as in the following:

Makefile	RCS	menudesc
----------	-----	----------

In this example, only a menudesc file appears, indicating that no submenus exist at the \$CARSPATH/menusrc/utility/data menu source level. The menudesc file in this case contains the actual menu options you can run from the Utilities/Data Entry menu. If desired, you can view the menudesc file using *vi*, *more*, or other UNIX commands.

The Menu Description File: Example 2

The menudesc file in Example 1 contains references only to submenus, that is, each line in the menudesc file that begins with MNU_SUB refers to a submenu. Another type of line can appear in the menudesc file: the MNU_OPT line. MNU_OPT lines do not refer to submenus; instead, they provide the link from the menu to the actual process (e.g., a program, script, report, or screen) you can run. For example, the menudesc file in the \$CARSPATH/menusr/utility/data directory contains the following type of information:

```
#
#Revision Information (Automatically maintained by 'make' - DON'T CHANGE)
#-----
#$Header: menudesc,v 8.1 97/10/03 15:34:36 jdoe Released $
#-----
#
TI=Utilities: Data Entry Menu
SD=Data Entry
LD=Data Entry
PW=@STD
MNU_OPT(common/programs/ide)
```

Interpreting the menudesc File in Example 2

In this example, the menudesc file has the following information:

- Comment lines (lines beginning with a pound sign [#])
- Title and descriptive information (lines beginning with TI [title], SD [short description], or LD [long description])
- Password information (the line beginning with PW)
- MNU_OPT line (the line showing the location of the menu option itself)

Notes:

- As with the menudesc file in Example 1, this file also can contain Keepif and Keepend lines to control the availability of optional processes.
- Although the menudesc file in Example 2 contains only one menu option (common/programs/ide), menudesc files can contain as many options as you care to display on a single menu.

Menu Option Files

The menudesc file for a particular menu or submenu points to the specific menu options you can run (e.g., Example 2 referred to the single menu option common/programs/ide, indicating that a common program is the only option available from the selected submenu).

To locate the menu option files for a particular menu, use the following command syntax:

cd menuopt/<directory on menudesc line>

For Example 2, the appropriate command is:

cd menuopt/common/programs

When you list the contents of the directory, the list resembles the following:

```
Makefile          ctcbatchr        ide              tuserid
RCS/              evtnt            ide,fa
adrtest           fo.op            perm.a
ctcbatch          fo.stu           perm.c
```

This directory example contains several menu option files that are linked to other menudesc files. By their names, you can usually determine the process the menu option executes. For example, ide and ide.fa run versions of *identry*, and fo.op and fo.stu run versions of *formnt*. More than

one version of these processes exist because, depending on the needs of each menu user, you may want to use different titles or parameters.

You can view the specific menu option file using *vi*, *more*, or other UNIX commands.

Interpreting the Menu Option File

The menu option file contains three primary types of information:

- A definition of the parameter screen that appears when a user selects the option from the menu, including information for comment lines
- The name of the screen, program, script, or report the menu option executes
- The parameters, if any, that determine how the menu option should execute

An excerpt from the *ide* file demonstrates these three types of information. The horizontal lines on the excerpt divide the types of information and have been added to help you interpret the file's contents.


```

----- Parameter screen -----
}
screen
{
          m4_center_clipped(ID DATA ENTRY,40)
          m4_center_clipped(PP_OFFICE[PA5], 40)
          m4_center_clipped(PP_TICK[PA7], 40)
}
end
attributes
SD: optional,
   default = "ID Data Entry";
PW: optional,
   default = "@STD";
RD1: optional,
   default = "`Functions:'           ";
RD2: optional,
   default = "";
RD3: optional,
   default = "`Enter ID Records for Individuals' ";
RD4: optional,
   default = "Enter ID Records for Non-individuals";
PR: optional,
-----Executed Process-----
   default = "BIN_PATH/identry";
-----Processing Parameters-----
m4_keepif(ENABLE_FEAT_AUTOMODE, `Y')
PA1: optional,
   default = "-a";
m4_keepeelse
m4_keepif(ENABLE_FEAT_FORCEQUERY, `Y')
PA1: optional,
   default = "-F";
m4_keepeend
m4_keepeend
PA2: optional,
   default = "-D";
PA3: optional,
   default = "3";
PA4: optional,
   default = "-o";
LU5 = ofc_table.txt, optional;
PA5: optional,
   comments = "COMMENT_OFC_TBCODE COMMENT_TBL",
   length = 4,
   lookup LU5 joining *ofc_table.ofc,
   upshift;
PA6: optional,
   default = "-T";
LU7 = tick_table.txt, optional;
PA7: optional,
   comments = "COMMENT_TICK COMMENT_TBL",
   length = 4,
   lookup LU7 joining *tick_table.tick,
   upshift;
end

```

Determining the Type of Executed Process

In the preceding example, the location of the menu option (\$CARSPATH/menuopt/common/programs) indicates that the menu option executes a program. The reference to BIN_PATH within the menu option file also indicates the execution of a program. However, many menu options execute other types of processes (e.g., screens, scripts, or reports).

The following list shows the type of process executed, the _PATH reference from the menu option file, and the type of menu option directory path in which the process resides.

Process type	_PATH reference	Menu option directory path
Program	BIN_PATH	/programs
C shell script	SCP_PATH	/scripts
Report	ARC_PATH OTH_PATH	/reports /others
SQL statement	INF_PATH	/informers
PERFORM screen	SCR_PATH FRM_PATH	/screens
Utility	UTL_PATH	/utilities

Summary of Menu Source, Menu Description, and Menu Option Information

When you need to modify any menu options in use at your institution, remember the following:

- Menu source directories reflect your CX menus and submenus.
- Menu source directories contain other menu source subdirectories and menu description files.
- Menu description files define the menu options that appear on each menu.
- Menu description files contain lists of the menu options and submenus.
- Menu option files contain the instructions needed to execute each CX process on every menu.

Menu Options

Programs, Screens, and Scripts

The following list associates each Transcript program, screen, and script menu option and corresponding menuopt file and identifies the menuopt locations and what the menu option accesses.

Note: The following menus and options are listed in the order in which they appear on the standard CX menu structure. Italicized parameters indicate those that a user can enter or change.

Registrar: Transcripts Menu

Course Work Evaluation

Note: This menu option is available if the macro ENABLE_FEAT_MULTI_SITE is set to Y.

Accesses: \$CARSPATH/modules/regist/reports/crseval (ACE report)

File: \$CARSPATH/menuopt/regist/reports/crseval

Parameters Passed:

- *Session*
- *Academic Year*
- *Program*
- *ID#*

Create/Edit Transcripts

Accesses: \$CARSPATH/src/regist/trans (Program)

File: \$CARSPATH/menuopt/regist/programs/trns

Parameters Passed:

- -f *form* (Name of desired form instead of using a menu)
- -L *site* (Site code to be used)
- -e (Print education records on transcript)

Display Transcripts

Accesses: \$CARSPATH/src/regist/trans (Program)

File: \$CARSPATH/menuopt/regist/programs/trns.d

Parameters Passed:

- -d (Display only mode)
- -f *form* (Name of desired form instead of using a menu)
- -L *site* (Site code to be used)

Enter Block Resident Work

Note: This menu option is available if the macro ENABLE_FEAT_BLK_RESIDENT is set to Y.

Accesses: \$CARSPATH/modules/regist/screens/stuacad (PERFORM screen)

File: \$CARSPATH/menuopt/regist/screens/stuacad

Parameters Passed: None

Update All Student Stats

Accesses: \$CARSPATH/src/regist/trans (Program)

File: \$CARSPATH/menuopt/regist/programs/trns.Cu

Parameters Passed:

- -C (Include current session on transcript)
- -u (Update statistics)
- -Y *academic year* (Financial aid academic year used for year-to-date statistics)
- -L *site* (Site value)

Update Select Student Stat

Accesses: \$CARSPATH/src/regist/trans (Program)

File: \$CARSPATH/menuopt/regist/programs/trns.Cusy

Parameters Passed:

- -C (Include current session on transcript)
- -u (Update statistics)
- -Y *academic year* (Financial aid academic year used for year-to-date statistics)
- -s *session* (Session code)
- -y *year* (Calendar year)
- -L *site* (Site value)

Update Student Stat By Crs

Accesses: \$CARSPATH/src/regist/trans (Program)

File: \$CARSPATH/menuopt/regist/programs/trns.CuUT

Parameters Passed:

- -C (Include current session on transcript)
- -u (Update statistics)
- -Y *academic year* (Financial aid academic year used for year-to-date statistics)
- -s *session* (Session code)
- -y *year* (Calendar year)
- -L *site* (Site value)
- -T *catalog* (Course catalog of students to select to update statistics)
- -U *course number* (Course number of students to select to update statistics)
- -N *section number* (Course section number of students to select to update statistics)

Operator Form Request

Accesses: \$CARSPATH/src/common/forment (Program)

File: \$CARSPATH/menuopt/common/programs/fo.op

Parameters Passed:

- -s *session* (Indicates the session in which the processes are run)
- -y *year* (Indicates the year in which the processes are run)
- -p *program* (Indicates the program in which the processes are run)
- -L *sitecode* (Indicates the site in which the processes are run)
- -o (Indicates the operator mode, as opposed to the student mode)

Select Transcripts - One

Accesses: \$CARSPATH/modules/regist/screens/transctc (PERFORM screen)

File: \$CARSPATH/menuopt/regist/screens/transctc

Parameters Passed: None

Select Transcripts - All

Accesses: \$CARSPATH/modules/regist/scripts/transctc (Csh script)

File: \$CARSPATH/menuopt/regist/scripts/transctc

Parameters Passed:

- *Session*
- *Academic Year*
- *Program*
- *Contact*
- *Due Date*

Create Official Trans.

Accesses: \$CARSPATH/src/regist/trans (Program)

File: \$CARSPATH/menuopt/regist/programs/trns.cop

Parameters Passed:

- -c (Produce transcripts from contacts)
- -o (Print official transcripts)
- -p *program* (Program code)
- -f *form* (Transcript form code)
- -L *site* (Site value)
- -R *resrc* (Contact resource code for printing transcripts)
- -D *due date* (Contact record due date used in selecting transcripts)

Create Official Trans/Curr

Accesses: \$CARSPATH/src/regist/trans (Program)

File: \$CARSPATH/menuopt/regist/programs/trns.Ccop

Parameters Passed:

- -C (Include current session on transcript)
- -c (Produce transcripts from contacts)
- -o (Print official transcripts)
- -p *program* (Program code)
- -f *form* (Transcript form code)
- -R *resrc* (Contact resource code for printing transcripts)
- -D *due date* (Contact record due date used in selecting transcripts)
- -L *site* (Site value)

Create Unofficial Trans.

Accesses: \$CARSPATH/src/regist/trans (Program)

File: \$CARSPATH/menuopt/regist/programs/trns.cp

Parameters Passed:

- -c (Produce transcripts from contacts)
- -p *program* (Program code)
- -f *form* (Transcript form code)
- -L *site* (Site value)
- -R *resrc* (Contact resource code for printing transcripts)
- -D *due date* (Contact record due date used in selecting transcripts)

Create Unoff Trans/Curr

Accesses: \$CARSPATH/src/regist/trans (Program)

File: \$CARSPATH/menuopt/regist/programs/trns.Ccp

Parameters Passed:

- -C (Include current session on transcript)
- -c (Produce transcripts from contacts)
- -p *program* (Program code)
- -f *form* (Transcript form code)
- -L *site* (Site value)
- -R *resrc* (Contact resource code for printing transcripts)
- -D *due date* (Contact record due date used in selecting transcripts)

Print Transcripts

Note: This menu option is available if the macro ENABLE_FEAT_LPS is set to Y.

Accesses: \$CARSPATH/src/util/lps (Utility)

File: \$CARSPATH/menuopt/utilities/programs/lps.trns

Parameters Passed: None

Registrar: Transfer Work Processing Menu

Add Transfer Courses

Accesses: \$CARSPATH/src/regist/trnsent (Program)

File: \$CARSPATH/menuopt/regist/programs/trne

Parameters Passed:

- -L *site* (Site of transfer student)
- -p *prog* (Program for in student transfer work)
- -c (Indicates entry of actual transfer work)
- -a (Allows adding of Transfer Equivalency records)

Add Pre-Transfer Courses

Accesses: \$CARSPATH/src/regist/trnsent (Program)

File: \$CARSPATH/menuopt/regist/programs/trne.pre

Parameters Passed:

- -L *site* (Site of transfer student)
- -p *prog* (Program for in student transfer work)
- -a (Allows adding of Transfer Equivalency records)
- -m (Allows updating of the move flag for pre-transfer work)

Print Student PreTrans Crs

Accesses: \$CARSPATH/modules/regist/reports/stucrsp (ACE report)

File: \$CARSPATH/menuopt/regist/reports/stucrsp

Parameters Passed:

- *ID#*

Move Pre-Transfer Courses

Accesses: \$CARSPATH/src/regist/trnsadd (Program)

File: \$CARSPATH/menuopt/regist/programs/trna

Parameters Passed: None

Move Pre-Trans By Student

Accesses: \$CARSPATH/src/regist/trnsadd (Program)

File: \$CARSPATH/menuopt/regist/programs/trna.i

Parameters Passed: None

PERFORM (Table Maintenance) Screens

Introduction

Transcript uses PERFORM screens for displaying maintenance tables and some records. You can access the screen files in the following directory path: \$CARSPATH/modules/regist/screens.

PERFORM Screens

Transcript uses the following PERFORM screens. Descriptions of PERFORM screens include:

- Purpose of the screen
- Tables and records used in the screen
- Master/detail relationships, if applicable

Student Academic Record screen

Enables you to add previous resident course work as a block of credit. Use the Resident option in Transcript Edit to enter the resident course work individually. The course work must be entered individually to use the Program Audit module.

Menu Access: Registrar: Transcripts Menu: Enter Block Resident Work option

File: stuacad

Tables/Records Used:

- cl_table
- ctry_table
- id_rec
- prog_table
- st_table
- stu_acad_rec
- title_table

Transcript Contact/Contact Table screen

Enables you to create a Contact record for a recipient of an official transcript, which is the first step in producing an official transcript. It also enables you to display and update Contact records of an individual.

Menu Access: Registrar: Transcripts Menu: Select Transcripts - One option

File: transctc

Tables/Records Used:

- ctc_rec
- ctc_table
- ctry_table
- id_rec
- st_table
- title_table

Csh Scripts

Introduction

Transcript contains Csh scripts to automate the processing of information. Csh scripts are UNIX-based program statements that can execute a series of SQL scripts or ACE reports. The Transcript Csh scripts are located in the following directory path:
\$CARSPATH/modules/regist/scripts.

Csh Scripts

The following list associates a Transcript menu option with the corresponding Csh script and provides a description of the script. Descriptions of Csh scripts include:

- Purpose of the script
- Menu access option, if applicable
- A list of ACE reports used, if applicable
- A list of SQL scripts used, if applicable
- A list of tables used, if applicable

transctc

Runs the transctc report, pipes the output to ISQL to set up contacts, and mails possible error messages to the user.

Menu Access: Registrar: Transcripts Menu: Select Transcripts - All

Reports: None

ACE Reports

Introduction

CX contains ACE reports for easy reporting of Transcript database information.

ACE Reports

Transcript provides the following ACE reports. Some reports listed do not appear on the CX menu system because they are used only in Csh scripts.

Course Work Evaluation Report

Provides a list of all courses for all sites for a person to determine whether any courses should be evaluated for transfer to the current institution.

Menu Access: Registrar: Transcripts menu: Course Work Evaluation option

File: \$CARSPATH/modules/regist/reports/crseval

Student Preliminary Transfer Course Report

Provides a list of courses taken by a specified student and classified as pre-transfer courses. These are courses taken at another institution that are in the process of being considered for transfer to this institution. A degree audit may be run against the pre-transfer courses.

Menu Access: Registrar: Transfer Work Processing menu: Print Student PreTrans Crs option

File: \$CARSPATH/modules/regist/reports/stucrsp

SECTION 10 – CUSTOMIZING THE TRANSCRIPT PROCESSES

Overview

Introduction

This section provides procedures for setting up and installing the features of the Transcript product. The following procedures are included:

- Assessing institutional needs for the module
- Reviewing and modifying data in tables and records
- Setting up the Transcript program
- Changing default field values in macros
- Modifying transcripts
- Adding page numbers to transcript format
- Reference information about sort keys in the Transcript program
- Preparing hold codes
- Maintaining historical information on Session records
- Setting up and using course equivalencies
- Setting up and using repeat logic

Basic Information

This section contains detailed procedures specific to the Transcript product. For information on performing basic procedures, such as using the MAKE processor and reinstalling options, refer to the following resources:

- *Database Tools and General Utilities* course notebook (SYS200)
- *CX System Reference Technical Manual*

Implementation Process Checklist

The implementation process checklist, *Implementing Grading and Transcript*, shows the general phases in the process to customize and install features of Transcript. The checklist defines the tasks you must complete to install the module and the correct sequence for the tasks.

Use the procedures in this section to help you complete your implementation of Transcript.

Cross-Functional Issues

Introduction

As you implement Transcript, various policy issues will arise about which you must make decisions. However, in addition to issues affecting only the Registrar's office, there are other issues that involve various offices at an institution. The following are some of these issues, as well as helpful information you can use in deciding how to resolve each issue.

List Of Cross-Functional Issues

The following lists each cross-functional issue, as well as a description of each, that should be addressed while implementing Transcript.

Use of course, section, and reference numbers

To ensure that the original intent of an institution's numbering system remains intact, answer the following questions:

- Does the meaning of the course number indicate academic year (e.g., freshman)? Does it include the credit value or some other value?
- What are the institution's policy and procedure on reusing course numbers? How often are numbers reused? Why?
- Are section numbers or letters used? If so, is a meaning associated with them? Do they appear on transcripts?
- Are course or section numbers assigned to academic units in a centralized (e.g., Dean's list or Registrar's office, Curriculum Committee) or decentralized manner (e.g., each department assigns its own numbers)?
- Does the current system use a reference number for each course section? How is the reference number created? How is it used?

Developing the annual processing calendar for catalog and schedule information

The Academic Calendar record (acad_cal_rec) must reflect the course catalog and schedule dates. Therefore, all offices at an institution must understand and participate in developing the calendar used for course catalog and schedule information.

Access to student records, course catalog, schedule, registration, grading, and transcript information by other offices at an institution

To ensure that individuals outside a particular office are able to access student records, course catalog, schedule, registration, grading, and transcript information efficiently and cost-effectively, while guarding against misuse of this ability to access information, answer the following questions:

Student records

Which office is responsible for creating student ID cards? Does the card include information from admitted students' files? How is the information produced?

Catalog and Schedule

- Is an individual other than the registrar responsible for publishing the printed course catalog? How is the master course inventory maintained?
- What information does the bookstore need for ordering books? How does the bookstore get the information it needs?
- Which offices need facility schedule information (e.g., Information Office, Campus Security, Physical Plant)?

Registration

- Which offices need student schedule information (e.g., Financial Aid office, Dean of Students, Information office, Campus Security, Emergency contact office)?

- Is an office other than the Registrar's office (e.g., Institutional Research office) responsible for reports such as enrollment and IPEDS?
- Is an office other than the Registrar's office responsible for or involved in facility assignment or reporting (e.g., Academic Dean, individual academic departments, Physical Plant office)?

Grading

- Do offices other than the Registrar's office maintain copies of anonymous class lists turned in by faculty?
- Which offices receive a copy of student grade reports (e.g., Academic Dean, individual academic departments, Dean of Students, Placement office)? How are the reports organized?
- Which office is responsible for grade reports (e.g., Dean's office, Institutional Research office)?
- Which offices are responsible for determining the dean's list? How do these offices get the information they need?
- Which offices determine which individuals are on probation or suspension? How do these offices get the information they need?
- How do the Financial Aid office and the office responsible for merit scholarships get the information by which they assess satisfactory academic progress?
- Where and how is progress determined for veterans and social security students?
- How are requests for term grade information by third-party payees handled?

Transcript

- Which offices expect to receive an official or unofficial transcript on a regular basis? On a student-by-student basis? By some other grouping of students? How are such records used?
- Which offices expect to access online academic history information? For what purpose?
- Are transcripts provided to another campus office ever sent off campus? For what purpose?
- Who determines whether a degree is to be conferred with honors? On the basis of what information?

Course and section fees

Although the Registrar's and Bursar's offices usually coordinate course and section fees, sometimes fees or charges are handled manually by individual academic departments. CX supports virtually any kind of fee or charge associated with a course or section, and Jenzabar, Inc. recommends an institution-wide review of such charges.

All course or section requisites that govern enrollment

To implement the automated requisite checking feature of CX, complete these tasks:

- Establish requisites for each course that has requisites.
- Review the relationships among multiple requisites as they appear in the course catalog (e.g., review "and" and "or" logic).
- Identify non-course requisites (e.g., test scores and grades) that CX tracks.

All faculty or administration approvals required for students to enroll in courses and sections

Students occasionally must obtain approval from a faculty member, department, or dean in order to register for a course or section, if the requirement is course- or student-based. CX can help apply these requirements consistently. However, every office must completely understand the requirements so that the appropriate records can be created.

Decisions made during Course/Class Schedule implementations regarding student, Course, and Section records used in Registration

Review the decisions made and information entered in the Admissions and Course Class/Schedule applications regarding the following registration requisites to determine their effects elsewhere during registration:

- The academic status of registering students (e.g., admitted, continuing, or returning)
- The appropriate catalog and term for courses and sections

Responsibility (e.g., office or individual) for each of the registration and add/drop functions occurring in Jenzabar CX

The Registrar's office usually is responsible for registration and add/drop functions; however, some related registration processes often involve other offices. To improve the efficiency of the entire registration process and of the services provided to students, the offices involved in registration-related functions must do the following:

- Identify each registration-related function
- Identify who is responsible for each function
- Decide whether CX needs to be modified to accomplish each function

Chronological and physical dimensions of changes occurring with the Jenzabar CX registration process

Consider *when* and *where* registration is to occur. For example, if registration now occurs in an arena setting immediately before classes begin, the institution might consider one of these three alternatives:

- Spread the process over time, resulting in fewer disruptions to institutional operations and lower personnel costs
- Decentralized data entry to individual academic departments, in order to link academic advising and course enrollment more closely
- Implement the Interactive Voice Response (IVR) telephone registration of Jenzabar CX, enabling a student to register at a time and location of their own convenience

Staffing of the Jenzabar CX registration process

Staffing considerations are significant when registration is centralized and occurs over a short period of time, whether the institution uses regular staff, temporary staff, or students. During implementation, assess efficient and cost-effective approaches to staffing during registration.

Holds management

CX enables an institution to replace manual, staff-intensive procedures for tracking academic and administrative holds with automated holds capability. During implementation, all departments at the institution should identify the following information:

- All possible holds
- Who is authorized to place or remove each type of hold
- The intended effect of each type of hold
- Using CX, the institution can either decentralize or centralize hold functions (e.g., placing a hold, clearing a hold). In addition, the institution can label holds as either notation or absolute.

Identification, name, and address management

CX, because it is fully-integrated, is able to provide the following benefits to an institution regarding identification, name, and address management:

- Each individual or entity associated with the institution is assigned a single identification number
- The social security number, an optional feature in CX, calls up an individual's or entity's record just as the ID number does
- CX can track prior names, salutation names, and names that reflect relationships between individuals or entities, and affords virtually unlimited capability for different addresses
- In resolving this cross-functional issue, the offices at an institution must ensure that the following tasks are completed:

- Appropriate addresses are available when needed (e.g., billing, grade reporting, third-party)
- Offices are assigned responsibility for ensuring that identification, name, and address information is available when needed, and that all offices are working cooperatively

Formatting standards (e.g., address conventions)

For consistent, professional-looking correspondence, reports, and lists, the offices at an institution must agree on the following:

- How names and addresses are to be formatted
- What the valid values are for various data elements in the system (e.g., the names and abbreviations for degrees conferred, names of academic departments, majors)

Use of records

Detail records, or collections of related information that are used as supplemental data about an individual or an entity associated with an institution, use space in CX only if data exists in a record. Some of the detail records available in the Registration application require coordination by several offices so that the information is usable by all offices that have access to the information contained in them. The following detail records are examples:

- Relationship detail records identify the ways in which individuals or entities associated with an institution are related (e.g., father/son, sister/brother, husband/wife, employer/employee). Because different offices might view these relationships differently and might use the information in unique ways, it is important to meet the needs of all offices.
- The Accomplishment, Education, and Exam common detail records are critical to the Transcript application because the information contained in these records appears on academic records and transcripts, provided that certain key data are entered. The institution must decide whether these records are reserved exclusively for the registrar and, if not reserved exclusively for the registrar, what the rules are for use by other offices.

Admissions and academic status coding

A single status table (i.e., the Academic Status table) exists in CX to identify and track prospective students' admission statuses and enrolled students' academic statuses. However, there must be close coordination among the Admissions office, the Registrar's office, and the individual academic offices in order to identify the possible admission and academic statuses.

Determining completion of degree requirements and certifying degree completion

The offices involved in degree requirements or degree completion should organize the methods for determining when a student has met the requirements for a degree and for certifying that a degree has been earned and/or conferred. The offices must understand how these processes work, because CX can support several ways to organize these processes.

Note: The transcript forms (both official and unofficial) used in the Transcript application are very flexible regarding the information about degrees earned.

Transferring information from the Registrar's office to the Alumni Affairs office when a degree has been conferred (or at other times as appropriate)

CX electronically transfers information from the Registration application to the Alumni and Development application for individuals who have earned degrees. However, the Registrar's and Alumni and Development offices must agree on the following related issues:

- The definition of an alumnus
- The timing of the information transfer using CX
- The content of the information transfer using CX

Creating the Program Enrollment record from admissions information and accessing the Program Enrollment record for readmitted students

The Program Enrollment record is created when the Admissions office releases records of admitted students to the Registrar's office for registration; however, the Program Enrollment record must be created before a student can enroll in classes. The Admissions and Registrar's offices must resolve the following issues:

- When to create the records for both offices (because, once created, offices other than the Registrar's office are discouraged from accessing these records)
- Accessing the Program Enrollment records of former students who are being readmitted
- Updating the Program Enrollment record when a new Admissions record (adm_rec) is created, and the Program Enrollment record (prog_enr_rec) already exists

Assessing the Transcript Setup

Introduction

CX provides several ways to implement the options of the Transcript product. After assessing the needs of your institution, you can change the default settings of Transcript enable macros and reinstall the product.

This section lists and describes the features that you must assess before you can modify the macros.

Transcript Features

The Transcript module provides menu options accessed from the Registrar: Transcripts menu. You must determine whether to enable some of the menu options that can appear on this menu. They include:

- Course Work Evaluation

Note: This menu option is available if the macro ENABLE_FEAT_MULTI_SITE is set to Y.

- Enter Block Resident Work

Note: This menu option is available if the macro ENABLE_FEAT_BLK_RESIDENT is set to Y.

- Print Transcripts

Note: This menu option is available if the macro ENABLE_FEAT_LPS is set to Y.

Reviewing and Modifying Data in Tables and Records

Introduction

After assessing features of Transcript and setting the appropriate enable macros, you must review the setup of CX tables and records.

Procedure to Review Values

The following procedure provides the steps to review the values of CX tables and records.

1. For each Transcript table, review the codes supplied with CX. Determine whether the codes meet the needs of your institution. Make updates as appropriate.
2. Review the institution's records converted from the previous transcript system. Determine whether the records need to be updated to meet the needs of the CX reports. Make updates as appropriate.

Table and Record Information

For more information about the tables and records in the Transcript product, see the section *Transcript Tables and Records* in this manual.

Important Transcript Tables and Records

Although CX as delivered contains a set of table values that an institution can use to get started processing Transcript information, you must review the following tables in this order to ensure that the setup is correct.

1. Transcript Selection table (transsel_table)
2. Transcript Exam table (transexam_table)
3. Transcript Form table (transfrm_table)

Order of Table Information in This Section

Information about the setup of these tables appears in the following order in this manual:

- Tables that require modification appear first, in the order of implementation recommended by Jenzabar, Inc.
- Tables that do not require modification appear after the required tables, in alphabetical order

Table Setup Sequence

The following lists the sequence in which you should set up the Transcript tables. Information about setting up each of the tables below follows in this section. The Transcript tables appear in the Registrar: Table Maintenance menu.

Note: For reference information about the tables listed below, see the *Transcript Tables and Records* section in this manual.

1. Transcript Selection table (transsel_table)
2. Transcript Exam table (transexam_table)
3. Transcript Form table (transfrm_table)

Building the Transcript Tables

Introduction

The processes in the Transcript product use the Transcript tables to control data entry and to provide necessary information for financial aid processing. You must build these tables before performing any Transcript processing and before building any other Transcript tables.

Access

You access the Transcript tables from the Registrar: Table Maintenance menu.

Transcript Selection Table

The following lists the fields in the Transcript Selection table (transsel_table) in the order in which you complete them. To access this table, select Transcript Selection from the Table Maintenance: Registrar (T - Z) menu, then select **Finish**.

Transcript Selection Code

A code indicating the value passed to the Transcript program to identify the course categories to be included on the transcript (e.g., CC).

Description

A description of the code appearing in the Transcript Selection Code field (e.g., College Credit Program).

Course Category

A code indicating the course category to be included for this transcript selection (e.g., PREP Preparatory).

Transcript Exam Table

The following lists the fields in the Transcript Exam table (transexam_table) in the order in which you complete them. To access this table, select Transcript Selection from the Table Maintenance: Registrar (T - Z) menu.

Exam Type

A code identifying the type of exam to be included in the header of a transcript run with this transcript selection value (e.g., SAT).

Priority for Display

A number indicating the order in which this exam is to appear in the header of the transcript relative to other exam types for this same transcript selection (e.g., 4).

Transcript Form Table

The following lists the fields in the Transcript Form table (transfrm_table) in the order in which you complete them. To access this table, select Transcript Form from the Table Maintenance: Registrar (T - Z) menu, then select **Finish**.

Code

The code for this set of transcript form screens (e.g., JJJJ).

Official?

Either Y (Yes) or N (No) indicating whether this form is to be used on official transcripts.

Note: The default value is Y.

Lines Per Page

The number of lines on a single page of the transcript form (e.g., 66).

Note: The default value is 66.

Column

The maximum number of blank lines for a single column (e.g., 25).

Note: The default value is 25.

Number of Columns

The number of transcript columns this form uses (e.g., 1 or 2).

Note: When you specify "1" in this field, a one column unofficial transcript form (named MORE) is selected. When you specify "2" in this field, all other transcript forms can be used.

Width

The number of characters in a transcript column (e.g., 53).

Note: The default value is 53.

Spacing Between Columns

The number of spaces between transcript columns (e.g., 2).

Note: The default value is 2.

First Page Header

The name of the screen definition file used to print the header on the first page of the transcript (e.g., headW_0_1C).

First Page Footer

The name of the screen definition file used to print the footer on all pages of the transcript except the last page (e.g., footW_0_1).

Second Page Header

The name of the screen definition file used to print the header on all pages of the transcript after the first page (e.g., headW_0_2).

Last Page Footer

The name of the screen definition file used to print the footer on the last page of the transcript (e.g., footW_0_2).

Session Header

The name of the screen definition file used to print the header on all sessions except HEAD, PREV, and FOOT (e.g., sesshead).

Session Statistics

The name of the screen definition file used to print the statistics on all sessions except HEAD, PREV, and FOOT (e.g., sessstatsC).

Session Footer

The name of the screen definition file used to print the footer on all sessions except HEAD, PREV, and FOOT (e.g., sessfoot).

HEAD Session Header

The name of the screen definition file used to print the header on the Head session (e.g., HEADhead).

HEAD Session Statistics

The name of the screen definition file used to print the statistics on the Head session (e.g., HEADstats).

HEAD Session Footer

The name of the screen definition file used to print the footer on the Head session (e.g., sessfoot).

PREV Session Header

The name of the screen definition file used to print the header on the PREV session (e.g., PREVhead).

PREV Session Statistics

The name of the screen definition file used to print the statistics on the PREV session (e.g., PREVstats).

PREV Session Footer

The name of the screen definition file used to print the footer on the PREV session (e.g., sessfoot).

FOOT Session Header

The name of the screen definition file used to print the header on the Foot session (e.g., FOOThead).

FOOT Session Footer

The name of the screen definition file used to print the footer on the Foot session (e.g., sessfoot).

FOOT Session Statistics

The name of the screen definition file used to print the statistics on the Foot session (e.g., FOOTstats).

Column Footer

The name of the screen file (e.g., colfoot) used to print the last line of all but the last column (replaces session footer).

Course Line

The name of the screen file (e.g., crsline) used to print course information on the transcript where the course begin and end dates are 0 (zero).

Non-traditional Course

The name of the screen file (e.g., nontradcrs) used to print non-traditional course information on the transcript where the course begin or end date is not 0 (zero).

Course Title

The name of the screen file (e.g., crstitle) used to print additional course title lines (second, third, or section titles).

Course Flag

The name of the screen file (e.g., crsflag) used to print course flags (repeat, grading, or counting codes).

Transfer Work Header

The name of the screen definition file used to print the header on each group of transfer work (e.g., trfhead).

Block Transfer Header

The name of the screen definition file used to print the header for the block transfer work in any session (e.g., blkthead).

Block Transfer Line

The name of the screen definition file used to print the information for any Block Transfer record on the transcript (e.g., blkwline).

Block Transfer Footer

The name of the screen definition file used to print the footer for the block transfer work in any session (e.g., blktwfoot).

Education Header

The name of the screen definition file used to print the header for the Education records on the transcript (e.g., edhead).

Education Line

The name of the screen file (e.g., edline) used to print Education records at the beginning of the transcript (blank graduation session).

Education Session Line

The name of the screen file used to print Education records within transcript sessions having a valid graduation session code (e.g., edsess).

Exam Line

The name of the screen definition file used to print the information for any Exam records on the transcript (e.g., examline).

Exam Scores

The name of the screen definition file used to print the Exam table for any exam if the first label is not blank (e.g., examscore).

Accomplishment Line

The name of the screen definition file used to print the information for any Accomplishment record on the transcript (e.g., accompline).

Accomplishment Description

The name of the screen definition file used to print the description for any accomplishment, if it is not blank (e.g., accompdesc).

Signature Line

The name of the screen file used to print the signature line for the end of the transcript (e.g., signline).

Ending Statistics Line

The name of the screen file used to print the ending statistics line for the transcript (e.g., endstats).

The following lists the fields in the User Comments portion of the Transcript Form table in the order in which you complete them.

User

The name of the screen file used to print the transcript user comment line (e.g., comment).

The following lists the fields in the Grade Audit Comments portion of the Transcript Form table in the order in which you complete them.

Counting Change

The name of the screen file used to print the transcript counting change comment line (e.g., grdaucomm).

Grade Change

The name of the screen file used to print the transcript grade change comment line (e.g., grdaucomm).

Grading Change

The name of the screen file used to print the transcript grading change comment line (e.g., grdaucomm).

Hours Change

The name of the screen file used to print the transcript hours change comment line (e.g., grdaucomm).

Setting Up the Transcript Program (*trans*)

Introduction

Various compilation options for the Transcript program (*trans*) must be established prior to the initial compilation of the Transcript program for each client institution. The file that requires attention for customizing the output of transcripts is `$CARSPATH/include/custom/trans`. The compilation options that may vary between client institutions are defined in this file. Once this file has been set up correctly, the options will be included with any compilations of the program.

Setting Includes for Transcript Processing

The following lists and describes the includes that your institution can customize. The includes appear in the order in which they appear in the `$CARSPATH/include/custom/trans` file.

NOWHITE

In order to make the transcript as orderly as possible, information will be grouped together by session and by the information within a session. As a result, there will be some blank space at the bottom of the columns of the transcript where, to avoid breaking a group of data, the group was moved to the top of the next column. Defining the **NOWHITE** option indicates that the white space at the bottom of the columns on a transcript should have dots printed instead of being left blank. The objective is to make it obvious if someone attempts to add entries to their transcript. If you do not want to print these dots, do not define this variable.

GPA_PRECISION,

POINTS_PRECISION,

HOURS_PRECISION

GPA_PRECISION defines the number of digits to the right of the decimal to be stored for GPA values. **POINTS_PRECISION** defines the same for quality points values.

HOURS_PRECISION defines the same for hours values. Use "3" for "3.333" format, use "2" for "3.33" format, and so on.

If you make changes to these decimal positions, ensure that the appropriate forms in `$CARSPATH/src/regist/trans/SCR` allow for enough spaces to print the changes that were made to the includes and that the affected attributes are correct. Also, the grade change mailer in `$CARSPATH/modules/regist/grdrpt/final` may need similar types of changes.

Although these are the most important places where changes are required, there may be others, especially in reports. Be aware that points and hours generally refer to any fields that store points or credit hours. The more common records involved are `cw_rec`, `stu_acad_rec`, and `stu_stat_rec`.

GPA_TRUNCATE,

POINTS_TRUNCATE,

HOURS_TRUNCATE

GPA_TRUNCATE specifies to truncate GPA values to GPA_PRECISION decimal places instead of rounding. POINTS_TRUNCATE specifies the same for quality points values, and HOURS_TRUNCATE specifies the same for hours values.

If undefined as preceding, the GPA, points, and hours are rounded to the number of decimal places specified in the preceding _PRECISION includes. If any of the includes are defined, those values will be truncated to the decimal places specified in the preceding _PRECISION includes.

Example: If the GPA computes to 1.9999 and three decimal precision is defined, GPA_TRUNCATE will print a GPA of 1.999 while rounding would print 2.000.

RESREP

If students are permitted to repeat only resident or transfer course work, you should define the RESREP option so that CX applies repeat logic only to resident courses. If this option is not defined, CX calculates repeats (i.e., applies repeat logic) on all courses (both resident and transfer course work). When checking for repeats, CX matches course numbers for resident course work and matches equivalent course numbers for noncatalog and transfer course work.

REP_BY_GRD

When this option is defined, the repeated course with the highest grade value will be counted as the replacement course. If this option is not defined, the grade for the last time the course was attempted will be counted instead of the highest grade for previous attempts regardless of the grade received.

Example: A student takes MAT101 in FA99, gets a “D,” retakes the course to improve the grade in SP00, but gets an “F” for the SP00 attempt. If this include is defined, the repeat logic would retain the “D” and remove the “F” from grade point averages. If the include is undefined, the “F” would count in the GPA, and the “D” would be removed from the averages. The adjustments are made in the session in which the course was retaken; the original term remains unchanged.

Note: The repeat logic applied is very complex. Very important considerations are the values set in the crs_rec for repeatable (i.e., Y or N) and the number of times repeatable (e.g., 1).

CUM_BY_LAST

If any cases of repeating are dependent upon the number of hours the student has earned (or classification), you can use this option. If CUM_BY_LAST is defined, the earned hours at the time of the repeat attempt will be used to determine the type of repeating to be done. If it is not defined, CX uses the earned hours at the time of the original course.

Example: If a student may repeat a course only while a freshman, the CUM_BY_LAST option will be defined, meaning the hours the student has earned when the course was taken the second time will be less than the number of hours that makes the student a sophomore. If a student may repeat courses taken only while a freshman, the option would not be defined, meaning the hours the student had earned when the course was taken the first time will be less than the number of hours that makes the student a sophomore.

MAXREP

This option defines the maximum number of repeats a student is allowed. If defined and a number other than zero is specified, this represents the total number of times repeat logic will be applied to improve grades for a given program. This represents total attempts, not by a specific course. If there are limits on the number of times a course may be retaken to improve a grade, this is set on the `crs_rec` (e.g., repeatable "N" and number of times repeatable equals some nonzero value). If `MAXREP` equals 0, there is no limit to the number of times the repeat logic can be invoked to improve a grade.

Example: If set to 4, the student can use the repeat logic to improve grades only 4 times. Any attempts exceeding 4 times will cause them to be computed in the grade point average, but no additional hours will be earned.

ONE_6TH_HRS

Use the `ONE_6TH_HRS` option only if the institution has one-sixth credit hour courses as the smallest unit. Because of rounding in calculations, you must consider that the value one-sixth is a repeating decimal. If this option is not defined, all hours values will be printed with three decimal places instead of two, and rounding will be corrected to allow six courses of one-sixth credit hour each to total to one credit hour.

USE_CLOCK_ONLY

The `USE_CLOCK_ONLY` option will cause the Transcript program to use the Clock Hours (`cw_clock_hrs`) value in place of the Course Work Hours (`cw_hrs`) value from the Course Work record (`cw_rec`). This provides a way for registration and billing procedures to use one value for a course and another value for the Transcript program. This replacement is done for all printing and calculations, so the number of hours printed for a course will be the clock hours, not the course work hours. Also, the attempted, earned, pass, and quality hours are all calculated from the clock hours.

Example: An example of this use would be courses that extend across sessions. Suppose a six hour course was offered beginning in the Fall session and ending in the Spring session. Registration reports and student billing should reflect that the student is taking a three hour course in the Fall and a three hour course in the Spring. Registration would enroll the student for one course in each session. The transcript, however, should show only the six total earned hours in the Spring with no earned hours in the Fall. This would not be accomplished by assigning three hours to the `cw_hrs` value in each session, but by assigning zero hours to the `cw_clock_hrs` value in the Fall and six hours to the `cw_clock_hrs` value in the Spring. By setting up the Course records (`crs_rec`) and Section records (`sec_rec`) correctly, the values for the `cw_hrs` and `cw_clock_hrs` would default to the appropriate values with no extra intervention by the operator during registration.

HIGH_SCHOOL

This include affects the printing of a student's previous education on transcripts. It defines the high school code value used in the `sch_type` field. If defined, along with the `HIGH_SCHOOL_ONLY` include, high school Education records (`ed_rec`) can be printed on the transcript.

Example: By redefining the `HIGH_SCHOOL` include to a name matching the institution, for example "UNIV", and by defining the `HIGH_SCHOOL_ONLY` include, the degrees awarded by UNIV can be printed in the body of the transcript only if the `-e` parameter is passed in the menuopt for transcripts (or from the shell if `trans` is called by that method).

Note: Education records (`ed_rec`) are included on a transcript only if the `-e` option is passed to the Transcript program.

HIGH_SCHOOL_ONLY

This include affects the printing of a student's previous education on transcripts. If the HIGH_SCHOOL_ONLY include is defined, only those Education records that refer to a School record (sch_rec) with a Type Code (sch_type) matching the HIGH_SCHOOL value are printed. If the HIGH_SCHOOL_ONLY option is not defined, any Education record will print on the transcript.

TR_SLV_CMD

The TR_SLV_CMD value determines how the slave printer is to be initialized when printing to a slave device. The standard options for condensed or regular print size include "condense", "elite", and "pica". The choice depends on the particular capabilities of the slave printers in use by the operator running *trans*.

BLKTW_SESS_SKIP "QZ"

Identifies a block transfer session in *trans* where the statistics block will be skipped and will not print during output. Set this include to an invalid session to disable it.

Example: BLKTW_SESS_SKIP "QZ" means the "QZ" block transfer session statistics will be skipped, if they exist.

FOOT_ACCOMP_LAST

Puts Foot session accomplishments at the end of the Foot session. If this include is defined, and if there are accomp_rec entries for a Foot session and a specified program (e.g., UNDG), these accomplishments can be printed as the last session of a transcript. Otherwise, do not define this include.

SECTION_TITLE_ONLY

Replaces course titles with section titles on printed transcripts. If not defined, both the course title and the section title will print on the transcript. If defined, the section title will print only if there is one; otherwise, the course title will print. In grade reports, the section titles always will print if there are any; otherwise, the course titles will print.

Note: This include does not affect the statement about grade reports.

COUNT_EDIT_HRS

The COUNT_EDIT_HRS option will cause courses added by the Transcript program to be included in the Student Academic Registered Hours (stuac_reg_hrs) value. If this option is commented out, only the courses added through the registration process will be included in this value.

This include also includes Transcript Edit courses (cw_rec.stat=E) into stu_acad_rec.reg_hrs. If resident courses are input through Transcript Edit, a cw_rec.stat=E is assigned. If the include is defined, these hours will be included in the stu_acad_rec.reg_hrs. If not defined, they will not count in registered hours.

BEG_SORT_SESS

BEG_SORT_YR

These includes define for all transcripts the first session where courses will be sorted by course number within each school ID. All sessions prior to this session will list the courses in the order they were entered. To have all sessions sorted, use a blank session ("") and zero (0) year. To have all sessions unsorted, comment out both definitions. If a session and year are specified in the two includes, *trans* will start sorting the courses beginning with that session and year. All other sessions will be in unsorted order, meaning the order in which they were entered in registration. If both includes are undefined, all sessions always will be in unsorted order, or the order in which the courses were entered.

Example: If defined as a blank session ("") and zero (0) year, all courses for all sessions will be sorted in course number order (e.g., ACC101, ENG202, MAT304, etc.) on the transcript.

RESET_GPA_SESS

RESET_GPA_YR

These includes allow the transcript to begin new grade point average calculations. The definitions contain the database field names from the Program Enrollment record that define for a student the ending session for the grade point average calculations on the first part of the transcript. This will cause the cumulative quality points and hours to be reset to zero before the next session is calculated on the transcript. Also, this will cause the repeat calculations to begin anew to prevent courses on the second part of the transcript to repeat courses on the first part of the transcript. These field values may correspond to the granting of a degree. If the session value is blank, or if the year value is zero, the grade point average will not be reset on that student's transcript. If the definitions are commented out, the grade point average will not be reset on any transcript.

Note: It is a common practice for a registrar to freeze the student's Academic record when they receive a terminal degree for a given program (e.g., bachelors degree for UNDG, masters degree for GRAD). No changes are to be made to the student's record that affect grades; however, it is a common practice for the student to continue taking course work in a program after earning the terminal degree either to earn another degree or just to take credit without any intentions of a degree. In many cases, the registrar will want to restart the grade point average computations the first session after the terminal degree was awarded. If so, these two includes would be defined. If the field names as shown are used, they will have to be added to the schema (\$CARSPATH/student/progenr) and added to the appropriate screens for operator input of data (\$CARSPATH/modules/regist/stuentry). The Transcript program will restart the quality hours and quality points the first session and year after the data specified in these two fields. Only quality hours and quality points restart; all other session and cumulative statistics continue in cumulative form, since only quality hours and quality points affect the grade point average. If a student repeats a course after the session and year are reset, the program will not adjust the repeat logic unless the course first was taken after the session and year specified in these two fields. If undefined, grade point averages will not be restarted and repeat logic will be applied as if it were only one continual record. An alternative to this process is to register a student in a different program if they planned to take additional course work at the same program level.

Example: A student earned a bachelors degree and wanted to take additional UNDG course work, and a new program (typically POST for post-bachelors) was introduced. This restarted everything, but it also would result in two transcripts. This is frequently the primary disadvantage if the registrar wants all course work for a given program on one transcript. CX produces only one transcript per program. There is no capability to produce one transcript for all course work for all programs.

Note: You may still use the preceding method, but a newer method using the `stu_acad_rec.restart_gpa` field is recommended. A "Y" in this field denotes a grade point average restart beginning with that session.

LABEL_FIELDS

LABEL_FIELDS defines screen fields that appear based on the data in another field. If the value of the data field is non-blank, the screen field contains either its default value or its data value. Otherwise, the screen field is blank. The first element is the screen field name from the screen definition file. The second element is the database field name of the value that must be non-blank for the screen field to be filled in. The third element is the keyword "DEFAULT" for the default value or "DATA" for the data value. The definition must appear in the following way:

```
{"screen_field", "database_field", "DEFAULT"}, \
      or
{"screen_field", "database_field", "DATA"}, \
```

The braces ({}), double quotes ("), and the backslash (\) are needed. Note the comma outside the braces on all but the last line, as in the following example:

Note: Be sure to update LABEL_FIELDS_NO after changing LABEL_FIELDS.

Example:

```
{"degdate", "prog_enr_rec.prog_deg_date", "DATA"}, \
 {"major1label", "prog_enr_rec.major1", "DEFAULT"}, \
 {"major2label", "prog_enr_rec.major2", "DEFAULT"}, \
 {"minor1label", "prog_enr_rec.minor1", "DEFAULT"}, \
 {"minor2label", "prog_enr_rec.minor2", "DEFAULT"} \
```

Note: The screen_field and the database_field must match the definitions in the attribute section of the program screen/form.

Typically the transcript is set up for particular fields to print. This is especially true of the transcript header but could apply to any fields on the transcript (e.g., ed_rec). However, many of these fields print the label even though there is no data in the field. For example, the label "Degree Date..." would be printed with all zeroes if no degree date had been assigned. The registrar may wish to suppress these fields unless there is data in them. This include provides the capability to selectively accomplish this requirement.

Note: It is beyond the explanation here to define all the requirements necessary to accomplish this action.

Attributes have to be added to the screen/form that suppresses the label in addition to defining the include. The attribute would be put in brackets just as any other field that is desired on the form definition and then defined in the attributes.

Example: Label "First minor" is printed only if there is actual data in the "minor1" field of the prog_enr_rec. On the form definition enter [minor1label] and in attributes enter "minor1label: optional, default="First Minor.";". Then, if an include is defined like the preceding one, the label "First Minor." prints only if there is a minor in the "minor1" field. Using LABEL_FIELDS to suppress whole lines of data may result in significant blank space on the transcripts. When the labels are suppressed, there is no compression of the space. Instead, the space is retained in the same number of lines that were defined for the form. This blank space may cause the registrar some difficulty because it opens the possibility of tampering with the transcripts by someone adding non-factual information in the blank space.

LABEL_FIELDS_NO

Defines the number of lines defined in LABEL_FIELDS.

TRANS_ADD_REASON

TRANS_VOID_REASON

Defines the reason codes put in the reg_recs added when a cw_rec is added or voided in Transcript Edit.

PAGE

The length of a transcript page is determined by the PAGE value. This number should be set to the length of the paper or form being used to print the transcripts.

MAXBLANK

As discussed with the NOWHITE option, there will be cases where there is some blank space at the bottom of a transcript column. The Transcript program will first attempt to put an entire session on a column without breaking it. If the session is very long, this could cause a great amount of blank space at the bottom of a column. This blank space can be controlled using the MAXBLANK value. The value determines the maximum number of blank lines that may be left at the bottom of a column. For example, if the columns of the transcript were 55 lines long and the MAXBLANK value was set to 20, at least 35 lines of the column would be used before the next column could be used. The correct value for this variable depends on the length of the page as well as the length of the headers and footers. As a general rule, the value should be set to about the size of the average session. If it is too small, groups of data will be broken unnecessarily; if it is too large, too much of a column may be left blank.

MAN_REP

Normally, only the Transcript program maintains repeat codes. There may be cases, however, where the repeat logic used by *trans* does not perform as desired. The MAN_REP option allows definition of repeat codes that can be changed only manually. This will allow the operator to flag a particular course as repeated that otherwise would not be calculated correctly.

VINC

The VINC option will cause the Transcript program to expect a Certificate table (*cert_table*) in the database and allow its values to be printed in the headers and footers of the transcript. This is not a standard data table in CX.

SHORT_STATS

The SHORT_STATS option will cause transcripts to be printed with only the attempted and earned hours and the GPA values. This option suppresses the printing of pass and quality hours and quality points.

TWO_DEC_GPA

The TWO_DEC_GPA option will force transcripts to be printed with GPAs of two decimal places. This option overrides the default printing of GPAs with three decimal places.

CLASS_RANK

The CLASS_RANK option will cause the Transcript program to print the class rank and size values from the Academic record (*stu_acad_rec*). These values are printed after the statistics block in each session if the class size value is not zero.

NO_SIGNATURE

The NO_SIGNATURE option will suppress the printing of the signature lines at the end of the last session on the transcript. If this option is commented out, four blank lines will follow the last session, a line of underscores for the signature followed by the date the transcript was printed, followed by the title "Office of the Registrar". If the option is used, the transcript will be six lines shorter than normal.

DTL_SESS_HEAD

The DTL_SESS_HEAD option changes the format of session headers to include the beginning and ending dates of the session from the Academic Calendar record (*acad_cal_rec*) and the classification code from the Academic record (*stu_acad_rec*). If the *acad_cal_rec* cannot be found, the beginning and ending dates are omitted. The "HEAD", "FOOT", and "PREV" sessions are not affected by this option.

CLOCK_HRS

The CLOCK_HRS option changes the format of the course lines to include the Clock Hours (cw_clock_hrs) value from the Course Work record (cw_rec). This value is sometimes referred to as "contact" hours. The session and cumulative totals of the earned clock hours also will follow each session statistics block. The CLOCK_HRS option *may not* be used if the ONE_6TH_HOURS option is used.

USE_TW_BLOCK

The USE_TW_BLOCK option will cause the Transcript program to print Block Transfer records (blktw_rec) . The standard use of this option is for institutions converting from the CX-GA System. The Block Transfer record contains a summary of attempted, earned, pass and quality hours, quality points and GPA from all courses to be transferred from another institution.

GA_FORMAT

The GA_FORMAT option will change the formatting of the transcript to more closely resemble the format of transcripts under the CX-GA System. The differences include: no end of column dashed line, session headers are centered with only eight dashes on either side of the data, and Head and Foot session headers are printed with equals signs (=) rather than dashes (-).

USE_MAJOR3 and USE_MINOR3

If the USE_MAJOR3 option is included, the Transcript program will allow the text from the Major table (major_table) to be printed on the transcript headers and footers from the "major3" field in the Program Enrollment record (prog_enr_rec). Similarly, the USE_MINOR3 option allows the text from the Minor Table (minor_table) to be printed from the "minor3" field in the Program Enrollment record. These fields are not standard in CX.

Variations in the Transcript Program

Three variations of the Transcript program (*trans*) are available. The first is a display-only option. The second is used to print transcripts using Contact records. The third is used to update statistics. You can run all three as background tasks from the menu.

Display Transcripts

The display-only version of the Transcript program will permit the operator to print only a student's unofficial transcript to the screen. This option is executed as **trans -d** where "-d" is a parameter used to display the transcript. You can place it on the financial aid menu to allow the Financial Aid office to view a transcript without being permitted to edit or update a student's transcript or print any official transcripts.

Print Official Transcripts

This variation requires three parameters, one to identify it as printing transcripts from the Contact record, one for official, and one for the student's program. For example, **trans -c -o -p UNDG -C** will create a file of official transcripts for undergraduate students where each transcript will include course work from the current session. The operator will need only to enter the program code when prompted through the menu.

Print Unofficial Transcripts

This variation requires two parameters, one to identify it as printing transcripts from the Contact record, and one for the student's program. For example, **trans -c -p UNDG** will create a file of unofficial transcripts for all students with the appropriate Contact records. The "-C" parameter also may be added to the menu option. The operator will need only to enter the program code when prompted through the menu.

Update Student Statistics

You can update student statistics in the `stu_acad_rec` and `stu_stat_rec` using parameters for different circumstances. To update all student statistics, the “-u” parameter is passed to `trans` as in **trans -u**.

Examples:

- **trans -u -y 2000 -s FA** will update session student statistics for Fall 2000.
- **trans -u -p UNDG** will update all student statistics for students in the undergraduate program.
- **trans -u -I 201** will update all student statistics for ID number 201.
- **trans -u -Y 0001** will update the academic year-to-date statistics for students for the year 2000-2001. The Academic Calendar record (`acad_cal_rec`) must have the `ac_acyr` field updated with the correct academic year values for each session in the record.

Modifying Transcripts

Introduction

This subsection describes the steps to modify the format of a transcript. It describes the procedures for:

- Adapting the transcript to institution policy
- Creating and modifying transcript formats
- Locating student statistics information
- Listing student statistics database field names
- Locating other files containing transcript information
- Modifying the Transcript program screen
- Adding a new transcript form
- Updating the macro file and the menu options
- Checking transcript format modifications
- Changing the default form
- Changing the system default form
- Testing new forms
- Resolving transcript problems
- Adding page numbers to the transcript
- Preventing data from printing
- Using the default attribute
- Printing blank lines
- Printing inconspicuous characters
- Including course work in progress

How to Adapt the Transcript to Institution Policy - Reviewing the Options

Begin by reviewing the options that are available for the *trans* customization in the file `$CARSPATH/include/custom/trans`. This is a file that is used by *trans* and is incorporated into the program when it is compiled. See *Setting Includes for Transcript Processing* in this document, for further explanation. To disable an option, enclose it within the C-code comment symbols, `/* -----` and `----- */`.

An example of commenting in and commenting out a function follows. If the unused portion of the transcript is to be covered with dots, the option that tells *trans* to print the dots would look like this:

```
/* -----  
Print dots on all white area  
----- */  
#define NOWHITE
```

Note: Do not remove or alter the line(s) beginning with the `#` sign. The compiler looks for these lines when the program is compiled.

To tell *trans* not to print the security dots requires that the `NOWHITE` option be commented out in the following fashion:

```
/* -----  
Print dots on all white area  
#define NOWHITE  
----- */
```

The `NOWHITE` statement is now "commented out" and will not be used by *trans*.

Any other option, used or not used, must be either commented in or commented out. A standard set of options is commented in. A brief description accompanies each of the options. Read each option carefully to determine whether the option should be used at your institution.

How to Adapt the Transcript to Institution Policy - Changing the File

The following lists the steps for making any changes in the file. The file must be checked out, changed, checked in, and installed in order for it to be available for the Transcript program.

1. Enter **cd include/custom** at the shell prompt to change to the directory containing includes.
2. Enter **make co F=trans** to check out the include file for transcripts and lock the file to prepare to update it.
3. Enter **vi trans** to access the file in Edit mode.
4. Use the vi editor to move the comment lines as necessary to change the includes as needed for your institution.
5. Enter **:wq!** to save your changes to the include file.
6. Enter **make cii F=trans L="Adapted transcript program for our college."** to unlock the file and document your reasons for making the changes.

The file is now ready to be used to produce customized transcripts. When all of the options have been set, you must recompile *trans* to bring in the changes. The following lists the steps to recompile *trans*:

1. Enter **cd src/regist/trans** at the shell prompt to change to the source directory.
2. Enter **make cleanup** to recompile *trans*, erasing object files.
3. Enter **make REINSTALL>& make.out &** to recompile *trans* and the files *trans* uses.

The previous commands use the MAKE processor to recompile *trans*. An explanation of these commands follows:

- The "make cleanup" erases all of the object files and forces all parts of *trans* to recompile.
- The "make REINSTALL" recompiles all of the parts of *trans*, as well as all of the files (in the subdirectory SCR) that *trans* uses.
- The trailing "&" causes the process to run in the background.
- The output from the REINSTALL is stored in the file "make.out" in the directory \$CARSPATH/src/regist/trans.

When the process is finished, review the file to ensure that there are no errors by using the following command at the shell: **more make.out**

Transcript Formats - Defined

Several transcript formats have been defined. If there is student course work data available on the system, it is possible to print a transcript using one of the transcript production options from the menu.

Transcript Format - Example

The Transcript program uses application program screens (SCR) to define the output format of transcripts. These screens determine the database fields that print, as well as the textual information. The Transcript Form table (transfrm_table) defines form codes and links them to the names of screen definition files.

When you print a transcript, you choose a form code, and *trans* uses the set of screens defined by that form code to output the data that must be printed. The transcript format consists of 34 different types of lines, and each type uses a separate screen definition file. A transcript format diagram using all these screens follows.

```

1=====
1                                     FIRST PAGE HEADER
1=====
2          Previous Education          14----- Fall 199X -----
2                                     15 PHIL499  Philosophy of the 3.00 C
3 Princeton H.S. Cincinnati, OH 45242 16 Ancient Greeks and
4----- HEAD session ----- 16 Their lifestyle
5----- Transfer Credits ----- 16 Men's Section
6 University of Cincinnati          17 ** Letter grading **
6          3.00 3.00 0.00 3.00 6.00 2.000 17 ** Earned for credit **
7----- 17 ** Repeated later **
8                                     18 10/01/9X SAT Test completed
8          Att Earn Pass Qual  Pts  GPA 19 English Math Total
8 ses 3.00 3.00 0.00 3.00 6.00 2.000 19 320 370 690
8 cum 3.00 3.00 0.00 3.00 6.00 2.000 20 10/02/9X Forensics Award
9----- 21 Second place in Region
10----- Previous Transcript ----- 22
11                                     22 Att Earn Pass Qual  Pts  GPA
11          Att Earn Pass Qual  Pts  GPA 22 ses 3.00 3.00 0.00 3.00 6.00 2.000
11 ses 3.00 3.00 0.00 3.00 6.00 2.000 22 cum 9.00 9.00 0.00 9.00 18.00 2.000
11 cum 6.00 6.00 0.00 6.00 12.00 2.000 13(23)-----
13(12)-----
24-----
24                                     FIRST PAGE FOOTER
24-----
25-----
25                                     SECOND PAGE HEADER
25-----
26----- FOOT Session ----- 26----- FOOT Session (Cont.) -----
27 Comment line...          31 CARS College
28                                     31 Degree: Bachelor of Arts
28 Transfer work from Ohio State U. 31 Awarded: 06/01/9X
29 SOC200 Amer. Sociology 3.00 TR 31 Major: Philosophy / Sociology
29          10/08/9X - 12/29/9X 33(32)-----
30                                     33
30          Att Earn Pass Qual  Pts  GPA 33
30 ses 3.00 0.00 3.00 0.00 0.00 0.000 33
30 cum 12.00 9.00 3.00 9.00 18.00 2.000 33
13----- To be continued ----- 33 Office of the Registrar 01/01/9X
. . . . . 33
34-----
34                                     LAST PAGE FOOTER
34-----

```

Transcript Format - Line Types

The numbers at the beginning of the left margin and the beginning of the second column of transcript label each type of line. The following lists and describes the line types in a transcript.

1 - First page header

The associated field name on the Transcript Form Table is page1_head.

This line type prints at the very top of the first page of the transcript. It is not used on any other page.

The data values for the screen definition file are as follows: Any field from the id_rec, profile_rec, and prog_enr_rec may be used. The "adname" character field contains the student name from ADR. The "adrline0" character field contains the first address line from ADR. The "adrline1" character field contains the second address line from ADR. The "adrline2" character field contains the third address line from ADR. The "adrline3" character field contains the fourth address line from ADR. The "adrline4" character field contains the fifth address line from ADR. The "dtfld" date field contains the system date. The "fmtfield" character field may be used with a format attribute.

2 - Education record header

The associated field name on the Transcript Form Table is page1_head.

If any Education records (ed_rec) are to be printed at the beginning of this student's transcript, this header is printed once before all of the Education records. It is not used any other place on the transcript.

The data values for the screen definition file are as follows: The "dtfld" date field contains the system date. The "fmtfield" character field may be used with a format attribute.

3 - Education record

The associated field name on the Transcript Form Table is `ed_line`.

This line type prints for each Education record at the top of the transcript. Education records appear on a transcript only if the "-e" parameter is passed on the calling line to *trans*, either from the menu or from the shell. Otherwise they are ignored.

If the `ed_rec` has a blank Graduation Session (`ed_rec.grad_sess`), it prints at the beginning of the transcript. Any record with a valid Graduation Session uses the Education Session record (see line type 31).

If the "HIGH_SCHOOL_ONLY" option is defined in the file `$CARSPATH/include/custom/trans`, only those `ed_rec`s that link to a school whose School Type (`ctgry`) matches the "HIGH_SCHOOL" compile definition print at the beginning of the transcript.

The data values for the screen definition file are as follows: Any field from the `ed_rec`, `id_rec`, and `sch_rec` may be used, where the `id_rec` and `sch_rec` contain the school information from the `ed_rec.sch_id`. The "dtfld" date field contains the system date. The "fmtfield" character field may be used with a format attribute.

4 - Head session header

The associated field name on the Transcript Form Table is `head_head`.

This line type prints as the header for the Head session, if one is to be printed on this student's transcript. A Head session is printed if any transcript information has been added for that session. It is not used any other place on the transcript. The screen definition may not exceed one line.

The data values for the screen definition file are as follows: Any field from the `acad_cal_rec`, `stu_acad_rec`, and `stu_stat_rec` may be used. The "sesscont" character field contains its default value if the session is continued from the previous transcript column; otherwise the value is blank. The "dtfld" date field contains the system date. The "fmtfield" character field may be used with a format attribute.

5 - Block transfer header

The associated field name on the Transcript Form Table is `blktw_head`.

If any Block Transfer records (`blktw_rec`) are to be printed on this student's transcript, this header is printed once before all of the Block Transfer records in each session.

The data values for the screen definition file are as follows: See line type number 2, "ed_head."

6 - Block transfer record

The associated field name on the Transcript Form Table is `blktw_line`.

This line type prints for each Block Transfer record in any session.

The data values for the screen definition file are as follows: Any field from the `blktw_rec` may be used. The "dtfld" date field contains the system date. The "fmtfield" character field may be used with a format attribute.

7 - Block transfer footer

The associated field name on the Transcript Form Table is `blktw_foot`.

If any Block Transfer records (`blktw_rec`) are to print on this student's transcript, this footer prints once after all of the Block Transfer records in each session.

The data values for the screen definition file are as follows: See line type number 2, "ed_head."

8 - Head session statistics

The associated field name on the Transcript Form Table is `head_stats`.

This line type prints as the statistics block for the Head session, if one is to be printed on this student's transcript. A Head session prints if any transcript information has been added for

that session. The statistics come after all courses, exams, and accomplishments in the session. It is not used any other place on the transcript.

The data values for the screen definition file are as follows: Any field from the acad_cal_rec, stu_acad_rec, and stu_stat_rec may be used. Fields contain the following:

- tatt - transfer attempted hours for the session
- tearn - transfer earned hours for the session
- tpass - transfer passed hours for the session
- tqual - transfer quality hours for the session
- tpts - transfer quality points for the session
- tgpu - transfer GPA for the session
- satt - total attempted hours for the session
- searn - total earned hours for the session
- spass - total passed hours for the session
- squal - total quality hours for the session
- spts - total quality points for the session
- sgpa - GPA for the session
- sclk - total clock hours for the session
- cclk - total cumulative clock hours
- dtfld - system date
- fmtfield - may be used with a format attribute

9 - Head session trailer

The associated field name on the Transcript Form Table is head_foot.

This line type prints as the last line of the Head session, if one is to print on this student's transcript. A Head session prints if any transcript information has been added for that session. It is not used any other place on the transcript. This is replaced by the "Column trailer" (see line type 13) if it is the last item printed in the column. The screen definition may not exceed one line.

The data values for the screen definition file are as follows: Any field from the acad_cal_rec, stu_acad_rec, and stu_stat_rec may be used. The "sesscont" character field contains its default value if the session is continued on the next transcript column; otherwise, the value is blank. The "dtfld" date field contains the system date. The "fmtfield" character field may be used with a format attribute.

10 - Prev session header

The associated field name on the Transcript Form Table is prev_head.

This line type prints as the header for the previous session, if one is to be printed on this student's transcript. A previous session prints if any transcript information has been added for that session. It is not used any other place on the transcript. The screen definition may not exceed one line.

The data values for the screen definition file are as follows: See line type number 4, "head_head."

11 - Prev session statistics

The associated field name on the Transcript Form Table is prev_stats.

This line type prints as the statistics block for the previous session, if one is to print on this student's transcript. A previous session prints if any transcript information has been added for that session. The statistics come after all courses, exams, and accomplishments in the session. It is not used any other place on the transcript.

The data values for the screen definition file are as follows: See line type number 8, "head_stats."

12 - Prev session trailer

The associated field name on the Transcript Form Table is prev_foot.

This line type prints as the last line of the previous session, if one is to print on this student's transcript. A Previous session prints if any transcript information has been added for that

session. It is not used any other place on the transcript. This is replaced by the "Column trailer" (see line type 13) if it is the last item printed in the column, as in the preceding example of the transcript format. The screen definition may not exceed one line.

The data values for the screen definition file are as follows: See line type number 9, "head_foot."

13 - Column trailer

The associated field name on the Transcript Form Table is col_foot.

This line type prints as the last line of all columns of the transcript except the last. It either replaces the session trailer or is added to a broken session. A broken session refers to a session where some of the information is printed in one column and continues on the next column. The screen definition may not exceed one line.

The data values for the screen definition file are as follows: The "sesscont" character field contains its default value if the session continues on the next transcript column; otherwise, the value is blank. The "dtfld" date field contains the system date. The "fmtfield" character field may be used with a format attribute.

14 - Session header

The associated field name on the Transcript Form Table is sess_head.

This line type prints as the header for any session except Head, Previous, and Foot. A session prints if any transcript information that has been added for that session and year combination. The screen definition may not exceed one line.

The data values for the screen definition file are as follows: See line type number 4, "head_head."

15 - Course line

The associated field name on the Transcript Form Table is crs_line.

This line type prints for each traditional course in any session. Traditional refers to a course where both the begin and end dates in the Course Work record (cw_rec) are equal to zero (00/00/0000). See line type 29 for nontraditional courses.

The data values for the screen definition file are as follows: Any field from the crs_rec and cw_rec may be used. The cw_rec data is modified as follows:

crs_no - contain the data from the ncat_crs_no if the course is noncatalog or transfer

hrs and clock_hrs - divided by the unit_fctr from the Program table

Fields contain the following:

- grd - midsess_grd, if the option to print mid-session grades was selected
- att - attempted hours for the course
- earn - earned hours for the course
- pass - passed hours for the course
- qual - quality hours for the course
- pts - quality points for the course
- dtfld - system date
- title - first non-blank title value from the title1, title2, title3, sec_title, or tc_line
- title - section title, not course title (if compile option SECTION_TITLE_ONLY is defined in \$CARSPATH/include/custom/trans "fmtfield" may be used with a format attribute)

16 - Course title

The associated field name on the Transcript Form Table is crs_title.

This line type prints for each extra title line of any course in any session. The first title line is assumed to be printed on the "Course line" (see line type 15) or on the Nontraditional course line (see line type 29). The extra titles come from the Second and Third Course Titles (crs_rec.title2 and crs_rec.title3) and the Section Title (sec_rec.title). This type prints for each of these database fields that are not blank.

The data values for the screen definition file are as follows: The "title" character field contains a title value from one of the following:

- crs_title
- crs_title_line2
- crs_title_line3, sec_title
- tc_line.

The "dtfld" date field contains the system date. The "fmtfield" character field may be used with a format attribute.

17 - Flag descriptions

The associated field name on the Transcript Form Table is crs_flag.

This line type prints for the text descriptions of the course "flags" in the Course Work record (cw_rec) for any course in any session. This type prints once for the Grading Text (grdg_table.txt) if the Grading Print value (grdg_table.pnt) is not zero (0), once for the Counting Text (cntg_table.txt) if the Counting Print value (cntg_table.pnt) is not zero (0), and once for the Repeat Text (rep_table.txt) if the Repeat Print value (rep_table.pnt) is not zero (0).

The data values for the screen definition file are as follows: The "flagtext" character field contains a text value from one of the following:

- cntg_text
- grdg_text
- rep_text

The "dtfld" date field contains the system date. The "fmtfield" character field may be used with a format attribute.

18 - Exam record

The associated field name on the Transcript Form Table is exam_line.

This line type prints for each Exam record (exam_rec) in any session of the transcript. Exam records appear on a transcript only if the Exam Program Code (exam_rec.prog) matches the program of the transcript being printed. Otherwise they are ignored.

The data values for the screen definition file are as follows: Any field from the exam_rec may be used. The "dtfld" date field contains the system date. The "fmtfield" character field may be used with a format attribute.

19 - Exam scores

The associated field name on the Transcript Form Table is exam_score.

This line type prints for any Exam record in any session if the Category (exam_rec.ctgry) has a non-blank Exam Label 1 (exam_table.lbl1) in the Exam table (exam_table).

The data values for the screen definition file are as follows: See line type number 18, "exam_line."

20 - Accomplishment record

The associated field name on the Transcript Form Table is accomp_line.

This line type prints for each Accomplishment record (accomp_rec) in any session of the transcript. Accomplishment records appear on a transcript only if the Accomplishment Program code (accomp_rec.prog) matches the program of the transcript being printed; otherwise, the Accomplishment records do not appear on the transcript.

The data values for the screen definition file are as follows: Any field from the accomp_rec may be used. The "dtfld" date field contains the system date. The "fmtfield" character field may be used with a format attribute.

21 - Accomplishment description

The associated field name on the Transcript Form Table is `accomp_descr`.
This line type prints for any Accomplishment record in any session if the Accomplishment Code (`accomp_rec.code`) has a non-blank Accomplishment Text (`accomp_table.txt`) in the Accomplishment table (`accomp_table`).

The data values for the screen definition file are as follows: See line type number 20, "accomp_line."

22 - Session statistics

The associated field name on the Transcript Form Table is `sess_stats`.
This line type prints as the statistics block for any session except Head, Previous, and Foot. A session prints if any transcript information has been added for that session and year combination. The statistics come after all courses, exams, and accomplishments in the session.

The data values for the screen definition file are as follows: See line type number 8, "head_stats."

23 - Session trailer

The associated field name on the Transcript Form Table is `sess_foot`.
This line type prints as the last line of any session except Head, Previous, and Foot. A session prints if any transcript information has been added for that session and year combination. This will be replaced by the "Column trailer" (see line type 17) if it is the last item printed in the column, as in the preceding example of the transcript format. The screen definition cannot exceed one line.

The data values for the screen definition file are as follows: See line type number 9, "head_foot."

24 - First page footer

The associated field name on the Transcript Form Table is `page1_foot`.
This line type prints at the bottom of all pages of the transcript except the last page.

The data values for the screen definition file are as follows: See line type number 1, "page1_head."

25 - Second page header

The associated field name on the Transcript Form Table is `page2_head`.
This line type prints at the very top of all pages of the transcript except the first page.

The data values for the screen definition file are as follows: See line type number 1, "page1_head."

26 - Foot session header

The associated field name on the Transcript Form Table is `foot_head`.
This line type prints as the header for the Foot session, if required. A Foot session prints if any transcript data has been added for that session. It is not used any other place on the transcript. The screen definition may not exceed one line.

The data values for the screen definition file are as follows: See line type number 4, "head_head."

27 - Comment

The associated field name on the Transcript Form Table is `comm`.
This line type prints for all Transcript Comment records (`trans_comm_rec`) on the transcript.

The data values for the screen definition file are as follows: Any field from the `comm_rec` may be used. The "dtfld" date field contains the system date. The "fmtfield" character field may be used with a format attribute.

28 - Transfer work header

The associated field name on the Transcript Form Table is `trf_head`.
This line type prints before all transfer work from each school in any session.

The data values for the screen definition file are as follows: Any field from the `id_rec` and `sch_rec` may be used, where the `id_rec` and `sch_rec` contain the school information from the `ncat_sch_id`.

The "transfrom" character field contains its default value if the `ncat_sch_id` can be found in the `id_rec` or is negative; otherwise, the value is blank.

The "dtfld" date field contains the system date. The "fmtfield" character field may be used with a format attribute.

29 - Nontraditional course line

The associated field name on the Transcript Form Table is `nontrad_crs`.
This line type prints for each nontraditional course in any session. Nontraditional refers to a course where either the begin or end date in the Course Work record (`cw_rec`) is not equal to zero (00/00/0000).

The data values for the screen definition file are as follows: See line type number 15, "crs_line."

30 - Foot session statistics

The associated field name on the Transcript Form Table is `foot_stats`.
This line type prints as the statistics block for the Foot session, if one is to print on this student's transcript. A Foot session prints if any transcript information has been added for that session. The statistics come after all courses, exams, and accomplishments in the session. It is not used any other place on the transcript.

The data values for the screen definition file are as follows: See line type number 8, "head_stats."

31 - Education session record

The associated field name on the Transcript Form Table is `ed_stats`.
This line type prints for each Education record in any session of the transcript. Education records appear on a transcript only if the "-e" parameter is passed on the calling line to *trans*, either from the menu or from the shell. Otherwise, they are ignored. If the Education record has a valid Graduation Session (`ed_rec.grad_sess`), it prints at the end of that session of the transcript. Any record that has a blank Graduation Session uses the Education record (see line type 3).

The data values for the screen definition file are as follows: See line type number 3, "ed_line."

32 - Foot session trailer

The associated field name on the Transcript Form Table is `foot_foot`.
This line type prints as the last line of the Foot session, if one is to be printed on this student's transcript. A Foot session prints if any transcript information has been added for that session. It is not used any other place on the transcript. This is replaced by the Signature lines (see line type 33) if the signature lines are to be printed, as in the preceding example of the transcript format. The screen definition may not exceed one line.

The data values for the screen definition file are as follows: See line type number 9, "head_foot."

33 - Signature lines

The associated field name on the Transcript Form Table is `sign_line`.
This line type prints after the last session of the transcript. They replace the session trailer of the last session. The signature lines print only on a two-column transcript (i.e., the Transcript Form Columns value [`transfrm_table.cols`] is set to "2").

The data values for the screen definition file are as follows: See line type number 2, "ed_head."

34 - Last page footer

The associated field name on the Transcript Form Table is `page2_foot`.
This line type prints at the very bottom of the last page of the transcript. It is not used on any other page.

The data values for the screen definition file are as follows: See line type number 1, "`page1_head`."

Other Transcript Form Table Fields

The following lists and describes other fields in the Transcript Form table (`transfrm_table`).

active_date

A date field specifying the date (mm/dd/yyyy) when the row becomes active.

cntg_comm

An 18-character field defining the name of the screen definition file to print counting change comments on the transcript.

col_space

A numeric field specifying the number of spaces between columns.

col_width

A numeric field specifying the width of the columns on the transcript.

cols

A numeric field indicating if the form requires one or two columns on the transcript.

ed_sess

A 10-character field defining the screen name to print Education records within transcript sessions having a valid graduation session code.

end_stats

A 10-character field defining the name of the screen file to be used to print the end of transcript statistics.

grd_comm

A 10-character field defining the name of the screen definition file to print grade change comments on the transcript.

grdg_comm

A 10-character field defining the name of the screen definition file to print grading change comments on the transcript.

hrs_comm

A 10-character field defining the name of the screen definition file to print hours change comments on the transcript.

inactive_date

A date field specifying the date (mm/dd/yyyy) when the row becomes inactive.

max_blank

A numeric field indicating the maximum number of blank lines allowed on any column of the transcript.

ofcl

A one-character field specifying if the transcript is official or unofficial.

page_len

A numeric field indicating the number of lines to be printed on one page of the transcript.

transfrm

A four-character code identifying the transcript form.

trnsfr_head

A 10-character field defining the name of the screen definition file to be used as the header on each group of transfer work.

txt

A 24-character description of the transcript form.

user_comm

A 10-character field defining the name of the screen definition file to print user comments on the transcript.

Available Tables

All screens may look up table values from data available on that screen. The following lists the available tables with their database names.

- Academic Status table (acad_stat_table)
- Accomplishment table (accomp_table)
- Certificate table (cert_table)
- Classification table (cl_table)
- Concentration table (conc_table)
- Counting table (cntg_table)
- Degree table (deg_table)
- Exam table (exam_table)
- Grade table (grd_table)
- Grading table (grdg_table)
- Major table (major_table)
- Minor table (minor_table)
- Program table (prog_table)
- Repeat table (rep_table)
- Session table (sess_table)

How to Modify the Format of the Transcript

The printed size of a transcript is 11 inches wide by 8 1/2 inches long, or 110 columns wide by 51 lines long. For those institutions using printers with compressed print, where the printer will print at 13.5 characters per inch to use 8 1/2 inch by 11 inch paper, a minor program code can be made to set the page length for the paper accordingly.

The line length of the header information is limited to the number of characters for wide paper (usually 132). The lines containing the course work information and statistics are limited to approximately 65, which allows spaces between lines. Normally, the width of each column is 55-60 characters. The default format for the transcript is printed on wide paper in a two column format.

It is common to add statistical information to the transcript, such as the resident course work for the student. Resident course information is placed on a line below the session statistics for each session.

In general, the steps for modifying the transcript format are as follows:

1. Locate the data fields containing the information.
2. Modify the files that tell *trans* what the lines will look like.
3. Produce a new transcript.
4. Temporarily install all the elements of the transcript format.
5. Try printing the new transcript format.

How to Locate the Student Statistics Information

The course work grade statistics reside in the student statistics records (`stu_stat_rec` and `stu_acad_rec`) that the Transcript program maintains. An update option that keeps the student statistics records current is available in the Transcript program. Do the following to find the names and uses of the fields in the student statistics records.

1. Enter **cd \$CARSPATH/schema/student** at the shell prompt to change to the schema subdirectory for the student subdirectory.
2. Enter **more stustat** to view the `stustat` schema.
3. Enter **more stuacad** to view the `stuacad` schema.

Note: The two files "stustat" and "stuacad" are the source files from which the database dictionary of fields is built. These specific files keep track of the student grading statistics. The "stustat" file contains the cumulative statistics and the "stuacad" file holds those statistics for the current session. In the database dictionary, the "stustat" file is named "stu_stat_rec" and the "stuacad" file is named "stu_acad_rec".

How to List Student Statistic Database Field Names

Use ISQL, an INFORMIX database tool, to list only names of the fields and their database characteristics. Follow these steps to list only names of the fields and their database characteristics:

1. Enter **isql <\$CARSDDB>** at the shell prompt to access isql tools for viewing table and record fields and to change the prompt to `>`.
2. Enter **info columns for stu_stat_rec;** at the `>` prompt to view the fields in the `stu_stat_rec`.
3. Enter **info columns for stu_acad_rec;** at the `>` prompt to view the fields in the `stu_acad_rec`.

Note: Review the field names that contain the resident statistical information. You need this information when you add it to the Transcript Form file to tell *trans* to print the resident information.

How to Locate Other Files Containing Transcript Information

Other files that may contain information for the transcript are:

- Program Enrollment record (`prog_enr_rec`)
- ID record (`id_rec`)
- Profile record (`profile_rec`)
- Course Work record (`cw_rec`)
- Student Services record (`stu_serv_rec`)
- Registration record (`reg_rec`)

Note: Look at the file `$CARSPATH/src/regist/trans/def.c` to obtain all of the files *trans* uses.

The following fields contain the resident statistical information. The fields come from the Student Statistics record (`stu_stat_rec`).

- `res_att_hrs`
- `res_earn_hrs`
- `res_pass_hrs`
- `res_qual_hrs`
- `res_au_hrs`
- `res_qual_pts`
- `res_gpa`

How to Modify the Transcript Program Screen

The first step to modifying the *trans* files is to modify the program screen file that determines the format of the lines that make up the transcript. The program screens are located in the directory `$CARSPATH/modules/regist/progscr/trans`.

The file that controls the format of the statistics that appear at the end of each session is stored in the file "sessstats." Sessstats, without the revision header information, looks like the following:

```
screen
{
    attempt  earn    pass  quality    points  gpa
ses  [satt ] [searn ] [spass ] [squal ]    [spts ] [sgpa ]
cum  [catt ] [cearn ] [cpass ] [cqual ]    [cpts ] [cgpa ]
}
end

attributes
tatt: optional,
    format = "##&.&&";
tearn: optional,
    format = "##&.&&";
tpass: optional,
    format = "##&.&&";
tqual: optional,
    format = "##&.&&";
tpts: optional,
    format = "###&.&&";
tgpa: optional,
    format = "#&.&&&";
satt: optional,
    format = "##&.&&";
searn: optional,
    format = "##&.&&";
spass: optional,
    format = "##&.&&";
squal: optional,
    format = "##&.&&";
spts: optional,
    format = "###&.&&";
sgpa: optional,
    format = "#&.&&&";
catt = cum_att_hrs, optional,
    format = "##&.&&";
cearn = cum_earn_hrs, optional,
    format = "##&.&&";
cpass = cum_pass_hrs, optional,
    format = "##&.&&";
cqual = cum_qual_hrs, optional,
    format = "##&.&&";
cpts = cum_qual_pts, optional,
```

```

        format = "###&.&&";
cgpa = cum_gpa, optional,
        format = "#&.&&&";
sclk: optional,
        format = "###&.&&";
cclk: optional,
        format = "###&.&&";
dtfield: optional;
        fmtfield: optional;
end

```

Make a copy of this file, and rename it so that it can be used to replace the current `sesstats` file in the new format. The new name may not exceed 10 characters. Use a naming convention that uses the first two initials of the school as part of the new filename. In this example, the CC is for CARS College. The following procedure enables you to create the new file, modify the file, and add it to the Revision Control System (RCS).

1. Enter **cd \$CARSPATH/modules/regist/progscr/trans** at the shell prompt to change to the directory where transcript forms reside.
2. Enter **make add F=CCsesstat** to create the skeleton file.
3. Enter **vi CCsesstat**, press **<Enter>**, and then enter **/screen dG** to eliminate the skeleton file that is to be replaced.
4. Enter **:r sesstats** to read in the original version of "sesstats".
5. Use the command "dd" to delete the header information from the old "sesstats" file.
6. Delete the information preceding the screen definition line.
7. Change the screen definition to look like the following and you will:
 - Delete the old header information.
 - Delete the information preceding the screen definition line.
 - Change the screen definition.

```

-----
screen
{
        attempt earn    pass    quality points    gpa
ses    [satt ] [searn ] [spass ] [squal ] [spts    ]    [sgpa ]
cum    [catt ] [cearn ] [cpass ] [cqual ] [cpts    ]    [cgpa ]
res    [ratt ] [rearn ] [rpass ] [rqual ] [rpts    ]    [rgpa ]
}
end

```

8. Change the attributes section to add the definitions of the added screen fields. It will look like the following:

```

attributes
tatt: optional,
        format = "###&.&&";
You change the attributes section
tearn: optional,
        format = "###&.&&";
tpass: optional,
        format = "###&.&&";
tqual: optional,

```

```

        format = "##&.&&";
tpts: optional,
        format = "###&.&&";
tgpa: optional,
        format = "#&.&&&";
satt: optional,
        format = "##&.&&";
searn: optional,
        format = "##&.&&";
spass: optional,
        format = "##&.&&";
squal: optional,
        format = "##&.&&";
spts: optional,
        format = "###&.&&";
sgpa: optional,
        format = "#&.&&&";
catt = cum_att_hrs, optional,
        format = "##&.&&";
cearn = cum_earn_hrs, optional,
        format = "##&.&&";
cpass = cum_pass_hrs, optional,
        format = "##&.&&";
cqual = cum_qual_hrs, optional,
        format = "##&.&&";
cpts = cum_qual_pts, optional,
        format = "###&.&&";
cgpa = cum_gpa, optional,
        format = "#&.&&&";
ratt = res_att_hrs, optional,
        format = "##&.&&";
rearn = res_earn_hrs, optional,
        format = "##&.&&";
rpass = res_pass_hrs, optional,
        format = "##&.&&";
rqual = res_qual_hrs, optional,
        format = "##&.&&";
rpts = res_qual_pts, optional,
        format = "###&.&&";
rgpa = res_gpa, optional,
        format = "#&.&&&";
sclk: optional,
        format = "###&.&&";
cclk: optional,
        format = "###&.&&";
dtfield: optional;
fmtfield: optional;
end

```

9. Enter **:wq!** to write the changes to the file and exit the file.
10. Enter **make tinstall F=CCsesssstat** to temporarily install the file.
11. Review the file to determine whether there are any errors.
12. If there are any errors, use the vi editor to correct them and then tinstall the file again. The result is that you have a new format for the session statistics.

The new format for the session statistics is now ready to be included in the file that defines the layout of the transcript.

How to Add a New Transcript Form

The following procedure allows you to add the new transcript form that includes the redefinition of the session statistics line. This procedure adds a transcript format file named CCUN; however, you can use any unique four-character name.

1. Select Transcript Form from the Table Maintenance: Registrar (T-Z) menu, then press **<Enter>**. The Transcript Form Table screen appears.
2. Select **Query**, then press **<Esc>** to start a query on the available forms.
3. Select **Next** as many times as needed to display the form defined as WIDE (i.e., WIDE should appear in the Code field of the screen). The form defined as WIDE displays on the Transcript Form Table screen.
4. Select **Add**, then enter a four-character code (e.g., CCUN) in the Code field for the name of the new transcript format file. This is the name of the new transcript format file.

Note: Select any name that is different from the others that are present for the new format.

5. Press **<Ctrl-p>** to ditto or copy in the information that was present in the WIDE format.
 - Repeat this for every line, with the exception of the session statistics line.
 - You will see the name "sesssstats" in one of the brackets. Replace it with the name you entered in step 4 (e.g., CCUN).
6. Press **<Esc>**. The system creates a new transcript definition for an official transcript.

Note: Follow the same steps when you are in Add mode to create a transcript form that will be unofficial, instead of official. Change the flag in the Official? field from Y to N, and use **<Ctrl-p>** to copy all the other format lines to this new unofficial transcript format.

How to Update the Macro File and the Menu Options

You must update the macro file and the menu options to reflect that there is now another unofficial transcript format. Do the following to access the macro file, check out the file, modify the file, check in the file, and reinstall the file. When you have completed this procedure, go to the transcript menu options and reinstall them to bring in the changes you made in the macros they use.

1. Enter **cd \$CARSPATH/macros/custom** at the shell prompt to change to the macros subdirectory.
2. Enter **make co F=table** to lock the table file and prepare to change it.
3. Enter **vi table** to access the vi editor, then look for the following to locate the place in the file where you add your new form name.

```
-----
-----
{*** Transcript Form Table valid and include values ***}
m4_define(`TRANSFRM_VALID', `WIDE,MORE," ")
```



```
m4_define(`TRANSFRM_INCL', `include=(TRANSFRM_VALID), upshift')
```

4. Change the file to read as follows to make CCUN the unofficial form name.

```
-----  
-----
```

```
m4_define(`TRANSFRM_VALID', `WIDE,MORE,CCUN," ")
```

5. Enter **wq!** to save your changes to the file.
6. Enter **make cii F=common L="Updated available transcript forms."** to check in and install the file and document the changes you made.
7. Enter **cd \$CARSPATH/menuopt/regist/programs**
8. Press **<Enter>**
9. Enter **make reinstall F="trns.*"** to reinstall the remaining files.

Checking Transcript Format Modifications

The "-f" option is available with *trans* to pass a default form code. The values in the fields "prog_table.ofcl_trans" and "prog_table.unofcl_trans" in the Program table contain the default values for the transcript form codes if the "-f" parameter is not used.

How to Change the Default Form

Follow this procedure to change the transcript form that is used as a default, by modifying the Program table values used to designate the official and unofficial Transcript form codes.

Note: The form WIDE is the official transcript default form, and UNOF is the unofficial transcript default form in the CX standard product.

1. Select Registrar from the Student Management: Main menu. The Student Management: Registrar Main menu appears.
2. Select Table Maintenance, then enter the password. The Registrar: Table Maintenance menu appears.
3. Select Registrar (M - Q). The Table Maintenance: Registrar (M - Q) menu appears.
4. Select Program/Program Regist., then select **Finish**. The Program table (prog_table) screen appears.
5. Access the correct table mode to modify the transcript form code to reflect the default official and unofficial transcript forms.

How to Change the System Default Form

If a particular transcript form instead of the system default form is to be used by an individual, you must update the individual's Program Enrollment record (prog_enr_rec). A transcript form specified in an individual's Program Enrollment record takes precedence over the system transcript default form.

1. Select Registrar from the Student Management: Main menu.
2. Select Data Entry, then select **Finish**. The Data Entry - Form Menu appears.
3. Select Program Enrollment. The Data Entry - Program Enrollment screen appears.
4. Query first on the selected student, then, when the data appears, select **Next Page** to access the second screen and update the transcript form defaults for the student.

How to Test New Forms

Test the new forms, both in the interactive mode (unofficial transcripts) and the batch mode (official transcripts). Access the menu for transcript processing, and use either the Display Transcripts or the Data Entry menu option to select an unofficial transcript. Use the form CCUN to verify that the resident statistics are being printed and that the values are correct.

To verify that the official transcript form is correct, create a Contact record for a student with the contact resource code of TRANS. Select the menu option to create an official transcript, then use the LPS option (Letter Production System) to print the transcript.

If the session statistics line on the form is acceptable, enter the following commands to install it to make it a permanent part of the Revision Control System:

```
cd $CARSPATH/modules/regist/progscr/trans  
make cii F=CCsessstat L="Modified sessstat for our institution"
```

Note: See *Resolving Transcript Problems* in this technical manual for information about problems you may experience when you modify your transcript forms.

Resolving Transcript Problems

The following list describes problems you may experience as you modify your transcript forms.

No transcript printing

This occurs if the specified form does not exist. Access the Transcript Form table (transfrm_table) from the Registrar: Table Maintenance menu to add the new transcript form, if necessary.

Not a valid transcript form

One of the following occurred:

- The macro file has not been updated to include the new form name.
- The macro file has been updated but has not been reinstalled.
- The macro file has been updated and installed, but the menu option file that uses the TRANSFRM_VALID macro has not been reinstalled.

An unofficial form is required

In the interactive mode, *trans* accepts only unofficial transcript forms. If the form was mislabeled "official" instead of "unofficial," access the Transcript Form table (transfrm_table) and update the form file to indicate that the form is unofficial instead of official. Try the interactive mode of *trans* again to verify that the form is now accepted.

How to Add Page Numbers to the Transcript

The Transcript program will print transcripts either with or without page numbers. If printing with page numbers, the format in the CX standard product is "Page x of y", where the word "Page" can be any page identifying phrase the user desires. Page numbers can be located in either the header or footer of the transcript but not in both places simultaneously.

The following information describes the steps necessary to add page numbers to the format of the transcript.

Note: The locations are given for the files used by the Transcript program.

1. To activate page numbers, go to the \$CARSPATH/modules/regist/progscr/trans directory. Determine which of the transcripts are to have page numbering.
2. Determine if the pages will appear in the header or footer of the transcript page.
3. Check out the particular header or footer by entering the following at the shell prompt: **make co F=filename**

4. Insert the following characters into either the header or footer where the page numbers are to appear on the transcript: **Page [pageid]**

Note: The key here is the word *pageid*. It must be within the brackets followed by *four* (4) spaces. This tells the Transcript program where the "x of y" part of the page numbering goes. The word "Page" can be any page identifying word that the user desires or can be omitted completely if only "x of y" is desired. To eliminate page numbering, simply remove the characters referenced above from the appropriate header or footer.

5. Add **pageid: optional**; in the attributes section of *all* page header and page footer screens even if you do not want page numbering.
6. Once page numbering either is enabled or disabled, enter **make cii F=filename** to checkin and reinstall the affected headers and/or footers.

Note: The CX standard product has page numbering turned on in all of the official and unofficial (wide) transcript headers.

Screen Usage to Prevent Data from Printing

The Transcript Form table (transfrm_table) may be used not only to define the screens used to format the various types of data, but also to prevent specific types of data from printing on the transcript.

Example: You want to create a format that prints exactly like the standard official format except that no Exam records or Accomplishment records are to be printed. A new record can be added with the fields "exam_line," "exam_score," "accomp_line," and "accomp_desc" blank. If the screen name is blank in the Transcript Form table for any type of line, that type of line is omitted from the transcript output.

One of the special features used in the screen definition files is the format attribute on character fields. An example of a screen that uses this feature is the session header. On character type screen fields, the format clause allows one or more screen fields to be concatenated and printed within that field. It allows any field name to be referenced using "CSH" shell variable syntax.

Consider the following example screen.

```
screen
{
fmtfield                               ]
}
end

attributes

sesstext = sess_text, optional;
sess = ac_sess, optional,
      lookup sesstext joining sess_table.sess_tbcode;
sessyr = ac_yr, optional, format = "<<<&";
fmtfield: center, optional, format =
"----- ${sesstext}${sessyr}-----";
...
```

The field that actually prints is "fmtfield". First, notice that in the screen definition there is no opening bracket ([) for the field. This allows the data from the field to be printed in the first column of the screen. Also, notice that this field contains the data from the "sesstext" field (which comes from a table lookup on the Session table), the "sessyr" field, and some other characters not coming from the database. When the data is formatted into "fmtfield", the blanks are removed from the end of each field before it is inserted. The "center" attribute is also assigned to "fmtfield." This causes the data that is printed to come from the middle of the resulting format. This produces the following results:

```
----- Spring 200X -----
```

Default Attribute

Another special feature used in the screen definition files is the default attribute. The transfer work header is an example of a screen that uses this feature. Some of the screens contain special field names that either are blank when printed or contain their default value. Consider the following example screen:

```
screen
{
  Transfer work[fmtfield                ]
}
end

attributes

transfrom: default = "from", optional;
trfname = name, optional;
fmtfield: optional, format = "${transfrom}${trfname}";
...
```

Notice that from the description of the "ttransfrm_trf_head" field, the field "transfrom" contains its default value only if the ncat_sch_id field is a valid ID number or is negative. If the Noncatalog record contained the ID number of Ohio State University, as in the transcript example, the result would be as follows:

```
Transfer work from Ohio State U.
```

However, if the Noncatalog record contained an ID number of zero (0), the following would result:

```
Transfer work
```

Screen Usage to Print Blank Lines: a Non-Working Example

Another special use of screen definition file features is printing a screen with blank lines at the end. The education header screen is an example of this feature. In any screen definition file, the system ignores all blank lines between the last non-blank data and the closing brace (}). Suppose one of the screen files was to be changed to print only two blank lines. The following screen would not work:

```
screen
{
}
end
...
```

Screen Usage to Print Blank Lines: A Working Example

Remember that the system ignores all blank lines before the closing brace (}), which is the entire screen definition here. The "fmtfield" field may be used to cause the blank lines to print, as shown in the following example:

```
screen
{
  [fmtfield]
}
end

attributes

fmtfield: optional, format = "          ";
...
```

Since the last line before the closing brace (}) is not blank, the system ignores no lines of the screen definition. Also, since the format for the "fmtfield" field is blank, the value printed is always blank.

Screen Usage to Print Inconspicuous Character

Another way to make a transcript appear to have blank lines would be to use an inconspicuous character on the last blank line, as shown in the following example:

```
screen
{
.
;
}
end
...
```

Since the last line before the closing brace (}) contains a period (.), the system ignores no lines of the screen definition. The inconspicuous character may appear anywhere on the line, as long as the last line to be printed has a non-blank character.

Inclusion of Course Work in Progress

The Registration program assigns the grade of "IP" for "In Progress" to each course at the point in which a student is registered for the course. The Grading program will assign the grade "NR" for "Not Reported" to any course the program calls where a grade is not entered. These grades should be included in the institution's Grade table (grd_table) without a grading type code.

Once a student's registration status has changed to "C" for "confirmed" in the Course Work records for the student, the courses can appear on the transcript. They will be calculated into session and cumulative hours and grade point averages if the operator specified use of current session work as one of the initial parameters to the Transcript program. This may include the grades of "IP" and "NR" so it is important that the attempted, earned, passed, and quality factors be set accordingly. If "IP" and "NR" grades are not to be included on transcripts, make sure that the print flag is set to "0" in the Grade table.

All courses will be flagged if the student is repeating a previous course, regardless of the grade assigned to either course. Courses with instructor-assigned grades will affect the cumulative statistics; these effects are governed by the setup of the Repeat table and the way the constant variables are customized. However, repeated courses with non-instructor assigned grades (primarily, grades of "IP" and "NR") do not affect the cumulative statistics.

Transcript Sort Keys

Introduction

The Transcript program does not use an alphabetic sorting algorithm to sort the items on the transcript; instead, it uses a complex algorithm to determine the positioning of transcript items.

Types of Sorting

The Transcript program uses three types of sorting:

- Session sorting
- Item sorting
- Group sorting.

Session Sorting

The system sorts sessions in the following manner:

- "HEAD" session
- "PREV" session
- year value (ac_yr), as defined in the Academic Calendar record (acad_cal_rec)
- session offset value (sess_yr_offset), as defined in the Session table (sess_table)
- "FOOT" session

Item Sorting

The following describes how the system sorts items within a session:

Comments before session header

Sort key - #999999999

Type – Sorted in the Transcript Comment record (trans_comm_rec)

Session header

Sort key – \$SESSYEAR

Type – Generated by the Transcript program

Comments after session header

Sort key - %999999999

Type – Sorted in the Transcript Comment record (trans_comm_rec)

If USE_TW_BLOCK is uncommented in \$CARSPATH/include/custom/trans, the following apply:

Text for block transfer

Sort key – cwnoa

Type – Sorted in the Transcript Comment record (trans_comm_rec)

Statistics for block transfer

Sort key - cwnob

Type – Sorted in the Transcript Comment record (trans_comm_rec)

Comments on courses

Sort key – cw_no [A-Z0-9]

Type – Sorted in the Transcript Comment record (trans_comm_rec)

First course title

Sort key – crs_noU

Type – Generated by the Transcript program

Second course title

Sort key – crs_noV
Type – Generated by the Transcript program

Third course title

Sort key – crs_noW
Type – Generated by the Transcript program

Section title

Sort key – crs_noX
Type – Sorted in the Transcript Comment record (trans_comm_rec)

Grading line

Sort key – crs_noY
Type – Generated by the Transcript program

Counting line

Sort key – crs_noZ
Type – Generated by the Transcript program

Repeat line

Sort key – crs_no[
Type – Generated by the Transcript program

Grade change line

Sort key – crs_no]
Type – Sorted in the Transcript Comment record (trans_comm_rec)

Comments after courses

Sort key – a999999999
Type – Sorted in the Transcript Comment record (trans_comm_rec)

Accomplishments

Sort key – b(date)#
Type – Generated by the Transcript program

Exams

Sort key – c(date)#
Type – Generated by the Transcript program

Comments before statistics

Sort key – d999999999
Type – Sorted in the Transcript Comment record (trans_comm_rec)

Statistics lines

Sort key – e[1-4]
Type – Generated by the Transcript program

Comments after statistics

Sort key – f999999999
Type – Sorted in the Transcript Comment record (trans_comm_rec)

Signature lines

Sort key – g[0-6]
Type – Generated by the Transcript program

Education record

Sort key - !9sequential#)
Type – Generated by the Transcript program

Group Sorting

The system sorts items within the same group in the following manner:

If a line is a block transfer line, the sort is based on the Block Transfer Sort (blktw_sort) field in the Block Transfer record (blktw_rec). You enter this field when adding a block transfer with Transcript Edit.

- In the future, comments associated with a block transfer will be in the following format, 'blktw_sort[A-Z0-9]'.
- If a line is a course line (sort[0] >= '0' and sort[0] <= '['), the lines are sorted by the School ID (sch_id) from the School record (sch_rec) and by the Course Work Number (cw_no) from the Course Work record (cw_rec).
- All other lines are sorted on the basis of the sort value.

Preparing Hold Codes

Introduction

CX uses three tables to control holds (restrictions) on accounts. The following describes the use of each table.

- The Hold table (hold_table) can restrict some functions (e.g., student accounts) from the student or another individual.
- The Hold Action table (hold_act_table) works in conjunction with the Hold table. As a result of a Hold table entry, the Hold Action table specifies if the hold is absolute or a warning hold.
- The Office Permissions table (ofcperm_table) can grant permission to an office and individual for updating holds.

The general sequence for making entries to these tables is as follows:

1. Create a Hold table entry.
2. Create the corresponding Hold Action table entry.
3. Create the Office Permission table entry for the specific hold.

Example: The Hold table has a code for student accounts, and the Hold Action table has an entry for transcript, with a Hold Action table entry of absolute. If a student with such a hold is delinquent in paying the student fees, an absolute hold is placed and the transcript cannot be released until the hold is cleared.

CAUTION: Absolute holds should be used with discretion, since they prevent some action and can create additional work if used carelessly.

How to Set up Hold Codes

The following lists and describes the process to set up hold codes. See the Getting Started User Guide and the *CX System Reference Technical Manual* for additional information about entering data in the hold table screens, examples of the hold table screens, and examples of hold table reports.

1. Select Office Permissions Report from the Table Maintenance: Common (G - O) menu to print out the Office Permissions report (tofcperm) for use in assigning user permissions within the Hold table.
2. Define the various types of holds you desire, and enter them through the Hold Table screen.

Example: REG (Registrar Office Hold)
BUS (Business Office Hold).

3. Define the hold action and absolute value (Y or N), and enter them through the Hold Action Table screen.

Example: REGIST Y (blocks the Registration Entry program)
REGIST N (warns operator)

Note: REGIST (Registration Entry program) and TRANS (Transcript program) are valid codes to establish blocks. Additional mailing holds may be created in the adr_rec (adr_rec.run_code, i.e., SDSBILL, when run in batch). The REGIST code is defined in \$CARSPATH/macros/custom/student under REGIST_HOLD_ACT.

4. Assign the office code and user ID of the person permitted to add and/or update hold codes

assigned by an office on an individual student record.

Note: This does not relate in any way to the other fields on the screen. If operators have an entry here, they can change the Hold Table for a student; if not, they cannot.

5. Run the following reports to check all table entries.

- Hold Table Report (thold)
- Hold Action Table Report (tholdact)
- Office Permissions Report (tofcperm)
- Office Table Report (tofc)

Note: See the *CX System Reference Technical Manual* for examples of these reports.

Helpful Hints Using Hold Codes

The operator's user ID is linked through the Office Permission table to the office added by the code that appears on the screen. In many programs the office code is passed in the menuopt and cannot be changed interactively. If the operator has permission to add and/or update holds related to the office code, the operator may continue processing. If not, a message appears denying the operator access.

Note: Never delete a hold code; instead, update the End Date field. If you want the hold to be terminated as of today, you must back date the End Date field to yesterday.

Maintaining Historical Information on Session Records

Introduction

Institutionally, it is important that a limited amount of Program Enrollment record (prog_enr_rec) information be retained for historical purposes. The Program Enrollment record is a single record per student per program. The record constantly changes, making it difficult to find information from previous years. To meet the historical data requirements, CX provides a process to update the student's Academic record (stu_acad_rec) as part of the registration process. Some basic columns (fields) of data such as the student's classification, academic status, resident state, and subprogram are defined in the CX standard product. Accordingly, registration updates these four columns with data from the Program Enrollment record when a student registers. This is appropriate since the student's Academic record is created for each session/year in which the student registers.

The institution can define additional columns of data that it wants to maintain historically from the Program Enrollment record.

The data retained is typically the information that was current at the beginning of the term. It is a common practice to keep this information current up to the date the institution determines to be its reporting date or statistical date. Since the data the registration process initially updates may change before the reporting date, it is important that any information changed on the Program Enrollment record be updated on the appropriate Academic record. The menu contains an option to run a process that, at the institution's discretion, updates the Academic record and/or provides a report of the inconsistencies between the two records.

Although the majority of the changes in historical data are accomplished either by registration or by the menu update process, there are situations where the student's classification may be changed by the update process in Transcript. However, the update process in Transcript will update a student's classification only to what it should be as of the beginning of a session. Any hours that a student earns during a particular session will not apply toward the classification the student had at the beginning of that session, but instead will apply toward the classification that the student will have at the beginning of the next session. In the CX standard product, a student's current classification (according to the number of hours or credits a student has taken) will be reflected in the classification stored in the student's Program Enrollment record (prog_enr_rec) or in the student's Academic record (stu_acad_rec) in the "cl_end" field.

Of all the possible historical data that could be retained, the student's classification is the most important to the system, since it has student billing implications. Jenzabar, Inc. recommends that no changes be made to this data field on the CX standard product.

CAUTION: Never update historical information for past terms. If you select a past session and year to update, the system replaces all of the historical information for the past term with current information. In other words, the integrity of the past term's historical data is destroyed. The registration process in CX does not permit this type of data corruption; however, external processes not related to CX could influence this data. Exercise extreme caution when modifying Session records.

Setup

After determining the data that should be retained institutionally, review and update the macros (includes) in \$CARSPATH/include/applic/regent. Follow the instructions precisely if changes are to be made. The two includes that need to be reviewed are the following:

- STUAC_AUD_FIELDS
- STUAC_AUD_LEN

STUAC_AUD_FIELDS

The definition `STUAC_AUD_FIELDS` is used to link the fields in the Academic record (`stu_acad_rec`) to the Program Enrollment record (`prog_enr_rec`) for audit trail purposes. The fields listed will be copied from the `prog_enr_rec` to the `stu_acad_rec` when it is created through the Registration program. The first element is the field name from the `prog_enr_rec`; the second element is the field name from the `stu_acad_rec`. The definition should appear in the following way:

```
#define STUAC_AUD_FIELDS { \  
    {"cl", "cl"},\  
    {"subprog", "subprog"}, \  
    {"res_st", "res_st"}, \  
    {"acst", "acst"} \  
    {"prog_enr_field", "stu_acad_field"}, \  
}
```

The braces (`{}`), double quotes (`"`), and the backslash (`\`) are needed. Also note that the comma is outside the braces on all but the last line.

The CX standard product provides for the four fields specified. Any changes to this include, whether additions or deletions, also should be made to the menu process that updates the data after registration. The SQL script for this processing is in `$CARSPATH/modules/regist/informers/updstuac`.

Also, the menu process provides for a report that will identify the Program Enrollment record and Academic record inconsistencies. Changes to the include also should be made to the report in `$CARSPATH/modules/regist/reports/lststuac`.

STUAC_AUD_LEN

Example: `#define STUAC_AUD_LEN 4`

The definition `STUAC_AUD_LEN` is the number of lines that are defined in `STUAC_AUD_FIELDS`. The number in the include must exactly match the number of data fields in `STUAC_AUD_FIELDS`. For example, if one additional data field is added to `STUAC_AUD_FIELDS`, the number in `STUAC_AUD_LEN` must be changed from 4 to 5.

If you make changes to the includes, modify the SQL script accordingly. The SQL script is located in `$CARSPATH/modules/regist/informers/updstuac`. Also, modify the report in `$CARSPATH /modules/regist/reports/lststuac` as necessary.

Notes:

- The informer, the report, and the includes must be consistent.
- A script runs the SQL script. The script, which normally requires no changes, is in `$CARSPATH/modules/regist/scripts/updstuac`.

Automatic Update

After the setup is accomplished, the processes are ready for use. As students register, the system creates their Academic records for the session. As part of the registration process, the data fields as specified in the include `STUAC_AUD_FIELDS` are updated to the Academic record with the current values contained in the Program Enrollment record. This is the only time CX automatically updates the data.

Cut-Off Date for Updates

Since the data on the Program Enrollment record may change after the system creates the Academic record, run the menu process to update the latter record, if so desired. The institution should determine the cut-off date for updating the historical information on the Session record. Typically, this cut-off date corresponds to the institution's statistical date. Jenzabar, Inc.

recommends, at a minimum, that this update process be run on this cut-off date to ensure the historical data will correspond to the data used in the official reports for the session.

Update Process

Use the Academic Record Updates menu option on the Registrar: Session Processing Menu to begin the update process. It can be run for any of the following three options:

- L (List) - provides a report of the records that have inconsistencies between the Program Enrollment record and the Academic record.
- U (Update) - updates the Academic records with current Program Enrollment record information.

CAUTION: Never run the update for past terms. If you select a past session and year to update, all of the historical information for the past term is replaced with current information. In other words, the integrity of the past term's historical data is destroyed.

- B (Both) - provides the report of inconsistencies and updates the records without having to do them as separate processes.

Testing the Processes

It may be desirable to test both the registration and update processes on a few students to do the following:

- Ensure that the institution understands them
- Validate that the proper data is being updated

Updating the Student's Classification

The update process in Transcript may update the student's classification when it is run. For example, during the spring term, a student registers early for fall. At the time of registration, the student's classification was freshman, so this classification was put on the Academic record by the registration process. However, as part of grade processing for spring, the student completed enough hours to be a sophomore. Therefore, the update process in Transcript that is run after grades are input will update both the Program Enrollment record and the fall Academic record to sophomore.

Printing Academic Statuses

The Transcript program provides the ability to print academic statuses on transcripts. At the institution's discretion, transcripts can be printed containing overall or session academic statuses.

Setting Up and Using Course Equivalencies

Introduction

An equivalent course is a course using a level of instruction equal to that of another course with a different name and number. The system evaluates equivalent courses the same when it determines degree requirements for a student. The information on these pages discusses the following three records or types of records that CX uses for equivalency purposes:

- Course Equivalency record (crseq_rec)
- Course History record (crshist_rec)
- Transfer Equivalency record (e.g., noncateq_rec)

Example of Equivalent Courses

It is common that some courses within a catalog are identical courses (i.e., each course has the same title, credit hours, and course content). As an example, the course PSY313, Early Childhood Education, is the same course as EDU313, Early Childhood Education. However, the psychology major would take the course as PSY313, while the education major would take the course as EDU313.

Equivalent Courses in Jenzabar CX

There are a number of valid reasons for equivalent courses. Equivalent courses are known by many different terms; one common term is a statement in the course description for PSY313 such as "Same as EDU313." Accordingly, the equivalent courses also are referred to as "same as" courses. An institution typically offers the equivalent courses in the same classroom, at the same time and on the same days, and with the same instructor. If so, the institution can cross-list the equivalent courses as the institution develops the courses, sections, and meeting times for a session. For more information about cross-listing sections, see *Cross-Listing Sections* in the *Course/Class Schedule Technical Manual*.

CX applies repeat logic to equivalent courses (e.g., a student fails PSY313 and retakes the course as EDU313). Equivalent courses satisfy prerequisite checking (e.g., a course with a prerequisite of PSY313 is satisfied if the student takes the course as EDU313). Equivalent courses also satisfy program audits (e.g., either PSY313 or EDU313 would satisfy a requirement for an upper-division course in Psychology).

How to Locate Course Equivalencies

You must add the proper records before you can fully use the three types of records CX uses for course equivalency purposes. Access the Registrar: Course/Class Schedule menu, select the Catalog Maintenance menu option, enter the password, and then select **Finish** to access the Catalog Maintenance screen. Enter the code for the session to which the course equivalencies apply, and then select **Finish**. Select "Course catalog and schedule" from the Catalog Maintenance Options window that appears, and then select **OK** to access the Course Catalog screen to enter equivalent type records for each of the three record types.

After you access the Course Catalog screen, use **Query** to select a course for which you want to add or update an equivalent course entry. When selecting Options from the Course Catalog screen, a drop-down menu appears containing several options.

Following are the three options from the drop-down menu in the Course Catalog screen that apply to course equivalencies:

- Course Equivalencies
- Transfer Equivalencies
- Course History

Also, you can print reports to determine this information. For information about how to produce, for example, the Course History report, see the section *Using the Course/Class Schedule Reports* in the *Course/Class Schedule User Guide*.

Setup Process - Course Equivalencies

The following lists and describes the overall process involved in setting up course equivalencies:

1. Determine which courses in a catalog are equivalent. For example, the course PSY313 and EDU313 are equivalent in the UG01 catalog.
2. Query for the desired course and catalog (e.g., the course EDU313 in the UG01 catalog). Select Options from the Course Catalog screen, and then select Course Equivalencies from the drop-down menu that appears. After you select Course Equivalencies from the drop-down menu, the Equivalencies Within Catalogs window appears. For information about how to complete the fields on the Equivalencies Within Catalogs window, see the section *Using the Course/Class Schedule Screens and Windows* in the *Course/Class Schedule User Guide*.

Notes:

- You can access the course PSY313 in the UG01 catalog or the course EDU313 in the UG01 catalog. You do not have to enter each of the courses individually to add equivalent courses. Entering the equivalent information between two courses establishes all other equivalent links between the courses. Use the Add command to add additional equivalent courses. You do not have to enter every possible combination of courses.
- You must make only one entry for each course in the sequence. Data that you enter in this process is retained in the Course Equivalency record (crseq_rec).

Example of Equivalencies Within Catalog Window

An example of the Equivalencies Within Catalogs window appears in the *Course/Class Schedule User Guide*.

Effects of Using Course Equivalencies

The following lists and describes the CX applications that use course equivalencies.

Registration

If a student attempts to register for a course for which the student has previously taken an equivalent course, the Repeat History window appears indicating that the student has taken the course previously. For example, the student attempts to register for the course EDU313, and the system displays the window showing that the student took the course previously, not as the course requested, but as the equivalent course(s). The Repeat History window also shows that the student actually has taken both equivalent courses SOC313 and PSY313. Based on the information that appears, the menu user can decide about further registering the student in EDU313.

If a course has a prerequisite of any of the three equivalent courses, any one of the three courses satisfies the prerequisite checking, as long as the student has the prerequisite minimum grade requirement. For example, EDU403 has a prerequisite of EDU313, with a minimum grade of "D." If the student tries to register for EDU403, either SOC313 or PSY313 would satisfy the EDU313 prerequisite if the student received a "D" or better grade in the course.

Transcript

The system applies repeat logic to the equivalent courses.

Example: The student received a "D" grade in SOC313, retook the course as PSY313, and received a "B" grade. Using repeat logic, the system replaces the SOC313 entry with the PSY313 entry.

Note: The system in this case allows the last grade to count in repeat logic. You can set up CX so that repeat logic can be applied to accept either the last grade earned or the higher grade of the two courses.

Program Audit

Any criteria that specify any one of the three equivalent courses is satisfied either by the specific course in the criteria or by any of the equivalent courses. The student also must satisfy the minimum grade requirement. For example, if the criteria specifies EDU313, then PSY313 satisfies the criteria, and the system prints the criteria on the audit accordingly.

Example of Repeat History Window

An example of the Repeat History window appears in the *Registration User Guide*.

Tracking Course Renumbering

Because an institution can renumber courses, CX includes a process to keep track of these changes on a historical basis.

Example: An institution changes MAT110 to MAT102, effective with the UG99 catalog. MAT102 in the UG99 catalog is the same course as MAT110 before the UG99 catalog. MAT110 loses its former identity when MAT102 originates. If MAT110 is still used in the UG99 catalog or later, it is a different course from the one before the UG99 catalog.

Historical Course Numbering in Jenzabar CX

Jenzabar CX considers historical course numbers in repeat logic and prerequisite checking, and treats program audit requirements the same as equivalent courses described previously.

Example: The course MAT201 in the UG01 catalog has a prerequisite of MAT102. When a student attempts to register for MAT201, the system determines whether the student has completed MAT101 in the UG01 catalog. If not, the system further checks to determine whether the student completed MAT110 in a catalog prior to UG01. If the system finds that the student completed MAT110 in an earlier catalog, prerequisite checking is successful.

A typical institution changes a course number several times over the years. The course history process links these changes together.

Course History Setup Process

The following lists and describes the overall process involved in setting up the course history for equivalent courses.

1. Determine which courses in a catalog are equivalent.
2. Query for the desired course and catalog (e.g., the course MAT102 in the UG01 catalog). Select Options from the Course Catalog screen, and then select Course History from the drop-down menu that appears. After you select Course History from the drop-down menu, the Course History window appears. For information about how to complete the fields on the Course History window, see the section *Using the Course/Class Schedule Screens and Windows* in the *Course/Class Schedule User Guide*.

Note: Remember that if the course MAT110 is in the UG01 catalog or a later catalog, then MAT110 is a different course from the MAT110 before the UG01 catalog. If the course

MAT110 is in a catalog prior to the UG01 catalog that also resulted from a course number change, Jenzabar, Inc. recommends that you also add the earlier course to the Course History windows by selecting the Add command. You can add as many of these entries as required. The Course History record (crshist_rec) retains the data that you enter in this process.

Example of Course History Window

An example of the Course History window appears in the *Course/Class Schedule User Guide*.

Effects of Using Course History

The following identifies and describes the CX applications that use course history.

Registration

If a student attempts to register for the course MAT102 in the UG01 catalog and has previously completed MAT102 in the UG01 catalog, or MAT110 in earlier catalogs, a window appears showing the previous courses taken. Based on the information that appears, a menu user can decide about further registering the student in MAT102.

If a student attempts to register for a course that has MAT102 as a requisite, any previous completions to the sequence explained above meet the requirements. The student also must satisfy the minimum grade requirement. For example, a prerequisite of the course MAT102 in the UG01 catalog with a minimum grade of "D", is satisfied by MAT110 in an earlier catalog if the student received a grade of "D" or higher.

Transcript

The system applies repeat logic to the historical courses similar to the process explained for course equivalencies. For example, a student earned a grade of "D" for MAT118 in an earlier catalog. The student retook the course as MAT102 in the UG01 catalog and received a grade of "B." The latter course (MAT102 in the UG01 catalog) replaces the MAT118 course.

Program Audit

Using the criteria of MAT102 in the UG01 catalog in an audit is satisfied if the student successfully completes one of the historical courses and meets the minimum grade specified.

Transfer Equivalencies in Jenzabar CX

When a significant number of students transfer from the same institution to another institution, the receiving institution frequently evaluates the same transfer course over and over to the same equivalent course.

The Transfer Equivalency records serve as default equivalent courses when the institution enters transfer courses through the Edit function in the Transcript program (*trans*). If the default equivalent course does not apply, an institution can enter a different equivalent course on a case-by-case basis. Only one transfer equivalent course can be linked to a single course and catalog.

Example of Transfer Equivalencies

ENGL115 from CARS College satisfies XYZ University's ENG101 requirement. Since XYZ University receives a number of transfers from CARS College, the receiving institution (XYZ University) must evaluate this general education course and enter ENG101 as the equivalent course when entering ENGL115 in the Edit function of Transcript. CX contains a process that associates ENGL115 from CARS College to the receiving institution's course of ENG101. This process simplifies evaluating the transfer transcript, minimizes data entry, and ensures that the transfer courses are applied uniformly for all students.

Transfer Equivalency Setup Process

The following lists and describes the overall process involved in setting up transfer equivalency.

1. Determine the courses within a catalog for which there is an equivalent course from another institution (e.g., the course ENG101 in the UG01 catalog).
2. Query for the desired course and catalog. Select Options from the Course Catalog screen, and then select Transfer Equivalencies from the drop-down menu that appears. After you select Transfer Equivalencies from the drop-down menu, the Transfer Equivalencies window appears. For information about how to complete the fields on the Transfer Equivalencies window, see the section *Using the Course/Class Schedule Screens and Windows* in the *Course/Class Schedule User Guide*.

Note: The data entered in this process is retained in the Noncatalog Equivalency record (noncateq_rec).

Example of Transfer Equivalency Window

An example of the Transfer Equivalencies window appears in the *Course/Class Schedule User Guide*.

Effects of Using Transfer Equivalencies

Transfer equivalencies are used only to enter transfer work through Transcript Edit. Transfer equivalencies assist in evaluating transfer work for courses that are frequently transferred.

Note: An explanation of Transcript Edit appears in the section *Editing Transcripts* in the *Transcript User Guide*.

Example: A student transfers ENGL115 from an institution with ID 16120, CARS College. This course usually meets ENG101 requirements at the institution to which the student is transferring. When a menu user enters the school ID 16120, course ENGL115, the system displays the credit hours, grading type, and counting code of the transfer equivalent course of ENG101. Also, the second line of the entry displays the equivalent course and catalog. The menu user can complete the transfer entry, changing any of the information that has defaulted to the screen as a result of setting up the two courses as transfer equivalencies.

Setting Up and Using Repeat Logic

Introduction

CX has six basic entities in the repeat logic process:

- **The includes used for repeat logic**
The RESREP, REP_BY_GRD, CUM_BY_LAST, and MAX_REP includes are located in the following directory path: \$CARSPATH/include/custom/trans.
- **Course record (crs_rec) repeat logic**
The Course record is located in the directory path \$CARSPATH/schema/student/crs. The applicable Course record screens are located in the directory path \$CARSPATH/modules/regist/progscr/crsent/course. The applicable entries are as follows: crs_rec.rep, crs_rec.max_rep, and crs_rec.hrs_max_rep.
- **Grade table (grd_table) repeat logic**
The Grade table is located in the directory path \$CARSPATH/schema/student/tgrd. The applicable Grading screen is located in the directory path \$CARSPATH/modules/regist/screens/tgrd. The applicable columns are as follows: grd_table.instr_assign, grd_table.val, and grd_table.rep.
- **Repeat table (rep_table) repeat logic**
The Repeat table is located in the directory path \$CARSPATH/schema/student/trep. The applicable screen is located in the directory path \$CARSPATH/modules/regist/screens/tflags.
- **Course Work record (cw_rec) repeat logic**
The Course Work table is located in the directory path \$CARSPATH/schema/student/cw.
- **Transcript program (trans) repeat logic**
The Transcript program is located in the directory path \$CARSPATH/src/regist/trans.

Setup Process - Repeat Logic

The following lists and describes the overall process involved in setting up repeat logic:

1. Understand the institution's repeat policy by direct consultation (e.g., needs assessment) and reading their catalog. Determine whether the standard CX repeat logic meets their requirements. Direct any unusual conditions to the Jenzabar, Inc. representatives most knowledgeable of the repeat logic and campus repeat policies.
2. Define the includes according to the institution's specifications.
3. Ensure that the client understands the implications of the entries in the Course record (crs_rec).
 - A "Y" in crs_rec.rep means that the course can be taken over for credit; the number of times depends on the values in crs_rec.max_rep and crs_rec.max_rep_hrs. Zero means unlimited. A "Y" in crs_rec.rep results in Repeatable (C) repeat logic (this value in the crs_rec points to the rep_table.ctgry column).
 - An "N" in crs_rec.rep means that a student cannot retake the course for credit, and if the student can retake the course, the attempt is to improve the grade earned previously. The value in crs_rec.max_rep or max_rep_hrs specifies the number of times/hours the course can be retaken to improve the grade, with zero meaning unlimited. If the student can retake the course only once to improve a grade, enter a 2 (to define the original time and the one time the course can be retaken). An "N" in rep applies both Repeated (R) and Substitute (S) logic to the cw_rec.
 - A blank in crs_rec.rep results in the system applying the same repeat logic as if you entered an "N" in the column.
 - If courses exist that a student can take more than once before earning credit, two columns, cw_rec.cr_rep and cr_max_rep, are available for this purpose.

In one possible setup, the `cr_rep` column specifies the number of times a course must be taken (e.g., five) before any credit is counted, and the `cr_max_rep` column provides a limit to the number of times this can occur (e.g., 15). This means that a student can go through this cycle only three times since 15 divided by five equals three. These columns do not appear on the Course Entry screens in the CX standard product.

4. Set up the Grade table (`grd_table`) properly. Following are the most important columns for applying proper repeat logic:
 - `grd_table.instr_assgn`
You must make an entry; the column cannot be blank. In general, the value should be "Y" for all grades except the system grades, IP and NR.
 - `grd_table.rep`
Enter a "Y" in the column for all grades to which you want repeat logic applied. Set all other grades to "N."
A common setting is for D and F grades to have `rep="Y"` and A, B, and C grades to have `rep="N"`.
 - `grd_table.val`
You must make a non-zero entry in the column, if `grd_table.rep` column is "Y". Otherwise, the system does not apply the repeat logic.

Note: In the `grd_table.val` column, the higher the value entered, the higher the grade. This is important when the system retains the higher grade (as controlled by the `REP_BY_GRD` include).

5. Understand the Repeat table (`rep_table`).

CAUTION: Do not make any changes to the `rep_table` without first consulting the Jenzabar coordinator, application coordinator, and Jenzabar, Inc. representatives.

If unusual repeat conditions apply, consider using manual repeat codes (`rep_table.ctgry = M`) or alternative methods such as introducing a new counting type in the `cntg_table`. Manual repeat codes apply statistics exactly as specified in the table for attempted, earn, pass, and quality.

Note: No program code exists to support these codes; therefore, the update function in *trans* ignores (does not reset) the codes, and any courses with manual repeat codes are not counted in `crs_rec.max_rep` or `max_rep_hrs`.

6. The `crs_rec.rep` column contains the repeat codes assigned automatically by the system or assigned manually by the institution. Only the institution can assign manual repeat codes. The Transcript program sets and resets all other categories (C, R, S) automatically, depending on the entries in the includes, `crs_rec`, `grd_table`, and `rep_table`.

Testing Process - Repeat Logic

After accomplishing the setup, perform repeat logic test cases by producing transcripts to verify that the repeat logic performs as expected. The number and type of test cases depends on the institution's policy. The following lists and describes the overall process involved in testing repeat logic.

1. Set up test cases to test both the Repeatable (C) and Repeated (R) repeat logic categories as specified in the `rep_table.ctgry` column. These categories relate to the values of "Y" and "N" respectively in `crs_rec.rep`.
2. Ensure that the limits established by non-zero entries in the columns `crs_rec.max_rep` and `max_rep_hrs` apply the proper repeat codes. After reaching any assigned limits, the repeat codes that are applied change (apply different repeat codes) for any additional repeats for the course.

3. Ensure that the system retains the proper grades and computes the grade point averages properly, based on the way the includes are defined. All adjustments to the statistics are made in the session in which the course was repeated.

Note: Several test cases are specified and supported by transcript output on the following pages dealing with test cases.

Troubleshooting Issues - Repeat Logic

The following lists the identified issues and possible solutions to use when troubleshooting repeat logic:

Possible change in the standard rep_table

Check the rep_table to ensure that it has not been changed from the original setup.

Association between crs_rec columns rep, max_rep, and max_rep_hrs, and the rep_table column rep

Check the crs_rec for the course(s) under review to determine the setup of the rep, max_rep, and max_rep_hrs columns. Based on these values (rep="Y" applies C category logic and rep="N" applies R category logic), you know what rep_table codes are applied. The rep_table.rep CU/CO codes depend on the values in max_rep/max_rep_hrs. Typically, the system applies the CU code when the student has taken the course(s) within the values specified, and the CO code if these values are exceeded. The system applies rep_table.rep values of RU/RO if the value in crs_rec.rep="N." The RU/RO and CU/CO codes in rep_table.rep are standard product codes.

Varying repeat logic applies to a course with a non-repeatable grade

If policy allows a student to retake a course with a non-repeatable grade (as specified in grd_table.rep="N"), determine the repeat logic value applied. The system applies the grd_table.rep values of RO (Repeated Over) and RP (Replaces a Prior course) to the original and retaken courses respectively, and updates the cumulative statistics to show earned credit for only one course. The system averages the grades earned for both courses into the grade point average.

No interchanging categories of repeat logic

Determine the category of repeat logic in the crs_rec. If the column crs_rec.rep="Y," the system applies only C category logic; if crs_rec.rep="N", only R logic applies. The system does not interchange categories of repeat logic.

Example: The crs_rec.rep="Y" for a course, and the student fails the course. The system counts the failure as one of the times the course can be repeated for credit and applies C category logic. Even if policy permits the student to retake the course to improve the grade, no current means exist to tell the system to ignore the failed course or to apply different repeat logic.

System-assigned repeat codes in cw_rec(s) could be corrupt

Review the cw_rec(s) to determine the repeat codes that have been assigned by the system or the institution. If you assigned a manual code to the cw_rec, the computations occur exactly as specified by the institution in the rep_table.

Proper setup of the following grd_table columns: instr_assgn, val, and rep

Review the grd_table with special attention to the columns instr_assgn, val, and rep. Ensure that the values are set according to those described in the previous pages describing the process for setting up repeat logic.

Troubleshooting Process - Repeat Logic

The following lists and describes the additional tests to perform when repeat logic is not performing as expected:

1. Check the `crs_rec.rep` value to ensure that the repeat code is not assigned. It is possible for the repeat code to be assigned, but not printed on the transcript (i.e., if the `rep_table.print` value is zero).
2. Check the `RESREP` definition in the directory path `$CARSPATH/include/custom/trans` if any of the involved courses are transfer work. If the include is undefined, the system ignores all transfer work for repeat logic.
3. Check the value of each of the grades (`grd_table.val`). The system ignores all grades with a value of zero for repeat logic.
4. Check each of the repeat codes. The system ignores all courses with manual repeat codes (`rep_table.ctgry = "M"`) for repeat logic.
5. Check the Repeat table to make sure that valid codes are entered, including the four categories of records: C, R, S, and M. The Repeat table must have one or more entries (in the C, R, and S categories) that match the restrictions for the repeated course.

For each category of a Repeat record, enter a record containing the minimal restrictions (i.e., no minimum hours and blanks). Otherwise, all possible combinations for the non-blank restrictions should be in the `rep_table`.

Note: Entries to the Repeat table columns are described below:

- `rep_table.min_hrs`
- If the course's value is higher than the student's hours, the system ignores the record.

CAUTION: In the base `rep_table`, all values in the `min_hrs` column equal zero.

Institutions rarely set the `min_hrs` value greater than zero; this value can be used to apply different repeat logic depending on the `min_hrs`. The `CUM_BY_LAST` include specifies the `min_hrs` used for repeat logic, either the cums at the time of the repeat attempt or the cums before the repeat attempt.

- `rep_table.grd`
If the value is not blank and does not match the `grd_table.rep` for the course, the system ignores the record.
- `rep_table.over_max`
If the value is "Y" and the repeat is not an over-maximum situation, or if the value is "N" and the repeat is an over-maximum situation, the system ignores the record. An over-maximum situation occurs when the number of repeats exceeds the limit for the student's entire transcript defined by the `MAXREP` include in the `$CARSPATH/include/custom/trans` file, or if it exceeds the limit in `crs_rec.max_rep` or `crs_rec.max_rep_hrs`.

Includes - Repeat Logic

This section contains only those includes that pertain to repeat logic. The includes are located in the directory path `$CARSPATH/include/custom/trans`.

RESREP

If you define this include, repeat logic applies only to resident courses (i.e., those offered by the institution). If undefined, repeat logic applies to both resident and nonresident (transfer) courses.

REP_BY_GRD

If you define this include, repeat logic applies based on the highest grade a student receives for a course. If this include is undefined, repeat logic assigns codes based on the last attempt of the course.

Example: A student takes MAT101 in FA01 and gets a "D" grade and retakes the course next spring, but receives an "F." If you define this include, the repeat logic

retains the "D" and removes the "F" from grade point averages. If the include is undefined, the "F" counts in the GPA, and the "D" is removed. The system makes adjustments in the session in which the course was retaken; the original term remains unchanged. (The repeat logic applied is complex, including the values set in the crs_rec: repeatable Y/N, and number of times/maximum hours repeatable.)

CUM_BY_LAST

If you define this include, repeat logic uses the cumulative hours for the last session a course is repeated to determine when to compute rep_min_hours from the rep_table.

Example: If you define the include and rep_table.min_hrs to equal 60, and the student had 50 cumulative hours when the course was taken the first time, and 65 cumulative hours when the course was repeated, the system uses the 65 hours and applies the repeat code as appropriate to the table entry in the rep_table.min_hrs=60. If the include is undefined, the table entry does not apply for repeat logic, because the student did not have 60 cumulative hours when the course first was taken.

MAXREP

If you define this include, and you specify a number other than zero, you set the total number of times the system applies repeat logic to improve grades for a given program.

Note: The setting represents total attempts, not specific courses. If there are limits that restrict the number of times a course can be retaken to improve a grade, make the setting in the crs_rec.rep (repeatable N, number of times repeatable equals some non-zero value). If MAXREP equals zero, you set no limit to the number of times the repeat logic can be used to improve a grade. When set to "4", the student can use the repeat logic to improve grades only four times; any attempts above four cause the grades to be computed in the GPA, but the student earns no additional hours. The repeat code that the system assigns corresponds to the rep_table.over_max="Y" (or blank).

Course Record - Repeat Logic

The following three columns of the Course record (crs_rec) apply to repeat logic:

- Repeatable
- Times Repeatable
- Repeat Hours

The Course Catalog screen is located in \$CARSPATH/modules/regist/progscr/crsent/course

Repeatable column

Repeatable refers to the crs_rec.rep column. The valid values are "Y" or "N." The column is the basis for much of the repeat logic.

- If you set crs_rec.rep to "Y," the course can be retaken multiple times for credit. The number of times or hours depends upon the values in the crs_rec.max_rep and crs_rec.max_rep_hrs columns. A "Y" entry relates to the rep_table.ctgry column whose value is Repeatable (C). In this case, only a rep_table.rep value of CU or CO applies, depending on the values in the crs_rec.max_rep and crs_rec.max_rep_hrs columns. Repeat logic never applies the substitute (S) category (rep_table.ctgry=Y) to courses that have a "Y" in the Repeatable column. If you set crs_rec.rep to blank, the system views the entry as "N."
- If you set crs_rec.rep to equal "N," the student cannot retake the course for credit, and the student retakes the course to improve the grade. The crs_rec.max_rep and crs_rec.max_rep_hrs column values control the number of times or hours the course can be repeated.
- The crs_rec.rep "N" code relates to the rep_table.ctgry column set to Repeated (R). When you set the columns accordingly, a RU or RO rep_table.rep value applies to the original or

repeated course. When defined, the REP_BY_GRD include assigns RU to the cw_rec of the lower of the two grades and assigns RP (corresponds to the rep_table.ctgry substitute (S) value) to the rep_table.rep column for the replacement grade. When the REP_BY_GRD include is undefined, repeat logic applies RU to the original course and RP to the repeated course.

- If the student earns a grade (grd_table.rep="N") for the course and retakes the course, the cw_rec reflects an RO and RP (cw_rec.rep) on the two repeats. The student earns credit for one repeat, but both grades are averaged into the GPA.

Times Repeatable column

Times Repeatable refers to the column crs_rec.max_rep.

- If you set the Repeatable column to "Y," the Times Repeatable column entry equals the number of times a student can repeat the course for credit. Entering zero equals unlimited repeats while entering a non-zero value sets the specific number of repeats. If the Course counts within the range specified in the crs_rec.max_rpe column, the system does not normally assign repeat logic to a course the first time it is taken. The system assigns all subsequent repeats the rep_table.rep value of CU in the cw_rec.rep, if within the range, and CO if the course is retaken more than twice.
- If you set the Repeatable column to "N," the Times Repeatable column entry equals the number of times the course can be retaken to improve a grade. Entering a zero equals unlimited repeats, while entering a non-zero value sets the specific number of repeats. The value repeat logic applies (RU or RO) to the cw_rec.rep column for the course depends on the following:
 - When the grade earned the first time is grd_table.rep="Y" (the grade can be repeated) and is within the range specified in the Times Repeatable column, repeat logic applies the appropriate RU/RP codes to the two courses. The specific code depends on the definition of the REP_BY_GRD include.
 - When the grade earned is grd_table.rep="N" and the course is repeated, repeat logic assigns cw_rec.rep values of RO and RP to the last two repeats. The student earns credit only once, and both grades are averaged into the grade point average.
 - When the grade earned the first time is grd_table.rep="Y" (the grade can be repeated) and the value in the Times Repeatable column is exceeded, repeat logic applies RO/RP codes to the last two course repeats in the cw_rec.rep columns. The student earns credit only once, and both grades are averaged into the grade point average.

Repeat Hours column

Repeat Hours refers to the crs_rec.max_rep_hrs column. The repeat logic for the Repeat Hours column is the same as that of the Times Repeatable column, except that the logic for Repeat Hours applies to hours instead of courses.

Times Repeatable and Repeat Hours

Repeat logic uses the Times Repeatable and Repeat Hours columns in combination with each other.

- If the Times Repeatable column is non-zero and the Repeat Hours column is zero, the Times Repeatable logic applies.
- If the Times Repeatable column is zero and the Repeat Hours column is non-zero, the Repeat Hours logic applies.
- If the Times Repeatable and Repeat Hours columns are both non-zero, the system assigns repeat logic based on the limit reached first.

Example: If Times Repeatable equals two and Repeat Hours equals twelve, and the student takes the course a third time, but the course does not exceed the Repeat Hours value, the repeat logic is based on Times Repeatable.

If Repeatable equals "N" and Times Repeatable and Repeat Hours both equal zero, the student can repeat the course indefinitely to improve a grade. The cw_rec.rep value on the original course is RU, and the last course taken is RP since the last grade counts (when the REP_BY_GRD include is undefined).

Repeat logic applies RP only when the grade earned on the first repeat contains an "N" in the grd_table.rep column and exceeds the MAXREP include. When you set the include to four, repeat logic applies to a program (e.g., UNDG) a maximum of four times. RO/RP logic applies when course repeats to improve a grade exceed four.

Notes:

- Repeat logic makes all adjustments to the cumulative statistics in the session in which the student repeats the course.
- The preceding repeat logic applies equally to those courses established as equivalents in the crseq_rec and/or course number changes as defined in the crshist_rec.
- Repeat logic does not apply any for transfer courses cw_rec equivalents when you define the RESREP include. In this case, the repeat logic applies only to resident courses, and excludes courses with a cw_rec.stat="T".

Grade Table - Repeat Logic

The following three columns of the Grade table (grd_table) apply to repeat logic:

- Instr Assgn
- Val
- Rep

The Grade Table report is located in \$CARSPATH/modules/regist/reports/tgrd.

Instr Assgn column

The Instr Assgn column (Instructor Assign) on the Grade Table report refers to the grd_table.instr_assgn column. The values for the Instr Assgn column are "Y" and "N." Set all values to "Y," except for the two system-assigned grades for In Progress (IP) and Not Reported (NR).

Val column

The Val column (Value) refers to the grd_table.val column. Its numeric values are based on the value of the grade, where the higher the value is, the higher the grade. Repeat logic uses the column's values to determine which is the higher grade, and when the REP_BY_GRD include is defined, uses the values where the highest grade counts when a student repeats a course to improve a grade. In general, all comparable grades should have similar, if not identical, values.

Example: All grades comparable to "A" have the same range of values, and the Val column values are all greater than the values for "A," for example.

CAUTION: Before the system can apply repeat logic to the Val column, you must set the column to a non-zero value.

Rep column

The Rep (Repeat) column refers to the column grd_table.rep. The values for the Rep column are "Y" and "N."

- If you set the grd_table.rep column to "Y," when the student earns a grade in a course and the student repeats the course, the grade is a candidate for repeat logic to apply the proper codes in an attempt to improve the grade. The following conditions apply:

- The `grd_table.val` column must be non-zero. The higher the `grd_table.val`, the higher the grade; therefore, failing grades should have the lowest value, and the "A" (or comparable) grades should have the highest value.
- A corresponding entry in `rep_table.grd="Y"` must exist. The `grd_table.rep` in the standard product is set to RU for this requirement. In this case, `grd_table.rep` points to `rep_table.grd` to determine which repeat logic the system applies.
- The system might not apply RU when the limits for retaking a course (`crs_rec.rep="N"`) are non-zero values in `crs_rec.max_rep`, and/or the `crs_rec.max_rep_hrs` are exceeded. In this case, the `rep_table.rep` value RO applies. The RO value also can apply when the value of repeated allowed in the MAXREP include is exceeded.
- If you set the `grd_table.rep` column to "N," the grade is not a candidate to improve the grade. The value does not correspond to the RU `rep_table.rep` value since the `rep_table.grd` equals "Y." Therefore, when `rep_table.grd` is blank (meaning that either "Y" or "N" from the `grd_table.rep` applies), the system applies the RO `rep_table.rep` repeat code.

Notes:

- The `grd_table.rep` values specify whether the grade can be repeated in an effort to improve the grade; therefore, it applies only to the `rep_table.ctgry` of Repeated (R). However, if the `crs_rec.rep="Y"` and the Repeatable (C) logic overrides, the `grd_table` applies first. Accordingly, a corresponding substitute (S) category (`rep_table.ctgry="S"`) always applies either to the original course or to the repeated course. If you define the REP_BY_GRD include, the higher of the two grades has the substitute logic applied (RP in the standard product) and the lower grade has the repeat logic, probably RU.
- The system makes all adjustments to cumulative statistics in the session of the repeat.
- When the system applies RO/RP logic to two courses, repeat logic allows earned credit for only one course, but both grades are averaged into the grade point average.

Repeat Table - Repeat Logic

The Repeat table (`rep_table`) is the most important entity in the repeat logic process. All other associated entities point to the `rep_table` to determine which repeat codes to assign to repeated courses. Repeat codes are stored in the `cw_rec.rep` column by the update function of Transcript, except for any manually assigned codes (`rep_table.rep="M"`). Manual codes are applied directly to the `cw_rec.rep`.

The following is a standard Repeat Table report and should be used as a basis for comparison to the institution's Repeat table.

```

pg8862.out
File Edit

Fri Oct 30 1998          CARS College          Page 1
09:23                   REPEAT TABLE REPORT          trep

          --- Factors ---          Audit
          Min Ovr Rp Session      Cum      I H
Cd Text      T Hours Max Gr A E P Q  A E P Q P  c s  Act/Inact
-----
*CO Repeated too many times C 0.00 Y   1 0 0 1 1 0 0 1 1 1 1 01/01/93
CU Repeatable      C 0.00 N   1 1 1 1 1 1 1 1 1 1 1 01/01/96
MP Replaces A Prior Course M 0.00   1 1 1 1 1 0 0 1 1 1 1
MU Repeated      M 0.00 N   1 1 1 1 1 1 1 1 1 1 1 1
RO Repeated      R 0.00   1 0 0 1 1 0 0 1 1 1 0 0
RU Repeated      R 0.00 N   Y 1 0 0 0 0 0 0 0 1 0 0
RP Replaces a Prior Course S 0.00   1 1 1 1 1 1 1 1 1 1 1 1

```

CAUTION: Because repeat logic is a very complex issue, Jenzabar, Inc. discourages an institution from making changes to the Repeat table without consulting with Jenzabar, Inc. first. See appropriate includes, crs_rec, cw_rec, and the grd_table, for additional information on the Repeat Table report.

Repeat Table Fields

The following lists alphabetically and describes the fields in the Repeat table.

Note: The Transcript program automatically assigns repeat codes according to flags in the Repeat (rep_table) and Grade tables (grd_table). You can set up the Transcript program to assign the repeat code for replacement to the course with the better grade or to the course taken most recently, despite the grade. Session and cumulative statistics will not change until the session where the repeat code takes effect. As a result, you must establish the Repeat and Grade tables prior to the installation of the Registration or Transcript programs.

Attempt (Sess/Cum)

Either 1 (Yes) or 0 (No) indicating whether to include the course in hours attempted.

Note: The system uses numeric values because it performs arithmetic operations with corresponding field values in the grd_table and rep_table (if applicable) to determine whether the course counts in hours attempted. The arithmetic operations also apply to hours earned, passed, and quality.

Audit Hrs

Either 1 (Yes) or 0 (No) indicating whether the course hours should be included for degree audit processing.

Code

The repeat code for counting repeating courses (e.g., CU, CO, RU, RO, RH, RP).

Earn (Sess/Cum)

Either 1 (Yes) or 0 (No) indicating whether to include the course for earned credit.

Note: Earned credit equals the sum of pass hours and quality hours, except when a student fails courses, which increases quality, but does not count as earned credit.

Grade

Either Y (Yes) or N (No) indicating whether the value appearing in the Code field is for a course with a repeatable grade.

Include Audit

Either 1 (Yes) or 0 (No) indicating whether the course should be included for degree audit processing.

Minimum Hours

The minimum number of cumulative earned hours before the repeat code is valid.

Note: This field applies only to institutions that apply different repeat codes depending on the number of cumulative earned hours for the student.

Over Maximum

Either Y (Yes) or N (No) indicating whether to use the code for a repeat when a course has been taken more times than the value contained in `crs_max_rep`. A value of "Y" also indicates that the code can be used when a student repeats a course that takes the student over the total maximum repeats allowed for a transcript.

Pass (Sess/Cum)

Either 1 (Yes) or 0 (No) indicating whether to include course hours in session passed hours.

Note: Pass should include all hours that are earned but do not affect the grade point average. Included in pass hours are pass/fail course work, transfer (unless it counts in the GPA), advanced placement, CLEP, and military.

Print

Either 1 (Print) or 0 (No) indicating whether the text associated with the repeat code (earned or not earned) is to print on the transcript or grade report.

Quality (Sess/Cum)

Either 1 (Yes) or 0 (No) indicating whether to include course hours in session quality hours.

(Text for Code)

The description for the value in the Code field.

Note: This unlabeled 24-character field appears to the right of the Code field.

Type

The code for the type of category, which signifies how the repeat code is to be applied.

Valid values are:

- C (Repeatable courses, i.e., courses that can be repeated for credit - starts with `crs_rec.rep=Y`)

Note: This code is applied to courses that were taken more than their allowed number of times. This is determined by the fields `crs_rep` (which must be "Y") and `crs_max_rep`.

Example: If the value in the Course record for "`crs_max_rep`" is "5", the sixth time a student takes the course, the proper type "C" repeat code will be applied.

- M (Manually assigned repeat logic)

Note: This code indicates that the repeat code is maintained manually. The repeat logic in *trans* ignores any course with a manual repeat code. The manual

repeat codes are used whenever there are special exception repeats the current transcript repeat logic cannot handle.

- R (Repeated in an attempt to improve a grade, i.e., starts with crs_rec.rep=N)
Note: This code indicates that the repeat code is to be applied to a course once the course has been replaced.
- S (Substitute, or the replacement code, for repeated [R])
Note: This code indicates that the repeat code is for courses that are substitutes, or replacement courses.

Repeat Table Factors in Applying Repeat Logic

You must consider many factors to apply repeat logic properly. The following lists and describes some of the more important factors you must consider when completing fields in the Repeat table:

Code

Refers to the column rep_table.rep, sometimes referred to as repeat flags.

Note: For repeated courses, the update function of Transcript determines one of the values in this column, and assigns it to cw_rec.rep. You assign manual repeat codes directly to the cw_rec.rep column.

(Text for Code)

Refers to the rep_table.txt.

Note: This text prints on the transcript under the appropriate course if Print=1 (rep_table.pnt). Generally, all text prints except for CU courses (i.e., courses that are repeated for credit).

Print

Refers to the rep_table.pnt described in the (Text for Code) factor.

Type

Refers to the rep_table.ctgry. The types of categories are as follows:

- C - for repeatable courses (courses that can be repeated for credit-starts with crs_rec.rep="Y")
- M - for manually assigned repeat logic
- R - for repeated in an attempt to improve a grade (starts with crs_rec.rep="N")
- S - for substitute, or the replacement code, for repeated (R)

Grade

Refers to the rep_table.grd.

Notes:

- If you set the column to "Y", you also should set one or more grades in the grd_table.rep with a value of "Y."
- Repeat logic attempts to match courses that are repeated in an attempt to improve the grade. Otherwise, some other logic applies.
- This factor applies only to the application of repeated (R) and substituted (S) categories. If the repeat logic cannot validly apply the RU code, it applies the RO code. Typically this occurs when the number of repeats specified in the MAXREP include is exceeded, or the values in the crs_rec (crs_rec.rep="N," crs_rec.rep_max/rep_max_hrs non-zero) are exceeded.

Minimum Hours

Refers to the rep_table.min_hrs.

Note: Normally the column is set to zero.

Over Maximum

Refers to the rep_table.over_max.

Note: This factor is used when a course has been taken too many times for credit, repeated too many times to improve the grade (starts with crs_rec.rep, crs_rec.max_rep, crs_rec.max_rep_hrs values), or the number of repeats specified in the MAXREP include has been exceeded. The "Y" and "N" value in crs_rec.rep determines the code used.

Session/Cumulative counters

All counters in Session/Cumulative for attempt, earn, pass, and quality are essentially the same as the counters in the grd_table and the cntg_table.

Note: The system takes these counters in its final computing of the attempt, earn, pass, and quality values that appear on the transcript. The update process in Transcript computes the session and cumulative statistics and then adjusts the cumulative statistics where repeat logic is applied. Therefore, these counters may not be computed exactly as presented, since the statistics are supported by program codes to ensure that they present accurate totals.

Course Work Record - Repeat Logic

The cw_rec.rep column of the Course Work record (cw_rec) is pertinent to repeat logic. The cw_rec.rep column contains the appropriate Repeat table codes (rep_table.rep) as determined by the update process in Transcript, based on the values in \$CARSPATH/include/custom/trans, rep_table, crs_rec, and grd_table.

You assign manual repeat codes directly to the cw_rec.rep. The update process in Transcript never updates or changes these codes.

Transcript Program - Repeat Logic

The Transcript program applies repeat logic when you run the update process in Transcript. The following lists the four categories of repeat logic available, listed in a logical order as established by the update process in Transcript.

M

Manually-assigned codes.

Note: The update process in Transcript ignores and never updates the code.

C

Repeatable courses.

Notes:

- Type C codes are CU and CO.
- Repeatable implies that the course counts for credit each time it is taken.
- The system determines the code when the crs_rec.rep="Y" for a course.

R

Courses that are attempts to improve a grade.

Notes:

- Type R codes are RU and RO.
- The system determines the code when the crs_rec.rep="N" for a course.

S

Substitute course (code RP).

Notes:

- RP is the type S code (code in the CX standard product).
- The system uses the code to replace the previously taken course or the course with the highest grade as determined by the REP_BY_GRD include.
- The code is the companion category for the R type logic.

CAUTION: Use manual repeats as a last resort for the following reasons:

- With a manual repeat code, the system treats a course as if it did not exist, other than to apply the repeat logic as contained in the Repeat table.
- The system ignores a course with a manual repeat in the number of times a course may be taken (either repeatable or for improving the grade as specified on the crs_rec.rep).

Note: Jenzabar, Inc. recommends that you explore other alternatives to manual repeats, such as Counting table (cntg_table) entries.

Repeat Logic Test Cases

The following are representative test cases that explain why the system assigns the repeat logic as listed on the transcript.

Test Case 1 - Repeat Logic

The MAXREP include is set to four. The repeat logic should change when the repeated (R) category (rep_table.ctgry) is used the fifth time. Also, the RESREP include is defined, so repeat logic applies only to the resident course; the REP_BY_GRD include is undefined, so the repeat counts in the grade point average.

Following is an example of actual transcripts for test case 1.

```

----- Fall Term 200X -----
ART124  New Art History                3.00  F
        ** Repeated-under **

        attempt  earn    pass  quality  points  gpa
ses     3.00    0.00   0.00   3.00    0.00   0.000
cum     3.00    0.00   0.00   3.00    0.00   0.000

----- Spring Term 200Y -----
ART124  CLEP                            3.00  CL
        ** CLEP Testing **
        ** Repeated-over **

        attempt  earn    pass  quality  points  gpa
ses     3.00    3.00   3.00   0.00    0.00   0.000
cum     6.00    3.00   3.00   0.00    0.00   0.000

----- Fall Term 200X -----
ART124  New Art History                3.00  C
        ** Replaces A Prior Course **

        attempt  earn    pass  quality  points  gpa
ses     3.00    3.00   0.00   3.00    6.00   2.000
cum     9.00    3.00   0.00   3.00    6.00   2.000

----- Spring Term 200Y -----
ART123  Art History                    3.00  D
        ** Repeated-under **
BIB102  The Kings of Israel              3.00  C-
        ** Repeated-under **
BIO110  Biology - Level I                 3.00  D-
        ** Repeated-over **

        attempt  earn    pass  quality  points  gpa
ses     9.00    9.00   0.00   9.00   10.02   1.113
cum    18.00   12.00   0.00  12.00   16.02   1.335

----- First Summer Session 200Z -----
TAX330  Tax Accounting - CPA Prep        3.00  B+
        attempt  earn    pass  quality  points  gpa
ses     3.00    3.00   0.00   3.00    9.99   3.330
cum    21.00   15.00   0.00  15.00   26.01   1.734

----- Fall Term 200Y -----
ART123  Art History                    3.00  C+
        ** Replaces A Prior Course **
BIB102  The Kings of Israel              3.00  B-
        ** Replaces A Prior Course **
TAX330  Tax Accounting - CPA Prep        3.00  C+
        attempt  earn    pass  quality  points  gpa
ses     9.00    9.00   0.00   9.00   21.99   2.443
cum    30.00   18.00   0.00  18.00   39.99   2.222

----- Spring Term 200Z -----
BIO110  Biology - Level I                 3.00  C+
        ** Replaces A Prior Course **

        attempt  earn    pass  quality  points  gpa
ses     3.00    3.00   0.00   3.00    6.99   2.330
cum    33.00   18.00   0.00  21.00   46.98   2.237

----- First Summer Session 200Z -----
TAX330  Tax Accounting - CPA Prep        3.00  A
        ** Repeated too many times **

        attempt  earn    pass  quality  points  gpa
ses     3.00    3.00   0.00   3.00   12.00   4.000
cum    36.00   18.00   0.00  24.00   58.98   2.458

```

Notes:

- The fifth occurrence of repeat logic occurred on BIO110 in Spring 200Y, and the system appropriately applied RO (Repeated Over) even though the student received a grade that would allow the student to repeat the course to improve the grade (as defined in the MAXREP include).
- The course ART124 was set up to repeat only once (crs_rec.rep="N," crs_rec.max_rep="2"). When the course was repeated the third time, the text appeared, "Repeated - Over for code RO," meaning that the course was repeated too many times to improve a grade. The system applied the RO code to the second attempt, the RP replacement code to the third attempt, and averaged both of the grades in the grade point average.
- TAX330 does not apply to the MAXREP condition. It is a repeatable course (crs_rec.rep="Y," crs_rec.max_rep="2"), which means the course can be retaken two times for credit. Note that this occurs with no text since the rep_table.prnt flag is not set to print. However, note that when the course was attempted the third time in Spring 200Z, the message "Repeated too many times" appears. Code CO applies the message to indicate that no more earned credit will be granted, but the last two grades are averaged into the grade point average.

Test Case 2 – Repeat Logic

The includes as specified in test case 1 above apply, and HIST299 has the values crs_rec.rep="Y," crs_rec.max_rep="3," and crs_rec.max_rep_hrs="6." The course can be repeated three times for credit or a maximum of six credit hours.

Following is an example of an actual transcript for test case 2.

----- Spring Term 200X -----						
HIST299	Senior Honors Colloquium				3.00	C+
	attempt	earn	pass	quality	points	gpa
ses	3.00	3.00	0.00	3.00	6.99	2.330
cum	3.00	3.00	0.00	3.00	6.99	2.330
----- Fall Term 200X -----						
HIST299	Senior Honors Colloquium				3.00	B+
	attempt	earn	pass	quality	points	gpa
ses	3.00	3.00	0.00	3.00	9.99	3.330
cum	6.00	6.00	0.00	6.00	16.98	2.830
----- Fall Term 200Y -----						
HIST299	Senior Honors Colloquium				2.00	A
	** Repeated too many times **					
	attempt	earn	pass	quality	points	gpa
ses	2.00	2.00	0.00	2.00	8.00	4.000
cum	8.00	6.00	0.00	8.00	24.98	3.123

Note: The third attempt exceeded crs_rec.max_rep_hrs="6"; therefore, the system allowed no earned credit for the last repeat attempt (cum earn = 6.00), but the last repeat did count in the grade point average (cum qual = 8).

Test Case 3 – Repeat Logic

ART145 has the values in the crs_rec.rep="N," rs_rec.max_rep="0," crs_rec.max_rep_hrs="0." A student repeats the course three times.

Following is an example of an actual transcript for test case 3.

```

----- Spring Term 200X-----
ART145 Photography - Darkroom Principle 3.00 D-
      ** Repeated-under **

      attempt  earn  pass  quality  points  gpa
ses    3.00    3.00    0.00    3.00    2.01    0.670
cum    3.00    3.00    0.00    3.00    2.01    0.670

----- Fall Term 200X -----
ART145 Photography - Darkroom Principle 3.00 B
      ** Repeated-over **

      attempt  earn  pass  quality  points  gpa
ses    3.00    3.00    0.00    3.00    9.00    3.000
cum    6.00    3.00    0.00    3.00    9.00    3.000

----- Spring Term 200Y -----
ART145 Photography - Darkroom Principle 3.00 A
      ** Replaces A Prior Course **

      attempt  earn  pass  quality  points  gpa
ses    3.00    3.00    0.00    3.00    12.00   4.000
cum    9.00    3.00    0.00    6.00    21.00   3.500

```

Notes:

- The first grade earned was "D-" in which the grd_table.rep="Y." The second grade earned was "B", grd_table.rep="N," and the third attempt "A," grd_table.rep="N."
- In the second attempt, the "B" replaced the "D-." Notice in the second attempt that the system allowed no more earned credit or quality hours, but incremented the quality points by six to nine. The source code for the update process in Transcript accomplished the increment and is the only way quality points can be incremented without changing quality hours.
- On the third attempt, the system marked "B" as RO (repeated over) since its grd_table.rep="N" (meaning that the course cannot be repeated to improve the grade). The cum earned remained at three (no more earned credit), but the quality hours totaled six, and the points incremented by twelve (meaning the "A" and the "B" grades are both averaged into the grade point average).

Test Case 4 – Repeat Logic

A student takes BUS110 twice and earns grades of "B+" and "A-." Both of these grades have grd_table.rep="N," and the course has crs_rec.rep="N," crs_rec.max_rep="0," and max_rep_hrs="0."

Following is an example of an actual transcript for test case 4.

```

----- Fall Term 200X -----
BUS110 Microeconomics - Nonmajors 3.00 B+
      ** Repeated-over **

      attempt  earn  pass  quality  points  gpa
ses    3.00    3.00    0.00    3.00    9.99    3.330
cum    3.00    3.00    0.00    3.00    9.99    3.330

----- Spring Term 200Y -----
BUS110 Microeconomics - Nonmajors 3.00 A-
      ** Replaces A Prior Course **

      attempt  earn  pass  quality  points  gpa
ses    3.00    3.00    0.00    3.00    11.01   3.670
cum    6.00    3.00    0.00    6.00    21.00   3.500

```

Note: Since neither grade could be repeated to improve the grade, the system applies the RO code (repeated over) and assigns RP (replaces a prior course) to the second

attempt. Notice that the system allowed earned credit only once, but both grades counted in the grade point average.

Test Case 5 – Repeat Logic

A student takes ART 145 three times and earns the grades of "B-," "D," and "A" consecutively. The course cannot be repeated for credit and has no limits to the number of times the course can be repeated to improve a grade.

Following is an example of an actual transcript for test case 5.

----- Spring Term 200X -----						
ART145	Photography - Darkroom Principle				3.00	B-
	** Repeated-over **					
	attempt	earn	pass	quality	points	gpa
ses	3.00	3.00	0.00	3.00	8.01	2.670
cum	3.00	3.00	0.00	3.00	8.01	2.670
----- First Summer Session 200X -----						
ART145	Photography - Darkroom Principle				3.00	D
	** Repeated-under **					
	attempt	earn	pass	quality	points	gpa
ses	3.00	3.00	0.00	3.00	3.00	1.000
cum	6.00	3.00	0.00	6.00	11.01	1.835
----- Fall Term 200X -----						
ART145	Photography - Darkroom Principle				3.00	A
	** Replaces A Prior Course **					
	attempt	earn	pass	quality	points	gpa
ses	3.00	3.00	0.00	3.00	12.00	4.000
cum	9.00	3.00	0.00	6.00	20.01	3.335

Notes:

- The "B-" grade earned first can be repeated to improve a grade. Therefore, when the student took the course the second time and earned a "D," the system averaged both grades in the grade point average.
- However, since the "D" grade can be repeated to improve the grade, the third occurrence that earned an "A" replaced the "D" in the grade point average.
- The end result is the "B-" and "A" grades are averaged into the grade point average, but the student earned credit only once.

SECTION 11 – PROGRAM ERRORS AND CRASH RECOVERY

Overview

Introduction

This section provides the following:

- A list of fatal errors
- Crash recovery procedures

Note: Refer to *Using Transcript* for a list of the more common status, field error, and warning messages that can occur when menu users execute the programs in Transcript.

Fatal Errors

Fatal errors can occur in the following programs in Transcript:

Transcript (*trans*)

If a fatal error occurs, use the following procedure:

1. Reload the program. Erroneous or corrupt data might cause the fatal error.
2. Try a different student's data to see whether the error recurs. If the error does not recur, the data for the first student may be causing the error.
3. Create a transcript for the student to see whether errors occur in that process. Use the output from the transcript to identify the source of the erroneous data.

Other Issues

The following list identifies the most common problems you may encounter and the possible solutions for them.

The system does a core-dump when running *trans* on a student.

There are two issues to address:

- Check your Course records (*crs_rec*) to ensure there are no blank course titles. If there are some, correct them, and rerun *trans*.

Note: Before rerunning *trans*, you may want to check the courses that have blank titles with the courses on the record of the student who caused *trans* to core-dump so you can determine whether there are any matches.

- In the *prog_enr_rec*, there are two columns where an official and/or unofficial transcript form can be specified for a student (*prog_enr_rec.trans_frm_ofcl* and *prog_enr_rec.trans_frm_unofcl* respectively). These values must correspond to valid *transfrm_table* codes. If a valid code is entered to the *prog_enr_rec* and is later removed from the *transfrm_table* (or updated to some other code), *trans* will core-dump for this student unless the code in the *prog_enr_rec* was also changed after the *transfrm_table* was changed.

Fatal Errors

Messages Received

The following lists alphabetically the fatal error messages that can appear when you are using Transcript.

Note: To locate these errors, enter:

1. `cd src/regist/trans`
2. `grep error *.c | more`
3. `grep ERROR *.c | more`
4. `cd src/regist/trans`

Declare error 202 at line 145 in file work.ec

Error message returned from Revfile.c.

Stu_stat_rec ('UNDG' program). INSERT: status 236.

Error message returned from Revfile.c.

Error and Crash Recovery Procedures

Introduction

The procedures to recover from a crash are organized by the seriousness of the error.

Core Dump Recovery

The following procedure describes the steps to recover from a core dump of an entry program.

1. Access the program screens directory for the entry program.

Example: % cd/\$CARSPATH/modules/regist/stuentry/progscr

2. Reinstall each program screen file.

Example: % make reinstall F=<filename>

Note: You also can reinstall all of the screens by entering the following:

– % make reinstall F=all

3. Attempt to execute the entry program. Did the reinstall of the program screens fix the error?
 - If yes, you are done
 - If no, go to step 4

4. Access the source code directory of the entry program.

Example: % cd/\$CARSPATH/src/regist/stuentry

5. Reinstall the source code for the entry program.

Example: % make reinstall

6. Attempt to execute the entry program. Did the reinstall of the program source code fix the error?

- If yes, you are done
- If no, go to step 7

7. In the source code for the entry program, delete the old compiled code for the entry program.

Example: % make cleanup

8. Reinstall the entry program source code.

Example: % make reinstall

9. Attempt to execute the entry program. Did the deletion of the old code and reinstallation of the program source code fix the error?

- If yes, you are done
- If no, go to step 10

10. Review the libraries for the entry program. In the source code for the entry program, review the file, Makefile. In the file, search for the parameter, ADDLIBS, which identifies the libraries that you must reinstall.

Example: % vi Makefile

/ADDLIBS

11. Reinstall the libraries for the entry program, then reinstall the source for the entry program.

Example: % cd <to appropriate library>

```
% make reinstall
% cd/$carspath/regist/stuentry
% make reinstall
```

Note: You must reinstall the source program to include any library changes.

12. Attempt to execute the entry program. Did the reinstallation of the libraries for the entry program fix the error?
 - If yes, you are done
 - If no, call Jenzabar, Inc. Support Services

INDEX

A

- Academic Calendar record
 - in Transcript program, 94
- Academic record
 - in Transcript program, 94
- academic statuses
 - printing, 124
- accessing
 - PERFORM screen files, 72
 - schemas, 23
 - Transcript Exam table, 83
 - Transcript Form table, 83
 - Transcript includes, 31
 - Transcript macros, 28
 - Transcript program, screen and window files, 43
 - Transcript Selection table, 83
 - Transcript tables, 83
 - Transfer Entry program, screen and window files, 57
- Accomplishment panel, 43
- adapting transcript to institution policy, 96, 97
- Add Pre-Transfer Courses
 - menu option, 70
- Add Pre-Transfer Courses - Pre-Transfer Course Entry screen, 57
- Add Transfer Courses
 - menu option, 70
- adding
 - a new transcript form, 111
 - page numbers to the transcript, 113
- Aid By Enrollment Status List screen, 44

B

- Block Transfer panel, 43
- Block Transfer record, 25
 - in Transcript program, 94
- broken session, 101

C

- C programs
 - relationship to macros and includes, 27
- changing
 - default form, 112
 - system default form, 112
- checking
 - transcript format modifications, 112
- columns
 - table, 13
- commands
 - student registration, 45
- Comment panel, 44

- conventions, 3
- Course record
 - in Transcript program, 90
 - repeat logic columns, 134
- Course Work Evaluation
 - menu option, 67
- Course Work Evaluation Report, 74
- Course Work record
 - in Transcript record, 94
 - repeat logic columns, 141
- Create Official Trans.
 - menu option, 69
- Create Official Trans/Curr
 - menu option, 69
- Create Unoff Trans/Curr
 - menu option, 69
- Create Unofficial Trans.
 - menu option, 69
- Create/Edit Transcripts
 - menu option, 67
- Create/Edit Transcripts screen file, 45
- cross-functional issues, 76
- Csh scripts
 - transctc, 73

D

- data dictionary, 23
- data flow descriptions
 - Transcript program, 38
 - Transfer Add program, 48
 - Transfer Entry program, 54
- dec.h files, 33
- def.c files, 33–35
 - example, 34
- definition files. *See* def.c
- definitions
 - SQL tables, 13
 - tables, records, 23
- diagrams
 - entity relationship 1, 17
 - entity relationship 2, 18
 - entity relationship 3, 20
 - entity relationship 4, 22
 - entity relationship key, 17
 - transcript format, 97
 - Transcript product, 9
 - Transcript program, 38
 - Transfer Add program, 48
 - Transfer Entry program, 54
- differences
 - in product, 1
- Display Transcripts
 - menu option, 67
- dmls, 33

dmlts, 33
dmms, 33
documents, related, 2

E

Education record
 in Transcript program, 90
Enter Block Resident Work
 menu option, 67
entity relationship diagram
 key, 17
entity relationship diagram 1, 17
entity relationship diagram 2, 18
entity relationship diagram 3, 20
entity relationship diagram 4, 22

F

field descriptions
 Transcript tables and records, 23
fields
 Repeat table, 138
 Transcript Exam table (*transexam_table*), 83
 Transcript Form table (*transfrm_table*), 83
 Transcript Form table (*transfrm_table*), 105
 Transcript Selection table (*transsel_table*), 83
files
 dec.h, 33
 def.c, 33–35
 mac.h, 33, 35–37
 Program Enrollment Records, 45
form
 system default, 112
 testing new, 113
 transcript default, 112
Form Entry (*formet*) program, 2
Form Order table, 2
forms
 transcript, 111

G

glossary entries
 additional information, 2
Grade table
 repeat logic columns, 136
Grade Table report, 136
group sorting, 118

H

Hold Action Table report, 121
Hold Table report, 121

I

includes

accessing, 31
 relationship to macros and C programs, 27
 repeat logic, 133
including
 course work in progress, 116
item sorting, 117

L

listing
 student statistic database field names, 107
locating
 course equivalencies, 125
 other files containing transcript information,
 107
 student statistics information, 107

M

mac.h files, 33, 35–37
macros
 accessing, 28
 relationship to includes and C programs, 27
maintaining
 historical information on session records, 122
manual
 conventions, 3
 intended audience, 1
 purpose, 1
menu options
 Add Pre-Transfer Courses, 70
 Add Transfer Courses, 70
 Course Work Evaluation, 67
 Create Official Trans., 69
 Create Official Trans/Curr, 69
 Create Unoff Trans/Curr, 69
 Create Unofficial Trans., 69
 Create/Edit Transcripts, 67
 Display Transcripts, 67
 Enter Block Resident Work, 67
 Move Pre-Trans By Student, 71
 Move Pre-Transfer Courses, 70
 Operator Form Request, 68
 Print Student PreTrans Crs, 70
 Print Transcripts, 70
 Select Transcripts - All, 69
 Select Transcripts - One, 68
 Update All Student Stats, 68
 Update Select Student Stat, 68
 Update Student Stat By Crs, 68
menus
 Registrar
 Transcripts menu, 67
 Transfer Work Processing menu, 70
modifying
 transcript format, 106
 Transcript program screen, 108

Move Pre-Trans By Student
 menu option, 71
Move Pre-Transfer Courses
 menu option, 70

N

Noncatalog panel, 44

O

Office Permissions report, 120
Office Table report, 121
Operator Form Request
 menu option, 68
options
 transcript sorting, 117

P

panels

- Accomplishment, 43
- Block Transfer, 43
- Comment, 44
- Noncatalog, 44
- Resident, Transfer, and Waiver, 43
- Section Title, 44

parameters

- Transcript program, 40
- Transfer Add program, 50
- Transfer Entry program, 56

PERFORM screen files

- accessing, 72

PERFORM screens, 72

pointers

- form and sqlda, 33

preventing

- data from printing, 114

Print Student PreTrans Crs

- menu option, 70

Print Transcripts

- menu option, 70

printing

- academic statuses, 124
- blank lines, 115
- inconspicuous characters, 116

procedures

- adapting transcript to institution policy, 96, 97
- adding a new transcript form, 111
- adding page numbers to the transcript, 113
- changing default form, 112
- changing system default form, 112
- checking transcript format modifications, 112
- including course work in progress, 116
- listing student statistic database field names, 107
- locating course equivalencies, 125

- locating other files containing transcript information, 107
- locating student statistics information, 107
- maintaining historical information on session records, 122
- modifying transcript format, 106
- modifying Transcript program screen, 108
- preventing data from printing, 114
- printing blank lines, 115
- printing inconspicuous character, 116
- resolving transcript problems, 113
- setting up course equivalencies, 126
- setting up hold codes, 120
- setting up repeat logic, 130
- setting up the course history for equivalent courses, 127
- setting up transfer equivalency, 129
- testing new forms, 113
- updating macro file and menu options, 111

process flow

- Transcript product, 9

product differences, 1

Program Enrollment record

- in Transcript program, 94

Program Enrollment Records file, 45

programs

- Form Entry (*forment*), 2
- Transcript (*trans*), 37
- Transfer Add (*trnsadd*), 47
- Transfer Entry (*trnsent*), 53

R

records, 24–26

- Block Transfer, 25

- field descriptions, 23

- Transcript Comment, 25

references. See documents, related

Registrar

- Transcripts Menu, 67

- Transfer Work Processing menu, 70

related documents, 2

repeat logic

- Course record columns, 134
- Course Work record columns, 141
- Grade table columns, 136
- includes, 133
- Repeat table, 137
- Repeat table factors, 140
- setup process, 130
- test cases, 142
- testing process, 131
- Transcript program, 141
- troubleshooting issues, 132
- troubleshooting process, 132

Repeat table

- fields, 138
- repeat logic, 137
- repeat logic factors, 140
- Repeat Table report, 137
- reports
 - Course Work Evaluation Report, 74
 - Grade Table, 136
 - Hold Action Table, 121
 - Hold Table, 121
 - Office Permissions, 120
 - Office Table, 121
 - Repeat Table, 137
 - schema file reports, 24
 - Student Preliminary Transfer Course Report, 74
- Resident, Transfer, and Waiver panel, 43
- resolving
 - transcript problems, 113
- rows
 - table, 13
- runtime parameters. *See* parameters. *See* parameters.

S

- schema file reports, 24
- schemas, 23
 - naming, 23
 - reports, 24
- screen
 - Add Pre-Transfer Courses - Pre-Transfer Course Entry, 57
- screens
 - Aid By Enrollment Status List, 44
 - Create/Edit Transcripts, 45
 - PERFORM, 72
 - Student Academic Record, 72
 - Transcript, 43
 - Transcript Contact/Contact Table, 72
 - Transcript Form table, 37
- screens and windows
 - Transfer Entry program, 57
- Section record
 - in Transcript program, 90
- Section Title panel, 44
- Select Transcripts - All
 - menu option, 69
- Select Transcripts - One
 - menu option, 68
- session sorting, 117
- setting up
 - course equivalencies, 126
 - course history for equivalent courses, 127
 - hold codes, 120
 - repeat logic, 130
 - transcripts, 88

- transfer equivalency, 129
- setup process
 - repeat logic, 130
- size
 - of printed transcript, 106
- sorting
 - transcript data, 117
- SQL
 - table definition, 13
- Student Academic Record screen, 72
- Student Preliminary Transfer Course Report, 74
- student registration commands, 45
- syntax
 - schema names, 23
- system
 - default form, 112

T

- tables
 - columns, 13
 - field descriptions, 23
 - Form Order, 2
 - rows, 13
 - sequence in implementation, 82
 - Transcript Exam, 25
 - Transcript Form, 25
 - Transcript Selection, 26
- terminology
 - additional information, 2
- test cases
 - repeat logic, 142
- testing
 - new forms, 113
- testing process
 - repeat logic, 131
- tracking
 - course renumbering, 127
- transcript
 - default form, 112
 - forms, 111
 - group sorting, 118
 - item sorting, 117
 - session sorting, 117
 - size of printed., 106
 - sort keys, 117
- Transcript
 - setting options, 88
- Transcript Comment record, 25
- Transcript Contact/Contact Table screen, 72
- Transcript Exam table, 25
 - accessing, 83
- Transcript Exam table (transexam_table)
 - fields, 83
- Transcript Form table, 25
 - accessing, 83

- screens, 37
- Transcript Form table (*transfrm_table*)
 - fields, 83, 105
- transcript format
 - diagram, 97
- transcript options
 - avoiding blank space, 88
 - block transfers, 94
 - certificate table, 93
 - class rank, 94
 - clock hours, 94
 - clock hours only, 89
 - contact hours. *See* transcript hours: clock hours
 - detailed headers, 94
 - high school only, 90
 - manual repeat codes, 93
 - maximum blank space, 93
 - no signature lines, 94
 - one-sixth credit hours, 89
 - page length, 93
 - repeat by grade, 89
 - repeat by hours, 89
 - resident repeats, 89
 - short statistics, 93
 - slave printer setting, 90
 - third majors and minors, 94
 - total repeats, 89
 - transcript-added courses, 91
 - two decimal place GPAs, 93
- Transcript product
 - diagram, 9
 - process flow, 9
- Transcript program
 - accessing screen and window files, 43
 - data flow description, 38
 - diagram, 38
 - parameters, 40
 - repeat logic, 141
- Transcript program (*trans*), 37
- Transcript Selection (*transsel_table*) table
 - fields, 83
- Transcript Selection table, 26

- accessing, 83
- Transcript tables
 - accessing, 83
- Transcript tables and records
 - field descriptions, 23
- transctc
 - Csh script, 73
- Transfer Add (*trnsadd*) program, 47
- Transfer Add program
 - data flow description, 48
 - diagram, 48
 - parameters, 50
- Transfer Entry program
 - accessing screen and window files, 57
 - data flow description, 54
 - diagram, 54
 - parameters, 56
 - screens and windows, 57
- Transfer Entry program (*trnsent*), 53
- troubleshooting issues
 - repeat logic, 132
- troubleshooting process
 - repeat logic, 132

U

- unavailable features. *See* product differences
- UNIX
 - table names, 23
- Update All Student Stats
 - menu option, 68
- Update Select Student Stat
 - menu option, 68
- Update Student Stat By Crs
 - menu option, 68
- updating
 - macro file and menu options, 111

W

- windows
 - Transcript, 43