
Jenzabar CX

Recruiting and Admissions



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: tmrecadm

Distribution date: 06/23/2002

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.
RECRUITING AND ADMISSIONS 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	3
Jenzabar-Specific Terms	5
Keystrokes	6
SECTION 2 - RECRUITING/ADMISSIONS PROCESSES	7
Overview	7
Introduction	7
Purpose of Recruiting/Admissions	7
Background Knowledge	7
Process Flow	9
Diagram	9
Recruiting/Admissions Programs	10
General Process Descriptions	10
Related Jenzabar CX Applications	11
SECTION 3 - RECRUITING/ADMISSIONS 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	15
Recruiting/Admissions Tables and Records	15
Required Tables and Records	15
Recruiting/Admissions Schemas	17
Introduction	17
File Naming Conventions	17
Field Descriptions	17
Schema File Reports	17
Recruiting/Admissions Tables and Records	18
Introduction	18
SECTION 4 - RECRUITING/ADMISSIONS MACROS AND INCLUDES	27
Overview	27

Introduction	27
The Relationship Among Macros, Includes, and C Programs	27
General Installation Procedures	27
Recruiting/Admissions Macros	28
Introduction	28
Definition and Function	28
Macro Files and Locations	28
Applocate Script	29
Macros in the Admissions File.....	30
Introduction	30
Macros that Control Features	30
Recruiting/Admissions Document Control Macros (ADM_DOC_1 - ADM_DOC_12).....	41
Macros In The Leads File	43
Introduction	43
Macros in the Periodic File	45
Introduction	45
Macros	45
Includes	46
Introduction	46
Purpose	46
Macro Dependency	46
How to Locate Includes	46
Application Includes.....	47
Includes Used By Lead Entry	49
SECTION 5 – JENZABAR CX PROGRAM FILES	51
Overview	51
Introduction	51
Program File Details	51
Definition File	51
Example of a def.c File	52
mac.h Files	53
Example of a mac.h File	54
SECTION 6 - ADMISSIONS ENTRY.....	55
Overview.....	55
Introduction	55
Program Features Detailed	55
Process Flow	56
Diagram	56
Process Flow Description.....	56
Database Input	57
Database Output	58
Parameters	59
Introduction.....	59
Parameter Syntax.....	59
Parameters	59
Program Screens.....	61
Purpose	61
Access	61
Screen Files and Table/Record Usage	61
SECTION 7 - ADMISSIONS STATISTICS	65
Overview.....	65
Introduction	65
Program Features Detailed	65

Process Flow	66
Diagram	66
Process Flow Description	67
Features of admstats	69
Database Input	71
Database Output	72
Locating Status Errors from admstats	72
Program Relationships	72
Parameters	73
Introduction	73
Parameter Syntax	73
Parameters	73
Program Screens.....	75
Purpose	75
Access	75
Screen File and Table/Record Usage	75
SECTION 8 - ELECTRONIC APPLICATIONS	77
Overview	77
Introduction	77
Electronic Application Terminology: Acceptance	77
Electronic Application Terminology: Rejection	77
Program Features Detailed	77
Process Flow	78
Diagram	78
Process Flow Description for the Web Application Scripts	81
Scripts that Comprise the Web Application	81
Process Flow Description for the Electronic Applications Program	82
Program Relationships	83
Tables and Records Used in the Program	83
Program Parameters	85
Introduction	85
Parameters	85
Program Screens.....	87
Purpose	87
Access	87
Screen Files and Table/Record Usage	87
SECTION 9 - CALL ENTRY	89
Overview	89
Introduction	89
Program Features Detailed	89
Call Entry Process Flow	90
Diagram	90
Processing Flow Description	91
Program Relationships	91
Related Processes	91
Call Entry Parameters	93
Introduction	93
Parameter Syntax	93
Parameters	93
Program Screens.....	95
Purpose	95
Access	95
Screen Files and Table/Record Usage	95

SECTION 10 - EVENT ENTRY	97
Overview.....	97
Introduction.....	97
Program Features Detailed	97
Process Flow.....	98
Diagram	98
Overview.....	98
Introduction.....	98
Program Features Detailed	98
Process Flow.....	99
Diagram	99
Data Flow Description	99
Program Relationships	99
Tables and Records Used.....	99
Parameters.....	101
Introduction.....	101
Parameter Syntax.....	101
Parameters	101
Program Screens and Windows.....	103
Introduction.....	103
Access	103
Screen Files and Table/Record Usage	103
SECTION 11 - LEAD ENTRY	105
Overview.....	105
Introduction.....	105
Program Features Detailed	105
Lead Entry Process Flow	106
Diagram	106
Processing Flow Description	107
Program Relationships	107
Related Processes	107
Lead Entry Parameters	108
Introduction.....	108
Parameter Syntax.....	108
Parameters	108
Program Screens.....	109
Purpose	109
Access	109
Screen Files and Table/Record Usage	109
Other Lead Record Fields	110
Lead Record Fields Not Appearing on Lead Form.....	110
Lead Entry Parameters Screen	111
Purpose	111
Example.....	111
Parameters Passed.....	111
Lead Entry Program Features	113
Introduction.....	113
Data Consistency & Keystroke Saving Features	113
Data Tracking Features.....	113
Duplicate Checking Features	114
Lead Tickler Program	115
Parameter Syntax.....	115
Parameters	115
Lead Soundex Program	116

Parameter Syntax	116
SECTION 12 - MENUS, SCREENS, SCRIPTS, AND REPORTS	117
Overview	117
Introduction	117
Directory Locations	117
Communications Management Menu Options	118
Menu Structure	119
Introduction	119
Menu Options	120
Introduction	120
Programs, Screens, Scripts, and Reports	120
PERFORM (Table Maintenance) Screens	155
Introduction	155
PERFORM Screens	155
SQL Statements	161
Introduction	161
SQL Statements	161
Csh Scripts	166
Introduction	166
Csh Scripts	166
Reports	168
Introduction	168
ACE Reports for Letter Preparation	168
Printed ACE Reports in the Reports Directory	168
Printed Reports in the others Directory	178
Special Purpose Reports in the others Directory	179
ACE Reports for Letter Creation	181
Introduction	181
The ltradmit ACE Report	181
The ltrenclose ACE Report	182
The ltrintconf ACE Report	185
The ltrmistran ACE Report	186
The ltrparent ACE Report	187
The ltrpreapp ACE Report	189
The ltrschlabl ACE Report	190
The ltrincapp ACE Report	190
The ltrschvst ACE Report	191
SECTION 13 - CUSTOMIZING THE RECRUITING/ADMISSIONS PROCESSES	193
Overview	193
Introduction	193
Other Implementation Tasks	193
Basic Information	193
Assessing the Recruiting/Admissions Setup	194
Introduction	194
Before You Begin	194
Macro Files	194
Reinstalls After Macro Changes	194
Reviewing and Modifying Data in Tables and Records	196
Introduction	196
Procedure	196
Table and Record Information	196
Order of Table Information in This Section	196
Table to Set Up in Sequential Order	196
Tables to Set Up in Any Order	197

Building the Recruiting/Admissions Tables	199
Introduction	199
Access	199
Academic Status Table	199
Decision Table	199
Reference Table	200
Enrollment Status Table	200
Enrollment Sequence Table	202
Country Table	203
Lead Country Table	203
Denomination Table	204
Lead Denomination Table	204
Ethnic Table	204
Lead Ethnic Table	204
Major Table	205
Lead Major Table	205
Tickler Table	205
Program Table	205
Session Table	206
Academic Calendar Record	206
Communication Table	206
ADR Runcode Table	206
Alternate Address Table	207
Relationship Table	207
Addressing Record	207
Contact Table	207
County Table	208
State Table	208
Zip Code Table	208
Hold Table	208
Hold Action Table	209
Entry Selection/Sort Table	209
Entry Selection/Sort Criteria Table	209
Office Table	209
Office Permissions Table	209
Division Table	209
Department Table	209
Privacy Act Table	209
Privacy Act Field Table	209
Accomplishment Table	209
Citizen Table	209
Degree Table	209
Exam Table	209
Handicap Table	209
Interest Table	209
Involvement Table	209
Marital Table	210
Suffix Table	210
Title Table	210
Visa Table	210
Setting Up Multiple Admissions Programs	211
Introduction	211
Example Process	211
Step 1 - Creating New Tickler and Program Codes	211
Step 2 - Creating a New Program Macro (Optional)	212
Step 3 - Modifying the PROG_VALID Macro	212

Step 4 - Creating a New Tickler Macro and Modifying the TICK_VALID Macro.....	213
Step 5 - Modifying the ADM_TICK_VALID Macro.....	213
Step 6 - Creating a New Menu Option File.....	213
Step 7 - Creating a New Form Menu.....	214
Step 8 - Creating a New Inquiry Form for a New Program	215
Step 9 - Creating a New Application Form for a New Program	215
Step 10 - Adding a New Menu Option to the Menu.....	216
Step 11 - Enabling Letter Creation with the New ADMM Contact Tickler Code	216
Step 12 - Creating a New WPVI FileCabinet.....	217
Step 13 - Creating ACE Reports for Medical Admissions Letter Creation	218
Step 14 - Reinstalling Menuopt Files to Recognize the New Tickler and Program Codes.....	218
Creating School Records for High Schools from a Data File.....	219
Introduction.....	219
ETS and SAT Data File Formats.....	219
Verifying Records Before Conversion	220
Setting Macros in Preparation of Conversion.....	220
Menu Options to Create High School Records	220
Process Flow Description of the High School Data Conversion Script.....	220
Report Output from High School Data conversion.....	221
Creating School Records for Colleges from a Data File	222
Comparison of Processes for High School and College Data Files.....	222
Process Description.....	222
ID Records with an ID of Zero	222
Report Output	222
Setting Up the Cron Process.....	223
Introduction.....	223
Reviewing the admprocess Script.....	223
Reviewing the admprocess Crontab file.....	223
Interpreting and Modifying the admprocess File	224
Example Values for crontab Entries	225
Selecting Days for Processing	225
Setting cron to Run.....	225
Purpose of the crontab -l Command	226
Setting Directory and File Read/Write Permissions	226
Creating a crontab Job for a Menu User	227
Preventing a cron Process from Running	227
SECTION 14 - SETTING UP RECRUITING/ADMISSIONS LETTERS	229
Overview.....	229
Introduction.....	229
Information Resources	229
Creating Departmental Letters Using Sourcing.....	230
Introduction.....	230
Setting Up the Sourcing Feature	230
Creating Entries for the Department Table.....	230
Creating Entries for the Major Table	230
Creating an Entry for the Contact Table.....	231
Creating the General Departmental Letter	231
Sample DEPTLTR Letter.....	232
Creating Individual Departmental Letters.....	232
Sample LAS Departmental Letter.....	232
Processing Departmental Letters	232
Creating Departmental Letters Using MSWord or WordPerfect	233
Creating Interview Confirmation Letters.....	234
Introduction.....	234
Unique Aspects of Interview Confirmation Letters	234

Contact Table Setup	234
Interview Record Creation	235
Macro Creation	235
Sample Letters	235
SECTION 15 - ADMISSIONS IMPORT PROGRAM/ LOADING ACT AND SAT EXAM DATA FILES..	237
Overview	237
Introduction	237
Creating Exam Records From A Data File	238
Introduction	238
Converting Data from Diskette Format	238
Running the ACT Exam Data Report	238
Section 1 of the ACT Exam Data Report: Report of Matched Records	238
Example of Section 1	239
Section 2 of the ACT Exam Data Report: Report of Unmatched Records with SS Number	239
Example of Section 2	240
Section 3 of the ACT Exam Data Report: Report of Records without SS Number	240
Example of Section 3	240
Summary Report Information	241
Rerunning the ACT Exam Data report	241
Loading Conversion Data	241
Example of Output from tpconvert Process	241
Creating Contact Records for Students Added Using Data Conversion	242
Data Conversion Errors	243
Records Created from ACT Exam Data File	243
Overview	246
Introduction	246
Creating Exam Records From A Data File	247
Introduction	247
Converting Data from Diskette Format	247
Running the ACT Exam Data Report	247
Section 1 of the ACT Exam Data Report: Report of Matched Records	247
Example of Section 1	248
Section 2 of the ACT Exam Data Report: Report of Unmatched Records with SS Number	248
Example of Section 2	249
Section 3 of the ACT Exam Data Report: Report of Records without SS Number	249
Example of Section 3	249
Summary Report Information	250
Rerunning the ACT Exam Data report	250
Loading Conversion Data	250
Example of Output from tpconvert Process	250
Creating Contact Records for Students Added Using Data Conversion	251
Data Conversion Errors	252
Records Created from ACT Exam Data File	252
SECTION 16 - SETTING UP WEB APPLICATIONS.....	255
Overview	255
Introduction	255
Macros to Modify	255
Affected Tables	255
Web Display Flag Setup	255
Web Server Setup	256
Verifying the Implementation	256
SECTION 17 - RECRUITING/ADMISSIONS MAINTENANCE PROCEDURES	259
Overview	259

Introduction	259
Daily Maintenance	259
Session-Based Maintenance	259
Annual Maintenance	259
Other Maintenance	260
Completing Session Processing	261
Introduction	261
Tips for Creating Program Enrollment Records	261
Tips for Adding Document Contacts	261
Tips for Updating Final Statuses	262
Tips for Locating Students Not Registered	262
Locking and Unlocking the Admissions Record	263
Background	263
Unlocking Admissions Records	263
Setup and Maintenance Procedures for Lead Processing and Data Conversion	264
Introduction	264
Distinguishing Between Leads and Inquiries	264
Maintaining Leads in the Lead Record	264
Creating Lead Records Using Data Conversion	265
Creating Lead Records Using ACT or ETS Diskettes	266
Introduction	266
Copying Lead Records Using ACT or ETS Diskettes	266
Adapting the Script	266
Eliminating Local Revisions	267
Running the ETS SS Data Report and Resolving Data Inconsistencies	267
Sample ETS Student Search Data Report	268
Using the ETS SS Convert/Add Menu Option	269
Jenzabar Recommendations for Verifying Lead Records	269
Updating New Lead Records Using SQL	270
Using the Lead Records to Create Lead Contact Records	270
Moving Leads to Inquired	272
Records Created	272
Flagging Leads to Move to Inquiry	272
Results of the Move Leads Process	273
Using Lead Records to Track Both Undergraduate and Graduate Leads	273
SECTION 18 - PROGRAM ERRORS AND CRASH RECOVERY	275
Overview	275
Introduction	275
Fatal Errors	275
Errors from admstats	275
Errors in admentry	277
SQL Errors and Messages from elecapp	279
Jenzabar CX Screen Package Errors in elecapp	279
Error and Crash Recovery Procedures	281
Introduction	281
Core Dump Recovery	281
Recovery in admentry	282
Recovery in admstats	282
INDEX	283

SECTION 1 - USING THIS MANUAL

Overview

Purpose of This Manual

This manual provides technical information required to install, customize, and maintain CX's Recruiting/Admissions product.

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 Recruiting/Admissions 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 Recruiting/Admissions 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 Recruiting/Admissions 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 Recruiting/Admissions product. The organization of the manual is as follows:

Overview information

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

Product reference information

- Section 3 - Tables used in the product
- Section 4 - Macros and Includes
- Section 5 - CX program files
- Section 6 through 9 - Programs used in the product
- Section 10 - Menus, screens, scripts, and reports

Product procedures

- Section 11 through 12 - Procedures to install and customize your processes
- Section 13 - Procedures to load ACT and SAT tape information
- Section 14 - Procedures to set up electronic Web applications
- Section 15 - Procedures to maintain the product

Error reference/Recovery procedures

- Section 16 - 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 Recruiting/Admissions.

QuickMate online Help

QuickMate User Guide

Terminology

Master Glossary

UNIX-based Help

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

User guides

Recruiting and Admissions 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 designate a program name (e.g., *admentry*) within paragraphs
- 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.

Percent symbol

Indicates the standard UNIX prompt. Your prompt may vary.

Pound symbol

Indicates the standard UNIX superuser prompt identifying the user has additional permissions. Your prompt may vary.

Quotation marks

Represent information written in this manual exactly as it appears on the screen (e.g., 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 for exception situations only. 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 barcodes.



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-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 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 enables 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 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 - RECRUITING/ADMISSIONS PROCESSES

Overview

Introduction

This section provides information on the purpose and process flow of Recruiting/Admissions.

Purpose of Recruiting/Admissions

The primary purpose of Recruiting/Admissions is to enable an institution to add and update recruiting and admissions information for students who apply, who are being recruited, and who are admitted to your institution. Other records associated with students can also be entered and maintained, enrollment status can be tracked, and electronic applications from the World Wide Web (Web) can be viewed and updated. Telemarketing personnel can use the call entry program to track their recruiting efforts.

Background Knowledge

The following list describes the necessary background information that you should know to implement and support the Recruiting/Admissions 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

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

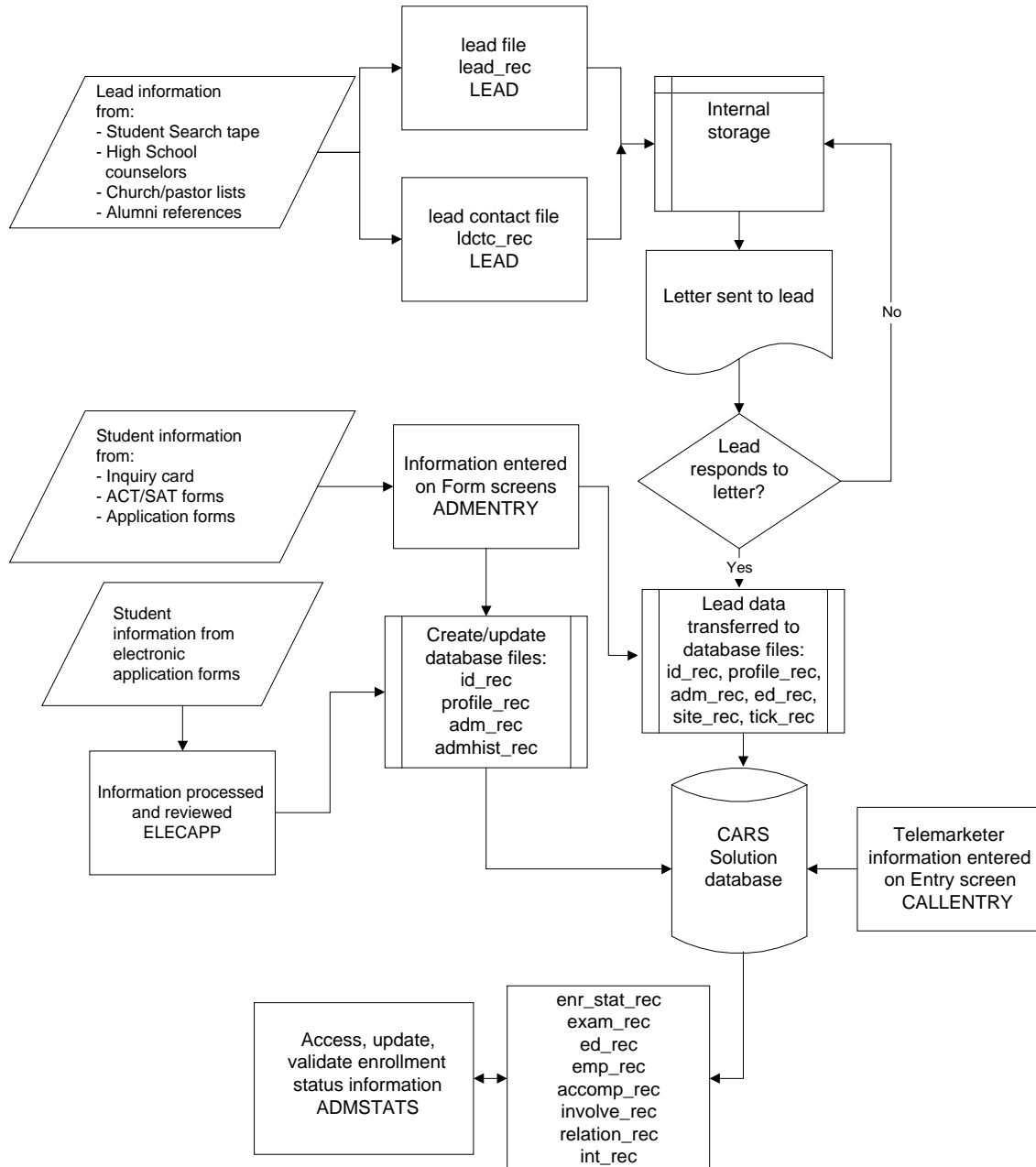
If you want to modify any CX programs to meet unique needs at your institution, you must know how to use the C programming language and have an in-depth knowledge of the CX code.

Process Flow

Diagram

The following diagram shows the process flow of the Recruiting/Admissions product. To help you locate the programs that perform each part of the process, program names appear in upper-case letters (e.g., ADMENTRY).

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



Recruiting/Admissions Programs

Recruiting/Admissions consists of the following programs:

- *admentry* (Admissions Entry)
- *admstats* (Admissions Statistics)
- *calentry* (Call Entry)
- *elecapp* (Electronic Application)
- *leadent* (Lead Entry)

In addition, *tpconvert* is a process that converts tape input from ACT or ETS into data that is recognized and used within CX. It is part of lead processing.

The Lead, Lead Contact, Contact, Enrollment Status, Exam, Education, Employment, Accomplishment, Involvement, Relation, and Interest records are optional, and can be used to store additional, biographical information about a student. The ID, Profile, and Admission records are required, and must be added to the system for a student, although not every field or part of those records must be populated with data.

General Process Descriptions

The following list contains general descriptions of each program in the Recruiting/Admissions product. For more information about each program, see the related program section in this manual.

Data conversion Description

The initial process in Recruiting/Admissions is to review Lead records to identify students who should be scheduled for a contact. The data conversion program can be used to create Lead records from information received on ACT or ETS electronic files or diskettes, or users can enter Lead record information directly into the Lead Entry screen. A lead is an individual who has not expressed interest in your institution; however, the institution wants to recruit him/her during the recruitment cycle. Once Lead records are created, the leads can be recruited, using Lead Contact records to send recruitment materials.

Admissions Entry Description

If a lead responds positively to a lead contact, information about that student is moved to the database, and the appropriate records (including an *id_rec*) are created or entered using the Admissions Entry form screens.

The *admentry* program lets users enter, view, and modify information about individuals on screens that can be modified to simulate the actual forms used by the admissions office, such as inquiry cards, ACT and SAT forms, and application forms. It also offers ID/Name lookup capability for parents, high schools and colleges, admissions counselors, and reference information.

Admissions Statistics Description

Once a student inquires for admission and the appropriate records are created, your institution can update a student's enrollment status and track his/her status changes. An Enrollment Status record (*enr_stat_rec*) is created each time an enrollment status changes for a student. This record is created and updated whenever a change in the enrollment status changes. You can use this information for a variety of reports. For example, you can use the Admissions Statistics record (*adm_stat_rec*) to create a tally or running total of how many students had a particular status for any given week during the year.

Electronic Application Description

Students can apply to your institution using the Web application form (if your institution purchases this optional feature). Applicants submit application information via the World Wide Web from any PC with Internet access. In the institution's admissions office, the *elecapp* program is then used to review these applications. If admissions office personnel want to accept the application for further consideration, they automatically create and/or

update ID, Profile, Admissions, Admission History, Alternate Address, Education, Interest, Involvement, Accomplishment, Site, Contact, and Enrollment Status records using the Electronic Application program. If the application is not considered worthy of further consideration (e.g., if it contains fictitious names or obviously incorrect addresses), admissions office personnel can reject it. Even if initially rejected, an application will reside in temporary records for later consideration if desired.

Call Entry Description

The Recruiting/Admissions product uses the *callentry* program to track the efforts made by call entry personnel who contact prospects for your institution. Callers can also enter the amount of time worked so reports can track productivity compared to time spent on the telephone.

Lead Entry Description

The Recruiting/Admissions product uses the *leadent* program to manage Lead records, Lead Tickler records, and Lead Contact records.

Related Jenzabar CX Applications

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

Financial Aid

Certain types of financial aid may depend upon a student's admissions status. The admissions office can also view the types and amounts of financial aid awarded by the financial aid office.

Registration

Certain pieces of data from the Admissions record can be copied into a Program Enrollment record once a student has confirmed his/her intention to attend your institution. This feature eliminates the need for the registrar to re-enter duplicate data. Once a Program Enrollment record is created for a student, there is the option to lock the student's Admission record. In addition, the admissions office has the ability to view a student's schedule of classes and to determine a prospective transfer student's progress toward a degree at your institution.

SECTION 3 - RECRUITING/ADMISSIONS TABLES AND RECORDS

Overview

Introduction

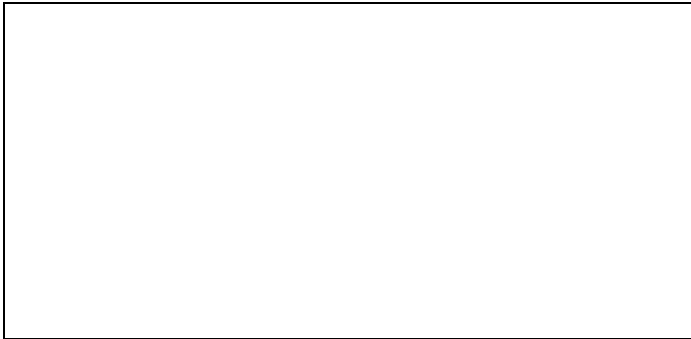
This section provides reference information about each table and record associated with the Recruiting/Admissions product. It also provides definitions of SQL and CX table and record terminology and differentiates among common, shared, and product-specific tables and records.

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, that is used 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.



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 from 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 can access records from CX program screens, detail windows, 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 Recruiting/Admissions 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

The following common schema files must be reviewed prior to beginning the data conversion process. Please note the common records, and discuss these with other departments at the institution as required. The common schema files are located in the \$CARSPATH/schema/common directory.

- Accomplishment (accomp_table)
- Address (adr_table)
- Alternate Address (aa_table)
- Classification (cl_table)
- Communication (comm_table)
- Contact (ctc_table)
- County (cty_table)
- Country (ctry_table)
- Degree (deg_table)
- Denomination (denom_table)
- Ethnic (ethnic_table)
- Exam (exam_table)
- Handicap (hand_table)
- Hold (hold_table)
- ID (id_rec)
- Interest (int_table)
- Involvement (invl_table)
- Office (ofc_table)
- Profile record (profile_rec)
- Relation (rel_table)
- Schedule record (schd_rec)
- State (st_table)
- Suffix (suffix_table)
- Tickler (tick_table)
- Title (title_table)
- User ID (userid_table)
- Visa (visa_table)
- Zip Code (zip_table)

Shared Tables and Records

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

- Academic Calendar record (acad_cal_rec)
 - Financial Aid
 - Registration
- Accomplishment record (accomp_rec)
 - Alumni/Development
 - Registration
- Admissions record (adm_rec)
 - Financial Aid
 - Registration
- Contact record (ctc_rec)
 - All products
- Church record (church_rec)
 - Registration
- Education record (ed_rec)
 - Financial Aid
 - Registration
 - Student Services
- Exam record (exam_rec)
 - Registration
- ID record (id_rec)
 - All products
- Interest record (int_rec)
 - Development
- Profile record (profile_rec)
 - All products
- Program Enrollment record (prog_enr_rec)
 - Financial Aid
 - Registration
 - Student Services
- Relation record (relation_rec)
 - All entry programs
- School record (sch_rec)
 - Financial Aid
 - Registration
 - Student Services
- Site record (site_rec)
 - All products

Recruiting/Admissions Tables and Records

Information about the tables and records that originate in the Recruiting/Admissions product appears later in this section.

Required Tables and Records

The following records are required to run the features of the Recruiting/Admissions product. Each admitted student must have entries in the following records:

- ID record (id_rec)
- Profile record (profile_rec)
- Site record (site_rec)

- Contact record (ctc_rec)
- Admission record (adm_rec)
 - Planned enrollment session (adm_rec.plan_enr_sess)
 - Planned enrollment year (adm_rec.plan_enr_yr)
 - Planned enrollment program (adm_rec.prog)

Each individual who is identified as a lead must have the following records:

- Lead record (lead_rec)

In addition, if your institution uses Tickler to schedule contacts, recipients must also have a Tickler record. For more information about using Tickler, see the CX document *Communications Management Technical Manual*.

Recruiting/Admissions 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 Recruiting/Admissions product in the \$CARSPATH/schema/admissions directory path.

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., *tdec* for the Decision table). For schema files containing definitions of CX records, the UNIX filename describes the English name of the record (e.g., *enrstat* for the Enrollment Status record).

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

Field Descriptions

Schema files contain descriptions of each field defined in a table or record. You can view descriptions of fields in Recruiting/Admissions tables and records by accessing the schema files. In addition, the comment line on each screen contains descriptive information about each field and the valid values it can contain.

Schema File Reports

The standard CX system includes three reports that provide information about the contents of the schema files. When table implementation begins, you can run these reports to provide the installation team with information about the tables and their fields.

Select the report options from the System Management: Data Dictionary menu.

The reports are as follows:

dbefield

Lists each column in the database by table, including its name, short and long descriptions, field type (e.g., char or date), and size.

dbefile

Lists the tables that relate to each track area (e.g., A = Admissions), including the table name, description, and purpose.

dbetrack

Combines the contents of *dbefield* and *dbefile*, displaying the tables for each product area and the columns in each table.

Recruiting/Admissions Tables and Records

Introduction

The following pages identify the tables and records that originate in the Recruiting/Admissions product. This information includes each table and record's purpose, association with programs, and association with other tables and records.

Tables and records described in this section are as follows:

- Admission History record
- Admission record
- Admissions Statistics record
- Advertising Medium table
- Advertisement record
- Application Documents table
- Call Entry Result table
- Call Entry Student record
- Call Entry Time record
- Call Results record
- Decision table
- Enrollment Sequence table
- Enrollment Status record
- Enrollment Status table
- Interview/Recommendation record
- Lead Contact record
- Lead Country table
- Lead Denomination table
- Lead Ethnic table
- Lead Major table
- Lead record
- Lead Tickler record
- Referral table
- Temporary Accomplishment record
- Temporary Application record
- Temporary Education record
- Temporary Interest record
- Temporary Involvement record
- Text Blob for Lead Contact Record

Admission History record

Maintains historical information from past Admission records. It is created automatically by the *admentry* program whenever the *adm_rec.plan_enr_sess* and/or the *adm_rec.plan_enr_yr* is changed.

UNIX filename: admhist

Informix filename: admhist_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: *admentry*, *elecapp*, *callentry*

Table/record interrelationships: adm_rec

Admission record

Maintains inquiry and application information about prospective students. Only one Admission record exists per student per program.

UNIX filename: adm

Informix filename: adm_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: admentry, admstats, elecapp, callentry

Table/record interrelationships: cl_table, dec_table, deg_table, id_rec, major_table, ref_table, sess_table

Admissions Statistics record

Maintains running totals of admission status changes, broken down into one week intervals. The *admstats* program creates and updates the Admissions Statistics records automatically. Information from the Admissions Statistics records is primarily used for reporting purposes.

UNIX filename: admstat

Informix filename: adm_stat_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: admstats

Table/record interrelationships: None

Advertising Medium table

UNIX filename: tadvmedium

Informix filename: adv_medium_table

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: None

Table/record interrelationships: None

Advertisement Record

UNIX filename: adv

Informix filename: adv_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: None

Table/record interrelationships: None

Application Documents

UNIX filename: reqadmdoc

Informix filename: reqadmdoc_table

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: None

Table/record interrelationships: None

Call Entry Result table

Identifies the valid Result codes for use in call entry.

UNIX filename: tcallres

Informix filename: tcallres_table

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: callentry

Table/record interrelationships: tcallres_rec

Call Entry Student record

Identifies call entry information about a prospective student.

UNIX filename: callstu

Informix filename: callstu_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: callentry

Table/record interrelationships: id_rec

Call Results record

Contains information about call entries.

UNIX filename: callresult

Informix filename: callresult_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: callentry

Table/record interrelationships: callresult_table, callstu_rec

Call Entry Time record

Maintains the amount of time worked in hours and minutes by callers for specified date ranges. This information is used in call entry statistical reports.

UNIX filename: call

Informix filename: call_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: None

Table/record interrelationships: None

Decision table

Contains all the valid admission Decision codes (e.g., FULL=Full Acceptance; PROB=Probationary Acceptance).

UNIX filename: tdec

Informix filename: dec_table

Schema location: \$CARSPATH/schema/admission

Program interrelationships: admentry

Table/record interrelationships: adm_rec

Enrollment Sequence table

Contains an entry for all valid and invalid enrollment status changes, defining which statuses may and may not be followed by another. The *admstats* program uses this table to check the logical progression of enrollment statuses.

UNIX filename: tenrseq

Informix filename: enr_seq_table

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: admstats

Enrollment Status record

Records all past and current enrollment statuses for each student. The *admstats* program automatically adds and deletes Enrollment Status records. Basic information includes the new enrollment status, previous enrollment status, beginning and added date of each status, and the program, session and year of application.

UNIX filename: enrstat

Informix filename: enr_stat_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: admentry, admstats, elecapp, callentry

Table/record interrelationships: adm_rec, adm_stat_rec, ctc_rec, ctc_table, enr_seq_table, enr_stat_table

Enrollment Status table

Defines all valid enrollment statuses used by the admissions office to track a student's progression through the recruiting/admissions cycle (e.g., INQUIRED, APPLIED, or ACCEPTED).

UNIX filename: tenrstat

Informix filename: enr_stat_table

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: admentry, admstats

Table/record interrelationships: adm_rec, adm_stat_rec, ctc_rec, ctc_table, enr_seq_table, enr_stat_rec

Interview/Recommendation record

Tracks information concerning an applicant's recommendation forms and interview evaluation forms. The tracked information includes whether or not the form has arrived, the date the form arrived, the name of the interviewer or recommender, and the evaluation score given the applicant by the interviewer or recommender.

UNIX filename: intvwrecom

Informix filename: intvwrecom_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: admentry

Table/record interrelationships: id_rec

Lead Contact record

Tracks all incoming and outgoing communications with a lead. These contacts are usually letters, but can also include documents, visits, phone calls, memos, or whatever an institution defines as a lead contact.

UNIX filename: ldctc

Informix filename: ldctc_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: None

Table/record interrelationships: ctc_table, lead_rec

Lead Country table

Contains the ETS country name and the institution's corresponding Country code. The *tpconvert* program uses this table to convert country names from an ETS tape to match the institution's Country codes.

UNIX filename: tleadctry

Informix filename: leadctry_table

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: *tpconvert*

Table/record interrelationships: ctry_table, lead_rec

Lead Denomination table

Stores the Denomination codes that are read from ACT and ETS tapes/diskettes, plus the institution's equivalent Denomination codes (as listed in the Denomination table). The *tpconvert* program uses this table to convert a Denomination code from an ACT or ETS tape/diskette into the institution's unique Denomination code which will then be used to populate a student's profile_rec.denom_code field.

UNIX filename: tleaddenom

Informix filename: leaddenom_table

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: *tpconvert*

Table/record interrelationships: denom_table, profile_rec

Lead Ethnic table

Stores the Ethnic codes that are read from ACT and ETS tapes/diskettes, plus the institution's equivalent Ethnic codes (as listed in the Ethnic table). The *tpconvert* program uses this table to convert an Ethnic code from an ACT or ETS tape/diskette into the institution's unique Ethnic code which will then be used to populate a student's lead_rec.ethnic_code field, or a student's profile_rec.ethnic_code field.

UNIX filename: tleadethnc

Informix filename: leadethnc_table

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: *tpconvert*

Table/record interrelationships: ethnic_table, lead_rec, profile_rec

Lead Major table

Stores the Major codes that are read from ACT and ETS tapes/diskettes, plus the institution's equivalent Major codes (as listed in the Major table). The *tpconvert* program uses this table to convert a Major code from an ACT or ETS tape/diskette into the

institution's unique Major code which will then be used to populate a student's lead_rec.major1 and lead_rec.major2 fields, or a student's adm_rec.major field.

UNIX filename: tleadmaj

Informix filename: leadmaj_table

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: tpconvert

Table/record interrelationships: adm_rec, lead_rec, major_table

Lead record

Stores information about possible recruits for your institution. This record is primarily used to load data from tapes/diskettes purchased from student search services (e.g., American College Testing or Educational Testing Services). Information from the Lead record can be copied into an ID, Profile, Admission, and Education record should a lead express interest in your institution.

UNIX filename: lead

Informix filename: lead_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: ldtickler, ldsndxinit, admentry

Table/record interrelationships: id_rec, ldctc_rec, ldtick_rec

Lead Tickler record

Contains information needed to place an individual on a specific tickler system and specifies the completion date.

UNIX filename: ldtick

Informix filename: ldtick_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: leadent, ldtickler

Table/record interrelationships: lead_rec, ldctc_rec

Referral table

Contains all valid Referral codes used by the recruiting/admissions office. Referral codes are used to track how a student initially heard of your institution (NEWS=Newspaper Advertisement), or how the institution acquired the student's name (e.g., ACTS=ACT Student Search, ETSS=ETS Student Search).

UNIX filename: tref

Informix filename: ref_table

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: admentry

Table/record interrelationships: adm_rec, lead_rec

Temporary Accomplishment record

Temporarily stores the details of an applicant's accomplishments that were received via the Web application. An accomplishment is an award or honor bestowed upon the student (e.g., All Conference athlete or Academic Honors).

UNIX filename: appaccomptmp

Informix filename: app_accomptmp_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: elecapp

Table/record interrelationships: accomp_table, accomp_rec, apptmp_rec, id_rec

Temporary Alternate Address record

Temporarily stores the details of an applicant's alternate address that were received via the Web application.

UNIX filename: appaatmp

Informix filename: app_aatmp_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: elecapp

Table/record interrelationships: aa_rec, aa_table, id_rec, apptmp_rec, ctry_table, st_table

Temporary Admissions record

Temporarily stores the details of a student's admissions information that was received via the Web application.

UNIX filename: appadmtmp

Informix filename: app_admtmp_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: elecapp

Table/record interrelationships: aa_rec, adm_rec, apptmp_rec, cl_table, deg_table, id_rec, major_table, profile_rec, prog_table

Temporary Application record

Temporarily stores the details of a student's application that was received via the Web application.

UNIX filename: apptmp

Informix filename: apptmp_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: elecapp

Table/record interrelationships: appaccomptmp_rec, appaatmp_rec, appadmtmp_rec, apptmp_rec, appidtmp_rec, appinttmp_rec, appinvtmp_rec, appproftmp_rec, appsitetmp_rec

Temporary Education record

Temporarily stores information about an applicant's attendance at other educational institutions (e.g., high schools or colleges), that were received via the Web application.

UNIX filename: appedtmp

Informix filename: app_edtmp_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: elecapp

Product interrelationships: apptmp_rec, ctry_table, deg_table, ed_rec, id_rec, sch_rec, sess_table, st_table

Temporary Identification record

Temporarily stores an ID record number for the applicant whose admissions information was received via the Web application.

UNIX filename: appidtmp

Informix filename: app_idtmp_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: elecapp

Product interrelationships: aa_rec, aa_table, apptmp_rec, ctry_table, id_rec, st_table

Temporary Interest record

Temporarily stores the details of an applicant's extracurricular interests that were received via the Web application.

UNIX filename: appinttmp

Informix filename: app_inttmp_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: elecapp

Product interrelationships: apptmp_rec, id_rec, int_rec, int_table

Temporary Involvement record

Temporarily stores the details of an applicant's extracurricular involvements that were received via the Web application.

UNIX filename: appinvltmp

Informix filename: app_invltmp_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: elecapp

Product interrelationships: apptmp_rec, id_rec, invl_table, involve_rec

Temporary Profile record

Temporarily stores the details of an applicant's profile information that were received via the Web application.

UNIX filename: appproftmp

Informix filename: app_proftmp_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: elecapp

Product interrelationships: apptemp_rec, citz_table, ctry_table, cty_table, ethnic_table, hand_table, id_rec, marital_table, profile_rec, st_table, visa_table

Temporary Site record

Temporarily stores a Site record for the applicant whose admissions information was received via the Web application.

UNIX filename: appsitetmp

Informix filename: app_sitetmp_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: elecapp

Product interrelationships: id_rec, site_rec, site_table

Text Blob for Lead Contact Record

Contains the TEXT field associated with a Lead Contact record.

UNIX filename: ldbctc

Informix filename: Informix filename: ldbctc_rec

Schema location: \$CARSPATH/schema/admissions

Program interrelationships: None

Table/record interrelationships: ldctc_rec

SECTION 4 - RECRUITING/ADMISSIONS MACROS AND INCLUDES

Overview

Introduction

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

The Relationship Among Macros, Includes, and C Programs

An m4 macro cannot be used directly in a C program since the system does not process C program code through the m4 processor. Therefore, CX uses includes so that a C program can communicate and process a macro. An include statement in an include file contains the information for defining a macro using syntax that a C program can interpret.

General Installation Procedures

See the *CX System Reference Technical Manual* for general procedures on setting and installing changes to macros and includes.

Recruiting/Admissions Macros

Introduction

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

Macro Files and Locations

The following macro files affect the Recruiting/Admissions product. The macro files are located in the \$CARSPATH/macros/custom directory path.

Additional information about the *admissions* and *periodic* macros follow on subsequent pages in this section.

admissions

Contains macros that define default values and enable some Recruiting/Admissions product features.

admrpt

The macros located in the macros/custom/admrpt macro file are used by the *profile* and *profone* reports located in the \$CARSPATH/modules/admit/others directory.

CAUTION: Contact Jenzabar before modifying these macros.

common

Contains macros used throughout CX. For more information about these macros, see *CX System Reference Technical Manual*.

configure

Contains macros to enable major features and functions within the Recruiting/Admissions product.

leads

Contains macros used for the Lead Entry program.

ltrwp

Contains macros that define two-character *nroff* (text formatter) macros. For more information about the *ltrwp* macros, see *Communications Management Technical Manual*.

matric

Contains macros that define the valid types of schools. The macros apply when you use data conversion scripts to create School records. The macros relevant to all CX institutions are:

- SCH_TYPE_HS
- SCH_TYPE_4YR_COL
- SCH_TYPE_2YR_COL

periodic

Contains macros that you change periodically for Recruiting/Admissions, including:

- Planned enrollment session and year
- Default high school graduation year
- Admissions Tickler code
- Planned lead enrollment session and year

table

Contains macros that define a variety of default, valid, and include values that relate to Registration, Financial Aid, and Recruiting/Admissions.

Applocate Script

You can also locate macros using *applocate*. The *applocate* script 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.

Follow these steps to run the *applocate* program.

1. Select Utilities 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., ADMISSION), and click **OK**.
6. Select **Finish**. The Output Parameters 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, *applocate.out*, and sends it to your home directory.

Note: Menu users can view the *applocate.out* file using the Display File to Screen menu option on the Utilities: File Options menu.

Macros in the Admissions File

Introduction

The following macros reside in the macros/custom/admissions file. They enable features, define default values, and provide contact information.

Macros that Control Features

The following macros enable features and define some default values used in the Recruiting/Admissions product, and appear in the order in which they are located in the \$CARSPATH/macros/custom/admissions file.

'ENABLE_FEAT_ADM_STUSERV', 'N'

Defines whether the admissions office has access to the student service PERFORM screen, and if the menu option for Student Services appears on the Recruiting/Admissions menu. If the admissions office is responsible for assigning residence hall rooms for incoming freshmen, and therefore must create the Student Service record, this macro should be set to Y. This macro is used by the \$CARSPATH/menusrc/admit/screens/menudesc file.

'ENABLE_FEAT_ADM_TICKLER', 'Y'

Defines whether the tickler program (the automatic scheduling of contact records) can be used by the admissions office. Setting this macro to Y will add the tickler submenu and tickler data fields to the Inquiry screen and to page 2 of the Application screen. This macro is used by the \$CARSPATH/menusrc/admit/commgmt/menudesc file, the \$CARSPATH/modules/admit/informers/moveleads file, and the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: inq_1 and longapp_2.

'ENABLE_FEAT_DISPLAY_FA', 'Y'

Defines whether the admissions office can view student financial aid information using a display-only version of the Financial Aid Entry program (*faentry*). Setting this macro to Y permits users to view a display-only version of *faentry* from the Recruiting/Admissions: Admissions Processing menu; otherwise, the option to view financial aid will not appear on the menu. This macro is used by the \$CARSPATH/menusrc/admit/screens/menudesc file.

'ENABLE_FEAT_DISPLAY_REGIST', 'Y'

Defines whether the admissions office can view student registration information using a display-only version of the Registration program (*regent*). Setting this macro to Y permits users to view a display-only version of *regent* from the Recruiting/Admissions: Admissions Processing menu; otherwise, the option to view registrations will not appear on the menu. This macro is used by the \$CARSPATH/menusrc/admit/screens/menudesc file.

'ENABLE_FEAT_LEAD_PROCESSING', 'Y'

Defines whether the admissions office can use the lead processing functions. Setting this macro to Y allows access to the Lead Processing submenu. This macro is used by the \$CARSPATH/menusrc/admit/menudesc file.

'ENABLE_FEAT_CHURCH', 'N'

Defines whether church information can be captured and tracked. Setting this macro to N removes the church_id field from all data entry screens, and removes menu options for reports that are related to church information from the menu. This macro is used by the \$CARSPATH/menusrc/admit/query/menudesc file and the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: admmenu, gadmmenu, inq_1, ginq_1, longapp_2, and glongapp_2.

'ENABLE_FEAT_ADD_DEF_INQ_CTC', 'Y'

Defines whether the Admissions Entry program (*admentry*) should add the initial Contact record by default through the use of the Inquiry screen. If set to Y, the Inquiry screen will

add the initial Contact record defined by the INQUIRY_DEFAULT_CTC macro. The Contact record must be in the Contact table before it is used by *admentry*. This macro is used by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: inq_1 and ginq_1.

'INQUIRY_DEFAULT_CTC', 'INQUIRED'

Defines the name of the default inquiry Contact record that will be added automatically by *admentry* when using the Inquiry screen. Note that the Contact code must be in the Contact table. If this default contact is to be a contact that will be used by the *admstats* program to update an inquirer's admissions status, then this code must also be added to the enr_stat_table and the enr_seq_table. This macro is used by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: inq_1 and ginq_1.

'ENABLE_FEAT_ADD_DEF_APP_CTC', 'N'

Defines whether the *admentry* program should add the initial application Contact record by default when using the Application screen. If this value is set to Y, the Application screen will add the Application Contact record defined by the APP_DEFAULT_CTC macro. The Contact record must be in the Contact table before it is used by *admentry*. This macro is used by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: longapp_1 and glongapp_1.

CAUTION: This macro should not be turned on at the time of implementation as new users frequently make the mistake of accidentally giving an inquirer an APPLIED status.

'APP_DEFAULT_CTC', 'APPLIED'

Defines the name of the default applicant Contact Record that will be added automatically by *admentry* when using the Application screen. Note that the Contact code must be in the Contact table. If this default contact is to be a contact that will be used by the *admstats* program to update an inquirer's admissions status, then this code must also be added to the enr_stat_table and the enr_seq_table. This macro is used by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: longapp_1 and glongapp_1.

'ENABLE_FEAT_ADD_DEF_DEC_CTC', 'N'

Defines whether *admentry* should add a default Contact record when using the Decision screen. Setting this macro to Y lets the Decision screen add a default Contact record defined by ADM_DEC_DEF_CTC macro. This macro is used by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: dec_1 and gdec_1.

CAUTION: This macro should not be turned on at the time of implementation as new users frequently make the mistake of accidentally giving an applicant an incorrect status.

'ENABLE_FEAT_DEC_RPT', 'Y'

Defines which document tracking report can be run from the Contact Reports submenu. A Y will display the Eligible for Decision report (elig_dec) which does not require any customization, and searches for missing documents in the Contact record (ctc_rec), Education record (ed_rec), Exam record (exam_rec), and the Interview/Recommendation record (intvwrecom_rec). An N will display another Eligible for Decision report (appcomp_pa) which requires customization, and only searches for missing documents in the Contact record. This macro is used by the \$CARSPATH/menusrc/admit/reports/ctcrpts/menudesc file.

'ENABLE_FEAT_CALLENTRY', 'N'

Controls the availability of call entry processing at the institution. Setting this macro to Y will add the Call Entry submenu to the Admissions Processing menu and enable the admissions office to use call entry to recruit students. This macro is used by the \$CARSPATH/menusrc/admit/menudesc file.

'ENABLE_FEAT_ADM_GRAD', 'Y'

Defines whether the admissions office can process graduate student applications. This macro should be set to Y if the institution offers academic programs for graduate students. This macro is used by the \$CARSPATH/menusrc/admit/screens/menudesc file.

'ENABLE_FEAT_ADM_WAITLIST', 'Y'

Defines whether the admissions office can place applicants on a waiting list for admission. Setting this feature to Y adds the waitlist rank field to the application and decision screens, and adds the Update Waitlist menu option to the Admissions Processing menu. This macro is used by the \$CARSPATH/menusrc/admit/screens/menudesc file, and by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: dec_1, gdec_1, longapp_1, and glongapp_1.

'ADM_WAITLIST_TRACK_BY', 'MAJR'

Defines the method by which the institution uses waitlists. Setting this macro value to PROG indicates the admissions office has one admission waitlist for all undergraduate or graduate applicants. Setting this macro value to MAJR indicates the admissions office has multiple admissions waitlists for the various majors within the undergraduate or graduate program. This macro is used by the \$CARSPATH/menusrc/screens/menudesc file.

'ADM_MAX_WAITLIST', '99'

Defines the maximum number of students the admissions office plans to place on a waiting list. This number must be larger than the number of potential majors using waitlists. This macro is used by the following files in the \$CARSPATH/modules/admit/reports directory: setwaitrnk and setmajrnk.

'ENABLE_EARLY_DECISION', 'Y'

Defines whether the Recruiting/Admissions Application screen displays the Early Decision field. This macro is used by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: longapp_1 and glongapp_1.

'ENABLE_IMMUNIZATIONS', 'N'

Defines whether *admentry* provides access to the Immunization Record detail window that records students' immunizations. This macro is used by the \$CARSPATH/include/applic/admit file.

'ENABLE_VIEW_FIN_AID', 'Y'

Defines whether *admentry* provides access to the Financial Aid Record detail window. This view-only window displays financial aid awards that have been processed by the financial aid office. This macro is used by the \$CARSPATH/include/applic/admit file.

CAUTION: This detail window must not be used to add or update financial aid records (aid_rec).

'ENABLE_FEAT_ADM_BUSINESS', 'N'

Defines whether Business records can be entered and tracked by the admissions office. If set to Y, the Business entry screen option will appear on the Forms menu when loading *admentry*. This macro should be set to Y if the admissions office needs to create ID records and business records for businesses. For example, if the admissions office tracks applicants' hometown newspapers, they will need to create ID numbers for newspapers using the Business screen before they can enter the newspaper ID numbers in the profile_rec.news1_id or profile_rec.news2_id fields. This macro is used by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: admmenu and gadmmenu.

'ENABLE_FEAT_AUTO_ADMSTATS', 'Y'

Determines whether the *admstats* program is called each time a prospect's Contact records are updated. Setting this macro to Y eliminates the need to run the *admstats* program in a batch mode to change the status. This macro is used by the \$CARSPATH/include/applic/admit file.

'ENABLE_FEAT_AUTO_DOC_TRACK', 'Y'

Controls whether the *reqadmdoc* SQL statement is invoked to automatically evaluate and add application document tracking records, (*ctc_rec*, *exam_rec*, *intvwrecom_rec*). If this macro is set to Y, the *adm_rec.add_doc* and *adm_rec.doc_ctgry* fields will appear on the Application screen. This macro is used by the `$CARSPATH/include/applic/admit` file, and the `$CARSPATH/modules/admit/progscr/admentry/longapp_1 & glongapp_1` files.

'ENABLE_FEAT_SSNO_CHECK', 'Y'

When this macro is set to Y, the *admentry* program ensures there are no duplicate social security numbers in any other ID record when doing data entry, and automatically enters dashes for the SSN.

'ENABLE_FEAT_PHONE_CHECK', 'Y'

When this macro is set to Y, the *admentry* program automatically inserts dashes within any phone number field, (including the phone number fields in the Employment and Other Address detail windows). This feature requires a data-entry person to enter an area code or three zeros if the area code is unknown. If this feature is turned on, a two-digit country code cannot be entered in lieu of the dashes for international students. If the Admissions office cannot operate with these restrictions, then this feature should be turned off.

'ENABLE_SCHEDULE_ENTRY', 'Y'

Defines whether the admissions office uses the Schedule Entry program (by setting the macro to Y) or the PERFORM screen to schedule recruitment trips to high schools and other colleges. Setting this macro to Y will enable the menu option for the newer entry program to appear on the Admissions Processing menu. This macro is used by the `$CARSPATH/menusrc/admit/screens/menudesc` file.

'ENABLE_FEAT_PREAPP', 'Y'

Determines whether to add another menu option to the Communications Management menu (by setting the macro to Y) that can be used to search the database for inquiries who have already sent in supporting documents (e.g., transcripts, recommendations, test scores), prior to their application. This feature also creates an (E)xpected Contact record that can be used to generate a letter telling a student what supporting documents have already arrived into the admissions office. This macro is used by the `$CARSPATH/menusrc/admit/commgmt/menudesc` file.

'ENABLE_FEAT_ELECAPP', 'N'

Controls whether to include Electronic Application (*elecapp*) menu options (by setting the macro to Y), and controls the capturing and tracking of electronic application information. This macro is used by the `$CARSPATH/menusrc/admit/menudesc` file.

'ENABLE_FEAT_ELECAPP_ED_CHK', 'N'

Determines whether the *elecapp* program will automatically add Education records. If enabled, during the update of an existing student's data with electronic application data, each of the applicant's Education records will be checked against the existing student's Education records to see if an Education record already exists with the applicant's education ID, CEEB, and School ID. If a record already exists, then the applicant's Education record is skipped. If a record does not exist, then the applicant's Education record is added to the existing student's data. If disabled, then all of the applicant's Education records are added to the existing student's data.

'ENABLE_FEAT_ELECAPP_CHGSCR', 'N'

Determines whether the *elecapp* program is to display the Change Student Information screen when updating a pre-existing student's data.

‘ENABLE_ADMHIST_PROMPT’, ‘Y’

When this macro is set to ‘Y’ the *admentry* and *callentry* programs first asks the user if he/she wants to create an Admission History record whenever he/she changes the planned session and/or year of enrollment for a student. This macro is used by the \$CARSPATH/include/applic/admit file.

‘ADM_CL_DEF’, ‘FF’

Defines the default Classification code most frequently used by the admissions office during data entry. This code must also exist in the Classification table (*cl_table*). This macro is used by the \$CARSPATH/menuopt/admit/informers/studctc file and the following files in the \$CARSPATH/modules/admit/screens directory: *admit*, *admit_q*, and *enrstat*.

‘ADM_CTC_DEF’, ‘ACTIVITY’

Sets the default letter Contact code that is used most often by the admissions office. The Contact code should also be defined in the Contact table. This macro is used by the \$CARSPATH/menuopt/admit/informers/studctc file and the following files in the \$CARSPATH/menuopt/admit/others directory: *ctcall*, *ctcond*, *ctcexpect*, and *ctcnct*.

‘ADM_COMP_CTC’, ‘APPCOMP’

Defines the Contact code used to indicate all supporting documents (e.g., transcripts, test scores, recommendations) for an applicant have arrived into the admissions office. This Contact code must also be defined in the Contact table. This macro is used by the \$CARSPATH/menuopt/admit/reports/appcomp_pa file.

‘ADM_DEC_DEF_CTC’, ‘DECISION’

Defines the name of the default Contact record that *admentry* adds automatically when the end user uses the Decision screen. Note that the Contact code must be defined in the Contact table. This macro is used by the \$CARSPATH/modules/admit/reports/appcomp_pa file, the \$CARSPATH/menuopt/admit/reports/appcomp_pa file, and the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: *dec_1* and *gdec_1*.

‘ADM_EDGOAL_DEF’, ‘B’

Sets the default educational goal value used by the data entry screens in *admentry*. The *adm_rec.ed_goal* field is primarily used by community colleges to accommodate a frequently asked question on community college applications. The value for this macro must also be a value in the ADM_EDGOAL_VALID macro. This macro is used by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: *longapp_1* and *glongapp_1*, and also the following files in the \$CARSPATH/modules/admit/screens directory: *admit* and *admit_q*.

‘ADM_EDGOAL_VALID’, ‘T,A, B, M, P,O’

Defines all valid values for all possible educational goals on the data entry screens in *admentry*. These values will be entered in the *adm_rec.ed_goal* field.

‘ADM_EDGOAL_INCL’, ‘include=(ADM_EDGOAL_VALID), upshift’

Defines the include that enters valid education goal values in uppercase. This macro is used by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: *longapp_1* and *glongapp_1*, and also the following files in the \$CARSPATH/modules/admit/screens directory: *admit* and *admit_q*.

‘ADM_EDGOAL_EX’, ‘(T)echnical, (A)ssc., (B)achelor, (M)asters, (P)hd., (O)ther’

Explains the valid values for the admissions education goals on the data entry screens in *admentry*. This macro is used by the following files in the \$CARSPATH/modules/admit/progscr directory: *longapp_1* and *glongapp_1*.

‘ACT_EXAM_DEF’, ‘ACT’

Defines the default Examination code to be used in Recruiting/Admissions menu options. This macro is used by the \$CARSPATH/menuopt/admit/reports/actprof file.

‘ADM_INQ_CL_DEF’, ‘JR’

Defines the default high school classification a student may have at the time he/she inquires about your institution. This macro can be used with the `adm_rec.inq_cl` field. This value must also exist in the Classification table (`cl_table`). This macro is used by the following files in the `$CARSPATH/modules/admit/screens` directory: `admit` and `admit_q`.

'ADM_OFFICE_DEF', 'ADMS'

Defines the default admissions office code to be used by *admentry*. This value must also exist in the Office table (`ofc_table`). This macro is used by the `$CARSPATH/modules/admit/others/screens/statuses` file, the `$CARSPATH/menuopt/admit/informers/moveleads` file and by the following files in the `$CARSPATH/modules/admit/others` directory: `actcol`, `etscol`, and `hstape`.

'ADM_PROG_MENU', 'admenu'

Determines the default form menu used by the *admentry* menu option. This macro is used by the `$CARSPATH/menuopt/admit/programs/adme` file.

'ADM_PROG_OFFICE', 'ADM_OFFICE_DEF'

Defines the default admissions office code for the default academic program, used exclusively by *admentry*. This macro is used by the following files in the `$CARSPATH/menuopt/admit/programs` directory: `adme`, `gadme`, and `calle`.

'ADM_PROG_PROGRAM', 'PROG_DEF'

Defines the default academic program to be used by *admentry*. This value must also exist in the Program table (`prog_table`). This macro is used by the `$CARSPATH/modules/admit/scripts/admprocess` file, and the following files in the `$CARSPATH/menuopt/admit/programs` directory: `adme`, `adms`, `elecapp`, and `elecappr`.

'ADM_PROG_TICK', 'TICK_ADM'

Defines the default Tickler code to be used by *admentry*. This value must also exist in the Tickler table (`tick_table`), and linked to its corresponding Program code in the Program table. This macro is used by the `$CARSPATH/menuopt/admit/reports/callctcs` file, the `$CARSPATH/modules/admit/scripts/admprocess` file, and the following files in the `$CARSPATH/modules/admit/programs` directory: `adme`, `adms`, `elecapp`, `elecappr`, and `calle`.

'ADM_STAT_EXCLUDE', 'DENIED'

Provides the admissions status used to indicate an applicant has been denied admission to your institution. This code must also exist in the Enrollment Status table (`enr_stat_table`), the Enrollment Sequence table (`enr_seq_table`), and the Contact table (`ctc_table`). This macro is used by the `$CARSPATH/menuopt/admit/informers/colldayctc` file

'ADM_STAT_NOFE', 'APPLNOFE'

Defines the admissions status value that indicates a student has applied for admission, but has not yet paid the application fee (e.g., `APPLNOFE`). This status is also used for students who have applied via the Web application. This macro is used by the `$CARSPATH/modules/admit/reports/not_elig` file, and the `$CARSPATH/include/applic/admit` file.

Note: This macro should be defined after the Enrollment Status table has been populated. A contact with the same name must also exist in the Contact table.

'ADM_STAT_ACPT', 'ACCEPTED'

Sets the admissions status value that indicates a student has been accepted for admission (e.g., `ACCEPTED`, `ADMITTED`).

Note: This macro should be defined after the Enrollment Status table has been populated. A contact with the same name must also exist in the Contact table.

'ADM_STAT_WAIT', 'WAITLIST'

Provides the admissions status value that indicates a student has been placed on a waiting list for admission (e.g., WAITLIST, ALTRNATE). This macro is used by the following files in the \$CARSPATH/modules/admit/reports directory: setmajrnk and setwaitrnk.

Note: You should define the value for this macro after you populate the Enrollment Status table. A contact with the same name must also exist in the Contact table.

'ADM_STAT_CONF', 'CONFIRM'

Defines the admissions status value that indicates a student has confirmed their decision to enroll at your institution (e.g., CONFIRM, DEPOSIT). This macro is used by the \$CARSPATH/menuopt/admit/informers/regctc file and by the \$CARSPATH/modules/admit/informers/regctc file.

Note: You should define the value for this macro after you populate the Enrollment Status table. A contact with the same name must also exist in the Contact table.

'ADM_STAT_DECLINE', 'DECLINED'

Sets the admissions status value that indicates a student has declined an offer of acceptance to your institution (e.g., DECLINED, REFUSED).

Note: You should define the value for this macro after you populate the Enrollment Status table. A contact with the same name must also exist in the Contact table.

'ADM_STAT_ENR', 'ENROLLED'

Provides the admissions status value that indicates a student has enrolled in classes (e.g., ENROLLED, REGIST). This macro is used by the \$CARSPATH/menuopt/admit/informers/regctc file and by the \$CARSPATH/modules/admit/informers/regctc file.

Note: You should define the value for this macro after you populate the Enrollment Status table. A contact with the same name must also exist in the Contact table.

'ADM_STAT_WTHDRW', 'WITHDREW'

Defines the admissions status value that indicates a student's application has been withdrawn from consideration.

Note: You should define the value for this macro after you populate the Enrollment Status table. A contact with the same name must also exist in the Contact table.

'ADM_STAT_NOSHOW', 'NOSHOW'

Sets the admissions status value that indicates a student who paid the enrollment deposit, but did not enroll (e.g., NOSHOW, ABSENT). This macro is used by the \$CARSPATH/menuopt/admit/informers/regctc file and by the \$CARSPATH/modules/admit/informers/regctc file.

Note: You should define the value for this macro after you populate the Enrollment Status table. A contact with the same name must also exist in the Contact table.

'ADM_INTENDHRS_DEF', '12'

Defines the default value to be used for the Intended Enrollment Hours (adm_rec.intend_hrs_enr) on the *admentry* data entry screens. This macro is used by the \$CARSPATH/modules/admit/progscr/callentry/callent_1 file, the following files in the \$CARSPATH/modules/admit/screens directory: admit, admit_q, and enrstat, and the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: longapp_1 and glongapp_1.

'ADM_INTENDHRS_VALID', '0:18'

Identifies the valid range of values that can be entered for Intended Enrollment Hours on the *admentry* data entry screens.

'ADM_INTENDHRS_INCL', 'include=(ADM_INTENDHRS_VALID)'

Defines the include that uses the default value set by the ADM_INTENDHRS_VALID macro for Intended Enrollment Hours on the *admentry* data entry screens. This macro is used by the \$CARSPATH/modules/admit/progscr/callentry/callent_1 file, the following files in the \$CARSPATH/modules/admit/screens directory: admit, admit_q, and enrstat, and the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: longapp_1 and glongapp_1.

'ADM_INTENDHRS_EX', 'Fulltime >= (12), Parttime < (12).'

Sets the values that will be used to define full time (greater than or equal to 12 hours) and part time (less than 12 hours). This macro is used by the \$CARSPATH/modules/admit/progscr/telmkt_1 file and the following files in the \$CARSPATH/modules/admit/progscr/admit/admentry directory: longapp_1 and glongapp_1.

'ADM_TICKLEVEL_DEF', 'M'

Provides the default value to be used for a student's admissions Tickler level (tick_rec.level). This macro is used by the \$CARSPATH/modules/admit/informers/moveleads file, and by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: inq_1, longapp_2, ginq_1, and glongapp_2.

'ADM_TICKLEVEL_VALID', 'A, M, Z'

Sets the valid values that can be entered for a student's admissions Tickler level. This macro is used by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: inq_1, ginq_1, longapp_1, and glongapp_1.

'ADM_TICKLEVEL_INCL', 'include=(ADM_TICKLEVEL_VALID), upshift'

Identifies the include that sets the value entered for the admissions Tickler level value (e.g., A, M, Z) to uppercase. This macro is used by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: inq_1, ginq_1, longapp_1, and glongapp_1.

'ADM_TICK_VALID', 'TICK_ADM,TICK_ADMG,TICK_SCHL'

Defines the valid admissions Tickler codes used by the admissions office. These Contact Tickler codes must also be defined in the Tickler table (tick_table). This macro is used by the \$CARSPATH/menuopt/admit/scripts/ltrrun.adm file, the following files in the \$CARSPATH/menuopt/admit/programs directory: adms, elecapp, elecappr, and calle, and the following files in the \$CARSPATH/menuopt/admit/reports directory: callctcs, callmktctc, callstat, callstat2, callstatd, and tcallmctc.

'ADM_TICKTRACK_DEF', 'HS'

Sets the default Tickler track value that will appear on the Inquiry and Application entry screens in the tick_rec.trk field (e.g., HS = High School track). The Track code must also be defined in the Tickler Track table (trk_table). This macro is used by the \$CARSPATH/modules/admit/informers/moveleads file, and by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: inq_1, longapp_2, ginq_1, and glongapp_2.

'ADM_CTC_APPT_DEF', 'APPTMENT'

Defines the default Contact code value to be used for scheduling student appointments with admissions counselors. This Contact code must also be defined in the Contact table. This macro is used by the \$CARSPATH/menuopt/admit/reports/schdcoun file and the \$CARSPATH/menuopt/admit/scripts/schdcal file.

‘ADM_REPORT_SORT_DEF’, ‘cnslr_name’

Controls the default sort value to be used with the User Defined Profile report. This macro is used by the \$CARSPATH/menuopt/admit/others/profile file.

‘ADM_ACE_DEF’, ‘ltradmit’

Defines the default ACE report value to be used when creating letters/labels in the Create Admissions Letters menu option located in the Communications Management menu. This macro is used by the \$CARSPATH/menuopt/admit/scripts/ltrun.adm file.

‘ADM_EXAM_1’, ‘ACT’

‘ADM_EXAM_2’, ‘SAT’

‘ADM_EXAM_3’, ‘TOFL’

‘ADM_EXAM_4’, ‘GRE’

These macros define the codes of the four most commonly used exams tracked by the Admissions office. These codes must also exist in the Exam table (exam_table). This macro is used by the \$CARSPATH/modules/admit/progscr/admentry/longapp_3 & glongapp_3 files, and the \$CARSPATH/modules/admit/progscr/callentry/callent_2 file.

‘ETS_REF_SRC’, ‘ETSS’

Sets the default Referral Source code used when loading ETS student search tapes. This macro is used by the following files in the \$CARSPATH/modules/admit/others directory: etsss and etsss_rep.

Note: This macro value must also exist in the Reference table (ref_table).

‘ACT_REF_SRC’, ‘ACTS’

Provides the default Referral Source code used when loading the ACT student search tapes. The macro is used by the \$CARSPATH/modules/admit/others/actss_rep file.

Note: This macro value must also exist in the Reference table (ref_table).

‘ADM_DOC_DEF’, ‘N’

This macro is no longer used by the Admissions module, and should be left with a value of N. This macro is used by the \$CARSPATH/modules/common/screens/tctc file.

‘ADM_DOC_VALID’, ‘N,R’

This macro is no longer used. It was once used to define valid entries for the Admissions Document Requirement code used in the Contact Table Entry screen. These values are further defined in the ADM_DOC_EX macro.

‘ADM_DOC_INCL’, ‘include=(ADM_DOC_VALID), upshift’

This macro is no longer used. It was once used to define the include for the ADM_DOC_VALID macro that enters the value in uppercase. This macro is used by the \$CARSPATH/modules/common/screens/tctc file.

‘ADM_DOC_EX’, ‘Enter N(one) or R(equired) or leave blank’

This macro is no longer used. It was once used to describe the valid values for the admissions Document Requirement codes used in the comment line of the Contact Table PERFORM screen. This macro is used by the \$CARSPATH/modules/commn/screens/tctc file.

‘CALL CTC COMM CODE’, ‘CALL’

Call Entry Contacts Communication Code Defines the communication code to be used when defining callentry contacts in the Contact table. This code must also exist in the Communication table. This macro is used by the

\$CARSPATH/modules/admit/progscr/callentry/ctc file, and various files within the \$CARSPATH/modules/admit/reports directory.

'CALL_RESP_DEF', 'T'

This macro should be set with the default code used to indicate the person responsible (i.e., T for Telemarketer or C for Counselor), for calling a prospective student. This macro is used by the \$CARSPATH/modules/admit/progscr/callentry/callent_1 file.

'CALL_RESP_VALID', 'T,C,B'

This macro should be set with the codes used to indicate all possible people who might be responsible for calling prospective students. This macro is used by the \$CARSPATH/modules/admit/progscr/callentry/callent_1 file.

'CALL_RESP_INCL', 'include = (CALL_RESP_VALID), upshift'

Defines the admissions call entry include values for all possible people responsible for making call entry calls to a particular prospective student. This macro is used by the \$CARSPATH/modules/admit/progscr/callentry/callent_1 file.

'CALL_RESP_EX', 'Enter (T)elemarketer, (C)ounselor, or (B)oth.'

This macro should contain the descriptions for the codes listed in the CALL RESP VALID macro. This macro is used by the \$CARSPATH/modules/admit/progscr/callentry/callent_1 file.

'CALL_INTEREST_INCL', 'include = (ADM_TICKLEVEL_VALID), upshift'

This macro should include all the codes used to indicate a prospective student's interest level in the college. If you want to use the same interest levels used by the Tickler program leave the macro set as it is shown above. This macro is used by the \$CARSPATH/modules/admit/progscr/callentry/callent_1 file.

'INST_ADMISSION_PHONE', '(513) 563-ADMIT'

Defines the phone number used in the Admissions Web Application. You will need to update this macro to reflect your institution.

'ASSIGN_CNLSR_BY_SCHL', 'Y'

When set to Y, this macro enables the *admentry* program, and Data conversion scripts to copy the sch_rec.cnslr_id field to the adm_rec.cnslr_id field. If admission counselor assignments are based on a student's major, set this macro to N. This macro is used by the \$CARSPATH/include/applic/admit file, and various files within the \$CARSPATH/modules/admit/others directory.

'ADM_LTR_ACE_INCL', (

ltradmit = "Create all other letters",

ltrenclose = "Create letters with enclosures",

ltrhsg = "Create housing assignment letters",

ltrincapp = "Create Incomplete Application letters",

ltrintconf = "Create Interview Confirmation letters",

ltrmistran = "Create missing Full/Final transcript ltrs",

ltrnoltr = "Complete contacts without creating letters",

ltrpreapp = "Create Documents Already Arrived letters",

ltrschvst = "Create School Visitation letters",

ltrparent = "Create letters to parents")

This macro lists the name and purpose of each Ace report used by the Admissions office to create merge files for letter/label creation. The values of this macro can be seen in the Report parameter when using the Create Admissions Letters menu option. Any locally created 'ltr' Ace report should be added to this macro. This macro is used by the \$CARSPATH/menuopt/admit/scripts/ltrrun.adm file.

'UNDG_DOC_CTGRY_DEF', 'NONE'

Defines the document category code used most often for undergraduate applicants. The value assigned to this macro must be listed in the reqadmdoc_table. Because of the unique

index of the reqadmdoc_table you cannot use the same code for both undergraduate and graduate document categories. This macro is used by the \$CARSPATH/modules/admit/progscr/admentry/longapp_1 file.

'GRAD_DOC_CTGRY_DEF', 'BLNK'

Defines the document category code used most often for graduate applicants. The value assigned to this macro must be listed in the reqadmdoc_table. Because of the unique index of the reqadmdoc_table you cannot use the same code for both undergraduate and graduate document categories. This macro is used by the \$CARSPATH/modules/admit/progscr/admentry/glongapp_1 file.

'IMPORTPATH', 'IDS_PATH/pcadm'

Path names for admissions import processing.

'ACT_IMPORTFILE', 'act'

Filename containing the ACT data, which is placed in the pcdm directory.

'ACTSS_IMPORTFILE', 'actss'

Filename containing the ACTSS data, which is placed in the pcdm directory.

'ETS_IMPORTFILE', 'ets'

Filename containing the ACT data, which is placed in the pcdm directory.

'ETSSS_IMPORTFILE', 'etsss'

Filename containing the ETSSS data, which is placed in the pcdm directory.

'ACTCOL_IMPORTFILE', 'actcol'

Filename containing the ACTCOL data, which is placed in the pcdm directory.

'SATEXAM_IMPORTFILE', 'satexam'

Filename containing the SATEXAM data, which is placed in the pcdm directory.

'ETSCOL_IMPORTFILE', 'etscol'

Filename containing the ETSCOL data, which is placed in the pcdm directory.

'HIGHSCHOOL_IMPORTFILE', 'highschool'

Filename containing the highschool data, which is placed in the pcdm directory.

'ENABLE_DISPLAY_ALL_IMPORTS', 'N'

If enabled, then all data from the import files (imported and non-imported) will display within the *admimport* program. If disabled, then only data from the import files that has not been imported into the database will be displayed within the *admimport* process.

'ENABLE_BLANK_SSN', 'N'

If enabled, the *admimport* program allows import records without a Social Security Number (SSN) to be inserted into the database. If disabled, import records without a SSN will not be inserted into the database.

'ENABLE_DUPLICATE_SSN', 'Y'

If enabled, then an import record is created containing the duplicate that was found in the datafile based upon the SSN in the import record matching an existing SSN in the database. The user will need to resolve the duplication before proceeding. If disabled, then checking is not done on the SSN.

'ENABLE_DUPLICATE_ZIP', 'N'

If enabled, then an import record is created containing the duplicate that was found in the datafile based upon the Name and ZIP code in the import record matching an existing student's in the database. If disabled, then duplication checking is not done on the Name and ZIP code.

'ENABLE_DUPLICATE_DOB', 'N'

If enabled, then an import record is created containing the duplicate that was found in the datafile based upon the Name and Birthdate in the import record matching an existing

student's in the database. If disabled, then duplication checking is not done on the Name and Birthdate.

'ACT_FMT_DEF', 'ACT'

Defines the name used to reference the ACT datafile format.

'ETS_FMT_DEF', 'ETS'

Defines the name used to reference the ETS datafile format.

'ADM_IMPORTFMT_VALID', 'ACT_FMT_DEF', 'ETS_FMT_DEF'

Defines the valid default values used to reference the various datafile formats used in the *admimport* program.

'ADM_IMPORTFMT_INCL', 'ADM_IMPORTFMT_VALID'

Defines the valid include values used to reference the various datafile formats used in the *admimport* program.

Recruiting/Admissions Document Control Macros (ADM_DOC_1 - ADM_DOC_12)

The ADM_DOC_1 through ADM_DOC_12 macros should be assigned values of contacts used to track miscellaneous documents that do not fall within the transcript, recommendation, or exam score categories. These macro values should reflect an incoming document contact code that exists in the Contact table. The macros used here will display on page three of the application screen if the student has a contact record by the same name. If your Admissions office does not use an admissions status code to indicate that a student's application is complete, but a contact record is still used to indicate a complete application (i.e., APPCOMP), then this contact code should be used as the value for either ADM_DOC_1 or ADM_DOC_12. The files that use these macros are \$CARSPATH/modules/admit/progscr/admentry/longapp_3 & glongapp_3, and \$CARSPATH/modules/admit/progscr/callentry/callent_2.

'ADM_DOC_1', 'R_APP'

Defines the Contact code for application received.

'ADM_DOC_2', 'R_APPEE'

Defines the Contact code for application fee received.

'ADM_DOC_3', 'R_SCORES'

Defines the Contact code for SAT/ACT scores received.

'ADM_DOC_4', 'R_STUDEV'

Defines the Contact code for student development form received.

'ADM_DOC_5', 'R_REF1'

Defines the Contact code for first recommendation form received.

'ADM_DOC_6', 'ESSAY'

Defines the Contact code for student essay received.

'ADM_DOC_7', 'HEALTH'

Defines the Contact code for health form received.

'ADM_DOC_8', 'R_REC2'

Defines the Contact code for second recommendation form received.

'ADM_DOC_9', 'R_HSTRN1'

Defines the Contact code for partial high school transcript received.

'ADM_DOC_10', 'R_HSTRN2'

Defines the Contact code for high school transcript received.

'ADM_DOC_11', 'R_TRCOL1'

Defines the Contact code for college transcript from school 1 received.

'ADM_DOC_12', 'R_TRCOL2'

Defines the Contact code for college transcript from school 2 received.

Macros In The Leads File

Introduction

The following macros are used by Lead Processing and appear in the order in which they are located in the \$CARSPATH/macros/custom/leads file.

'ENABLE_FEAT_LEAD_PROCESSING', 'Y'

If this macro is set to a value of "Y", the Lead Processing sub-menu will appear on the Recruiting/Admissions: Main Menu. If this macro is set to a value of "N", the Lead Processing sub-menu will not appear on the Recruiting/Admissions: Main Menu. This macro is used by the \$CARSPATH/menusrc/admit/menudesc file.

'ENABLE_FEAT_LEAD_TICKLER', 'Y'

If this macro is set to a value of "Y", the Lead Tickler sub-menu will appear on the Lead Processing: Lead Communications Management Menu, and the lead tickler fields will appear at the bottom of the Lead Form data entry screen. This macro is used by the following files: \$CARSPATH/menusrc/admit/leads/commgmt/menudesc, \$CARSPATH/modules/admit/progscr/leadent/lead, \$CARSPATH/include/applic/leads.

'ENABLE_FEAT_ADD_DEF_LEAD_CTC', 'Y'

If this macro is set to "Y", the Lead Entry program will automatically add an expected lead contact record whenever a new lead record is manually entered on the Lead Form screen. If this feature is turned on you eliminate the need to go into the Lead Contacts detail window, and manually add an expected lead contact record. This macro is used by the \$CARSPATH/include/applic/leads file.

'LEAD_DEFAULT_CTC', 'LEADLET1'

This macro tells the Lead Entry program which lead contact resource code to add automatically if the ENABLE_FEAT_ADD_DEF_LEAD_CTC macro has been turned on. This lead contact code must also exist in the Contact table. This macro is used by the \$CARSPATH/include/applic/leads file.

'LEAD_YEAR_INCL', 'include = (1985:2020)'

This macro defines the valid range of years that can be passed to menu options that run lead reports. This macro is used by the following files:
\$CARSPATH/menuopt/admit/reports/leadceeb,
\$CARSPATH/menuopt/admit/reports/leadmaj,
\$CARSPATH/menuopt/admit/reports/leadresrc,
\$CARSPATH/menuopt/admit/reports/leadsex,
\$CARSPATH/menuopt/admit/reports/leadsrc,
\$CARSPATH/menuopt/admit/reports/leadst,
\$CARSPATH/menuopt/admit/reports/leadzip.

'LEAD_TICKTRACK_DEF', 'HS'

This macro defines the Lead Tickler Track Default used in the Lead screen of the *leadentry* program.

The following macro is located in the \$CARSPATH/macros/custom/common file.

'TICK_LEAD', 'LEAD'

This macro defines the tickler code used for lead contact records, and lead tickler records. The value of this macro must be defined in the Tickler table, and used as the tickler code when defining lead contacts in the Contact table. If using the Lead Tickler program, the value of this macro must be used when defining a lead tickler strategy in the various tickler tables, (i.e., trk_table, step_table, step_rec, stepreq_rec, stepctc_rec, stepobj_rec). The following files use this macro:

```
$CARSPATH/menuopt/commgmt/ldtickler/tent.LEAD  
$CARSPATH/menuopt/commgmt/ldtickler/tick.LEAD  
$CARSPATH/menuopt/commgmt/ldtickler/tick.rLEAD  
$CARSPATH/menuopt/commgmt/ldtickler/tick.oLEAD  
$CARSPATH/menuopt/admit/scripts/ltrrun.ld  
$CARSPATH/menuopt/admit/informers/leadctc  
$CARSPATH/menuopt/admit/reports/leadctcdu  
$CARSPATH/menuopt/admit/programs/leade  
$CARSPATH/modules/admit/scripts/admprocess  
$CARSPATH/modules/common/scripts/ltbrun  
$CARSPATH/modules/admit/informers/moveleads
```


Macros in the Periodic File

Introduction

The following macros reside in the macros/custom/periodic file. They define default values and valid codes for data entry and display. You must update these macros annually, in accordance with the admissions office application processing cycle.

Macros

'ADM_ENR_SESS', 'FA'

Sets the value for the default planned enrollment session. This macro is used throughout the Recruiting/Admissions module.

'ADM_ENR_YR', '1999'

Defines the value for the default planned enrollment year. This macro is used throughout the Recruiting/Admissions module.

'ADM_HS_GRAD_DATE', '06/01/1999'

Controls the value for the default high school graduation date. This macro is used by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: inq_1, ginq_1, ref_1, and shortapp_1.

'ADM_TICK_CMPL_DATE', '09/01/1999'

Provides the value for the default admissions Tickler completion date. This macro is used by the following files in the \$CARSPATH/modules/admit/progscr/admentry directory: longapp_2, and glongapp_2.

'LEAD_ENR_SESS', 'FA'

This macro defines the default session of planned enrollment for a lead. When the Lead Entry program is in Insert Mode, the value of this macro will automatically appear in the "Session" field. To save keystrokes when entering lead data, set this macro to the most common academic term students enroll at your institution, (i.e., FA for Fall). The session code used for this macro's value must also exist in the Session table. This macro is used by the \$CARSPATH/modules/admit/progscr/leadent/lead file.

'LEAD_ENR_YR', '1999'

This macro defines the default year of planned enrollment for a lead. When the Lead Entry program is in Insert Mode, the value of this macro will automatically appear in the "Year" field. To save keystrokes when entering lead data, set this macro to the next academic year students may enroll at your institution, (i.e., 1999). This macro is used by the \$CARSPATH/modules/admit/progscr/leadent/lead file.

Includes

Introduction

The Recruiting/Admissions product uses includes that determine the features, which are enabled in various programs used by the Admissions module. An include can either be a compile option that enables or disables a feature, or a default value for a feature.

To enable a feature in the Recruiting/Admissions programs, you must define an include in the \$CARSPATH/include/applic/admit file.

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 Recruiting/Admissions programs 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 Recruiting/Admissions programs.

Macro Dependency

Includes usually have a dependency on macros. For Recruiting/Admissions includes, you do not directly modify the includes. Instead, you must modify a corresponding macro value and then reinstall the include/applic/admit file. After the include has been reinstalled, you must then reinstall the C code that sources the include. For example, if the Auto Admstats feature was turned off within *admentry* and you wanted to turn it on, you would need to complete the following procedure:

1. Enter the following at the UNIX prompt:

```
cd macros/custom  
make co F=admissions  
vi admissions
```

2. Within the admissions macro file, search for and turn on the macro to enable Auto Admstats as in the following example:

```
m4_define(`ENABLE_FEAT_AUTO_ADMSTATS',`Y')
```

3. Save and exit the file.
4. Enter the following at the UNIX prompt to check in your change and to reinstall the related include and source files:

```
make cii F=admissions  
cd include/applic  
make reinstall F=admit  
cd src/admit/admentry  
make reinstall
```

How to Locate Includes

To locate admissions includes, access the \$CARSPATH/include/applic/admit file.

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

Application Includes

The following list describes each include used by the Recruiting/Admissions product.

m4_keepif(ENABLE_FEAT_AUTO_DOC_TRACK,~`Y~')

#define ENT_DOC_TRACK

m4_keepend

When set to Y, causes *admentry* to automatically run the *admdocone* SQL statement (located in the \$CARSPATH/modules/admit/informers directory), whenever a student's adm_rec.add_doc field is flagged as Y.

m4_keepif(ENABLE_FEAT_AUTO_ADMSTATS,~`Y~')

#define ENT_CALL_ADMSTATS

m4_keepend

When set to Y, causes *admentry* to automatically run *admstats* whenever a change is made to a student's Contact records.

m4_keepif(ENABLE_IMMUNIZATIONS,~`Y~')

#define ENABLE_IMMUNE

m4_keepend

When set to Y, enables *admentry* to provide access to the Immunizations (immune_rec) detail window.

m4_keepif(ENABLE_VIEW_FIN_AID,~`Y~')

#define ENABLE_FIN_AID

m4_keepend

When set to Y, enables *admentry* to provide access to the Financial Aid Awards (aid_rec) detail window.

#define APP_OFC_ADD_BY "ADM_OFFICE_DEF"

Defines the default Office code to be used by *elecapp*.

#define APP_CTC_RESRC "ADM_STAT_NOFE"

Defines the default Contact code to be used by *elecapp* to update a new applicant's status.

#define APP_AA_CURR_CODE "LOC_AACODE_DEF"

Defines the default Alternate Address code to be used by *elecapp* when creating Alternate Address records (aa_rec).

#define APP_AA_PERM_CODE "ID_AACODE_DEF"

Defines the default Address code to be used by *elecapp* when creating ID records (id_rec).

m4_keepif(ASSIGN_CNLSR_BY_SCHL,~`Y~')

#define ENT_CNLSR_LOOKUP

m4_keepend

This include defines whether or not the admission counselor ID will be automatically looked up from the sch_rec.cnslr_id field when the first ed_rec.sch_id or ed_rec.ceeb field is populated using *admentry*.

m4_keepif(ENABLE_FEAT_PHONE_CHECK,~`Y~')

#define ENT_DASH_PHONE_CHCK

m4_keepend

This include defines whether to allow the *admentry* program to automatically insert dashes within any phone number.

m4_keepif(ENABLE_FEAT_SSNO_CHECK,~`Y~')

#define ENT_DUP_SSNO_CHCK

This include defines whether or not the admentry program will first check for duplicate social security numbers before committing the data to the database. When this feature is turned on the dashes for the SSN will be automatically inserted.

m4_keepif (WEB_ADMIT_ENABLE_EDUCATION,~'Y~)

#define ELECAPP_EDTMP_OPTION

m4_keepend

This include defines whether or not the *elecapp* program will enable or disable the display of the temporary education menu option.

m4_keepif (WEB_ADMIT_ENABLE_ACCOMP,~'Y~)

#define ELECAPP_ACCOMPTMP_OPTION

m4_keepend

This include defines whether or not the *elecapp* program will enable or disable the display of the temporary accomplishment menu option.

m4_keepif (WEB_ADMIT_ENABLE_INVOLVE,~'Y~)

#define ELECAPP_INVLTMP_OPTION

m4_keepend

This include defines whether or not the *elecapp* program will enable or disable the display of the temporary involvement menu option.

m4_keepif (WEB_ADMIT_ENABLE_INTEREST,~'Y~)

#define ELECAPP_INTTMP_OPTION

m4_keepend

This include defines whether or not the *elecapp* program will enable or disable the display of the temporary interest menu option.

m4_keepif (ENABLE_FEAT_ELECAPP_ED_CHK,~'Y~)

#define ENT_ELECAPP_ED_CHK

m4_keepend

This include defines whether or not the *elecapp* program will enable or disable the checking of Education records prior to any inserts into the database.

m4_keepif (ENABLE_FEAT_ELECAPP_CHGSCR,~'Y~)

#define ENT_ELECAPP_CHGSCR

m4_keepend

This include defines whether or not the *elecapp* program will display the Change Student Information screen when updating pre-existing student's data with electronic application data.

m4_keepif(ENABLE_ADMHIST_PROMPT,~'Y~)

#define ENT_ADMHIST_NTERACTIVE

m4_keepend

This include defines whether to have the admentry program to first ask the user if he/she wants to create an *admhist_rec* whenever he/she updates a student's planned session and/or year of enrollment. If this feature is disabled, the admentry program will automatically created an *admhist_rec* without first asking the end-user.

#define IMPORT_PATH

The path name where the import files are stored (ADMIMPORT).

#define ACT_IMPORT_FILE

The import filename (ACT).

#define ACTSS_IMPORT_FILE

The import filename (ACTSS).

#define ETS_IMPORT_FILE

The import filename (ETS).

```
#define ETSSS_IMPORT_FILE
```

The import filename (ETSSS).

```
m4_keepif (ENABLE_DISPLAY_ALL_IMPORTS,~'Y~)
```

```
#define ENT_DISPLAY_ALL_IMPORTS
```

```
m4_keepend
```

If enabled, then the *admimport* program will display all import records (imported and non-imported). If disabled, then only those import records that have not been imported will be displayed within the *admimport* program.

```
# m4_keepif (ENABLE_DUPLICATE_SSN,~'Y~)
```

```
#define ENT_DUPLICATE_SSN
```

```
m4_keepend
```

If enabled, the *admimport* program will force the user to resolve all duplications based on the SSN related to the import data.

```
# m4_keepif (ENABLE_DUPLICATE_ZIP,~'Y~)
```

```
#define ENT_DUPLICATE_ZIP
```

```
m4_keepend
```

If enabled, then the *admimport* program will force the user to resolve all duplications based on the Name and ZIP code related to the import data.

```
# m4_keepif (ENABLE_DUPLICATE_DOB,~'Y~)
```

```
#define ENT_DUPLICATE_DOB
```

```
m4_keepend
```

If enabled, then the *admimport* program will force the user to resolve all duplications based on the Name and Birthdate related to the import data.

```
# m4_keepif (ENABLE_BLANK_SSN,~'Y~)
```

```
#define ENT_BLANK_SSN
```

```
m4_keepend
```

If enabled, the *admimport* program will allow those records with blank SSN's to be added to the database even if other records with blank SSN's exist within the database. If disabled, import records containing blank SSN's will not be added to the database.

Includes Used By Lead Entry

The following list describes each include used by the Lead Entry program within the Recruiting/Admissions product. To locate Lead Entry includes, access the `$CARSPATH/include/applic/leads` file.

Note: You should reinstall the Lead Include file after all lead macros have been defined and installed. After you reinstall the Lead Include file, you should reinstall the Lead Entry program (`$CARSPATH/src/admit/leadent`).

```
m4_keepif(ENABLE_FEAT_LEAD_TICKLER,~'Y~')
```

```
#define ENT_LEAD_TICKLER
```

```
m4_keepend
```

This include tells the Lead Entry program whether or not it should create a lead tickler record (`ldtick_rec`) for a new lead. This include is set by the value of the `ENABLE_FEAT_LEAD_TICKLER` macro.

```
m4_keepif(ENABLE_FEAT_ADD_DEF_LEAD_CTC,~'Y~')
```

```
#define ENT_LEAD_CONTACT
```

```
m4_keepend
```

This include tells the Lead Entry program whether or not it should automatically create an expected lead contact record (ldctc_rec) for a new lead. This include is set by the value of the ENABLE_FEAT_ADD_DEF_LEAD_CTC macro.

#define LEAD_DEF_CTC “LEAD_DEFAULT_CTC”

This include tells the Lead Entry program which lead contact resource code to use if an expected lead contact record is to be automatically created for a new lead. This include is set by the value of the LEAD_DEFAULT_CTC macro.

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 File Details

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. The `def.c` file for *admentry* is located in the `$(CARSPATH)/src/admit/admentry` directory path.

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 (e.g., 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 program *admentry* . It includes representative lines only (the original def.c file contains over 300 lines), and illustrates the common components of a standard CX def.c file.

Note: The legend for the file contents follows:

1. mac.h include
2. schema file def.c's
3. local variables and program parameters
4. functions
5. structure definitions


```

1
#include "mac.h"

2
#define SCREEN_PATH "admit/admentry/"
#define DEFAULT_MENU "admmenu"
#define IDTYPE_SCREEN "Lib/libids/type"
#define MENU_TITLE "ADMISSIONS ENTRY"

3
/* Local variables and program parameters */
int auto_mode = FALSE;
int force_query = FALSE;
long common_id; /* buffer for common id */
long today; /* today's INFORMIX date */
char ofc_addby[] = "ADM "; /* Default office add by code */
char prog_code[] = "UNDG"; /* Default program code to use */

4
/* Local functions */
int selecttctc();
int selectttrk();

/* Special functions */
int idperm();
int edhold();

/* -----
   Array initializations
   ----- */

5
struct param_type prog_params[] = /* array of common db fields */
{
    {'d', (char *)&display_only, PRM_LOGICAL, 0, PRM_TRUE, "display_only",
    "pass parameter to limit to display only"},
    {'p', prog_code, PRM_CHAR, 4, NULL, "prog",
    "program code to be used, default is 'UNDG'"},
    {'T', tick_code, PRM_CHAR, 4, NULL, "tick",
    "tickler code to be used, default is 'ADM '"},
};
int max_params = (sizeof(prog_params)/sizeof(struct param_type));

struct scfile_type scrollfiles[] = /* array of scroll files */
{
    {"accomp_rec", "accomp", 'A', "Accomplishments ", ""},
    {"ctc_rec", "ctc", 'C', "Contacts ", ""},
#ifdef ENABLE_FIN_AID
    {"aid_rec", "aid", 'D', "fin aiD awards ", ""},
#endif
    {"ed_rec", "ed", 'E', "Education ", ""},
    {"enr_stat_rec", "enrstat", 'T', "", "enrollmenT status "},
};
int max_scfiles = (sizeof(scrollfiles)/sizeof(struct scfile_type));

struct file_type filename[] = /* array of filenames */
{
#ifdef ENABLE_IDPERMS
    {"id_rec", "id", NULL, ENT_FLGET|ENT_FLUPD, idperm},
#else
    {"id_rec", "id"},
#endif
    {"adm_rec", "adm_prim", "adm_prim", ENT_AUTOINS},
};
int max_files = (sizeof(filename)/sizeof(struct file_type));

/* File update order. */

```

mac.h Files

Every program uses a macro header (mac.h) file, located in the following path:
 \$CARSPATH/src/admit/admstats.

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 the mac.h file for *admentry*. It illustrates the most common component of a standard CX mac.h file, the includes for header files.

```
#include <util/cars.h>
#include <ctype.h>
#include <util/dbio.h>
#include <applic/entry.h>
#include <util/tbl.h>
#include <applic/common.h>
#include <applic/admit.h>
```

SECTION 6 - ADMISSIONS ENTRY

Overview

Introduction

This section provides reference information about the Admissions Entry (*admentry*) program. The Recruiting/Admissions product uses *admentry* to enter information about individuals, using screens that can be modified by the institution. The *admentry* program lets you perform data retrieval, data entry, and updates of all records used in the admissions office. The screens can be customized to simulate the actual forms used by the admissions office, such as inquiry cards, ACT and SAT forms, and application forms. It also offers ID/Name lookup for parents, high schools and colleges, admissions counselors, churches, businesses, and references.

Program Features Detailed

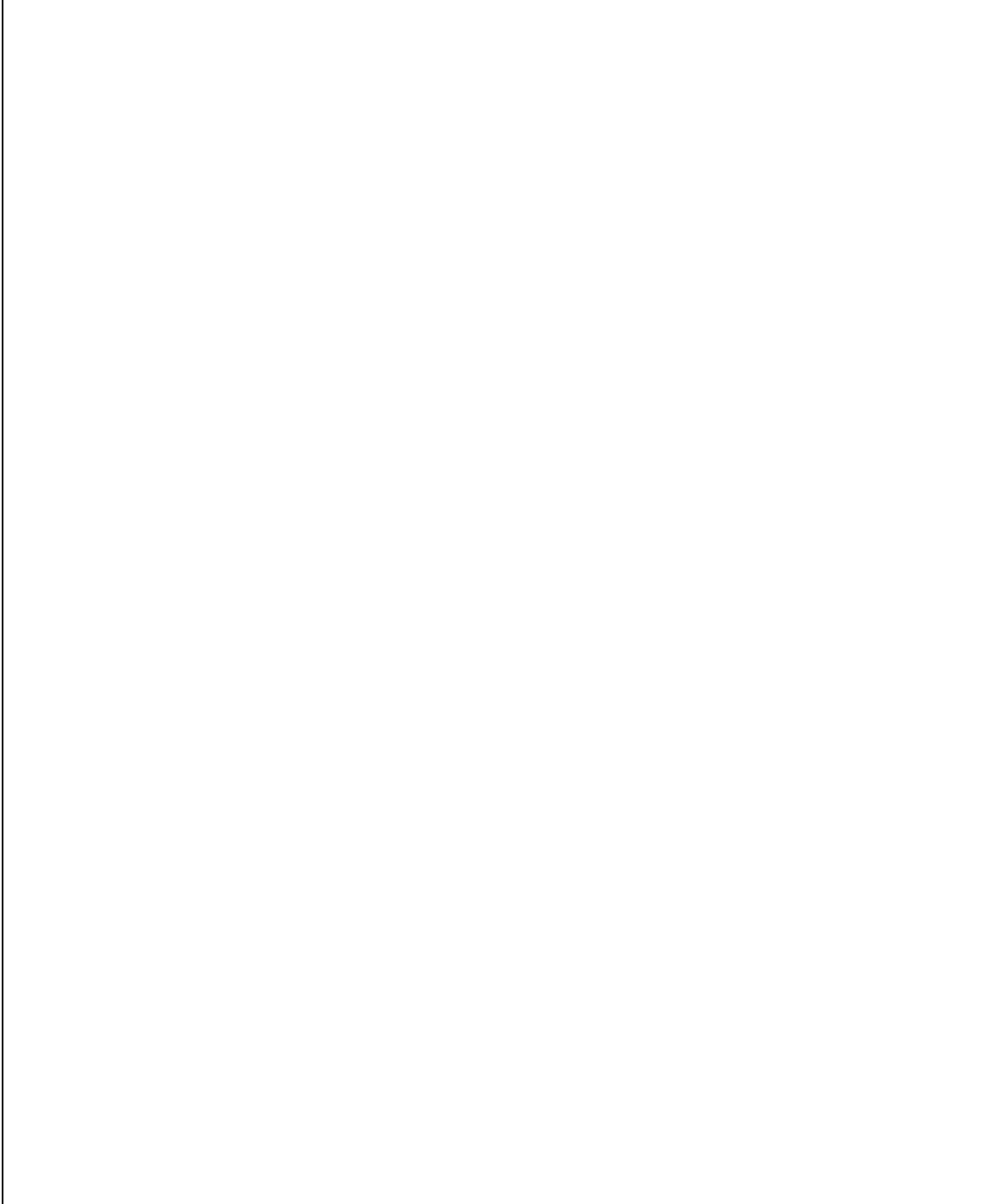
This section contains details about the following features of the *admentry* program:

- Process flow
- Parameters
- Program screens

Process Flow

Diagram

The following diagram shows the process flow in the *admentry* program.



Process Flow Description

The following describes the process flow in the *admentry* program.

1. Performs initialization and processes parameters passed to *admentry*.
2. Did the user or menuopt pass the -f parameter with a valid formtype?
 - If yes, go to step 3.
 - If no, load and display the Admentry Forms Menu screen and allow the user to enter form desired, or exit *admentry* by typing **E**.
3. Loads and displays form selected (or passed) and initialize all the record buffers for each file displayed on the form.
4. Displays the Command Panel for the desired action. This applies only if the form parameter is passed.
5. Enters Query mode The user enters an ID number or Social Security number, or types zero (0) to get into the *dufind* program. Exiting from *dufind* puts user in Add mode if permissions allow it. Otherwise, user enters Query mode again.
6. Begins entry routine. This step allows the user to either display (if display only) or add/update records based on permissions and files displayed. Query mode is a part of the entry routine.

Database Input

The *admentry* program uses the following database files as input:

- aa_rec
- aa_table
- acad_cal_rec
- acad_stat_table
- accomp_rec
- accomp_table
- addree_rec
- adm_rec
- aid_rec
- church_rec
- cl_table
- ctc_rec
- ctc_table
- cty_table
- ctry_table
- dec_table
- deg_table
- denom_table
- ed_rec
- ethnic_table
- exam_rec
- exam_table
- fac_rec
- hand_table
- id_rec
- immune_rec
- int_rec
- int_table
- intvwrecom_rec
- involve_rec
- invl_table
- major_table

- mrtl_table
- profile_rec
- prog_enr_rec
- prog_table
- ref_table
- relation_rec
- rel_table
- sch_rec
- sess_table
- site_rec
- st_table
- suffix_table
- tick_rec
- title_table
- visa_table
- zip_table

Database Output

The *admentry* program can update the following database files if the user has the appropriate permissions.

- aa_rec
- accomp_rec
- addree_rec
- adm_rec
- admhist_rec
- bus_rec
- church_rec
- ctc_rec
- disab_rec
- ed_rec
- emp_rec
- enr_stat_rec
- exam_rec
- hold_rec
- id_rec
- immune_rec
- int_rec
- intvwrecom_rec
- involve_rec
- milit_rec
- profile_rec
- prog_enr_rec
- relation_rec
- relsec_rec
- sch_rec
- site_rec
- tick_rec

Parameters

Introduction

CX contains parameters needed when running *admentry*. You can specify different parameters when loading *admentry* in a specified manner at the time of execution.

Parameter Syntax

You can display *admentry* parameters by entering the following command at your shell prompt.

admentry -,

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

```
admentry [-d] [-p prog] [-T tick] [-o ofc_added_by] [-m menuname] [-f form_selected]
[-t today] [-L site] [-P scr_path] [-a] [-F] [-M menu_title] [-D debug_level] [-S
pause_level] [-w idtype_screen] [-q]
```

Parameters

The following lists the parameters for running *admentry*.

Note: Although the C program does not specifically require any parameters to load, some parameters are required for successful processing. The following list flags the parameters that you must enter for successful processing as Required.

-d

Optional - Specifies that *admentry* should be run in Display-Only mode. Only Query commands can be used.

-p prog

Required - Overrides the default Program code defined in the program with the valid Program code specified. The default is UNDG (for undergraduate).

Example: *admentry -p WEC* (runs *admentry* with a Weekend College Program code)

-T tick

Required - Overrides the default Tickler code with the code you specify. This restricts the user to displaying and updating only Tickler records and contacts with the specified code. The default is ADM (for Admissions).

-o ofc_added_by

Optional - Overrides the default Office Added By code with one you specify. The default is ADM (for Admissions).

-m menuname

Optional - Specifies the name of the form menu screen to use when running *admentry*. The default is admmenu.

-f form_selected

Optional - Restricts the user to a specific form. If the *-f* parameter is passed with a valid form (e.g., *inq_1*), *admentry* will not bring up the Admissions Forms Menu with a list of forms to select, but will go into Query mode on the specified form.

-t today

Optional - Overrides the system's current date as the default date value.

-L site

Required - Permits a Site code to be specified. The default value is CARS.

-P *scr_path*

Optional - Specifies which screen path to use, overriding the default path of \$CARSPATH/modules/admit/progscr/admentry/.

-a

Optional - Passes a parameter to tell *admentry* to use Auto-mode.

-F

Optional - Passes a parameter to tell *admentry* to use the Force Query feature, which specifies that a query must be attempted before insert mode is permitted.

-M *menu_title*

Optional - Passes a parameter that lets you change the ring menu title to the one you specify.

-D *debug_level*

Optional - Specifies a higher debug value for printing debugging messages about the records that are being updated while *admentry* is running. The higher the value, the more messages. Options are 1, 3, 5, 7, and 9. The default debug level is zero (0).

-S *pause_level*

Optional - Lets you specify a lower pause value for pauses while *admentry* is running. The lower the pause value, the more pauses. Options are 1-9.

-w *idtype_screen*

Optional - Lets you specify a path for the ID Type window, overriding the default path of \$CARSPATH/modules/Lib/progscr/libids/type.

-q

Optional - Permits additional selection restrictions on a name query.

Program Screens

Purpose

The *admentry* program has a number of screens for performing the following interactive functions:

- Entering and updating information about inquiries.
- Entering and updating information about applicants.
- Reviewing and updating records created by the *elecapp* program.

Access

Screen files for the *admentry* program are located in the \$CARSPATH/modules/admit/progscr/admentry directory path. Common detail windows (e.g., sch_1, exam) that are used by other programs as well as *admentry* can be found in the \$CARSPATH/modules/Lib/progscr/libentry directory.

Screen Files and Table/Record Usage

The *admentry* screens appear in the following files and use the indicated tables and records.

act_1

Contains the ACT Exam Form screen. This form can be used if an admissions office chooses to manually enter information received from ACT's hard-copy examination results forms (instead of using the *tpconvert* program to load this information from a tape/diskette).

Tables/Records: adm_rec, ctry_table, cty_table, denom_table, ed_rec, ethnic_table, exam_rec, exam_table, hand_table, id_rec, major_table, profile_rec, sch_rec, sess_table, site_rec, st_table, suffix_table, title_table, zip_table

admhist

Contains the Admissions History detail window. This window is used to view Admission records for past sessions and years.

Tables/Records: admhist_rec, id_rec

admmenu

Contains the Undergraduate Admissions Form Menu screen.

Tables/Records: None

aid

Contains the Financial Aid Awards detail window. This window is used to view the types and amounts of aid awarded a student by the financial aid office.

Tables/Records: aid_rec, aid_table

bus_1

Contains the Business Form entry screen. This screen is used to enter and update information about businesses with which the admissions office interacts.

Tables/Records: bus_rec, ctry_table, id_rec, st_table, zip_table

ctc

Contains the Contact detail window used by an admissions office to track various types of communications with a student.

Tables/Records: ctc_blob, ctc_rec, ctc_table, id_rec

dec_1

Contains the Undergraduate Decision Form entry screen.

Tables/Records: adm_rec, dec_table, id_rec

gadmin

Contains the Graduate Admissions Form Menu screen.

Tables/Records: None

gdec_1

Contains the Graduate Decision Form entry screen.

Tables/Records: adm_rec, dec_table, id_rec

ginq_1

Contains the Graduate Inquiry Form entry screen. This screen is used to enter information about graduate student inquiries.

Tables/Records: acad_cal_rec, adm_rec, cty_table, ctry_table, ed_rec, ethnic_table, id_rec, major_table, ref_table, profile_rec, sch_rec, site_rec, st_table, suffix_table, tick_rec, title_table, trk_table, zip_table

glongapp_1

Contains the first page of the Graduate Application Form used to enter graduate student applications for admission.

Tables/Records: adm_rec, cl_table, ctry_table, cty_table, deg_table, ethnic_table, hand_table, id_rec, major_table, mrtl_table, priv_table, profile_rec, sess_table, site_rec, st_table, suffix_table, title_table, visa_table, zip_table

glongapp_2

Contains the second page of the Graduate Application Form used to enter graduate student applications for admission.

Tables/Records: adm_rec, ctc_rec, dec_table, id_rec, profile_rec, ref_table, tick_rec, trk_table

glongapp_3

Contains the third page of the Graduate Application Form. This display-only screen shows document contact control information.

Tables/Records: adm_rec, ctc_rec, ed_rec, id_rec

immune

Contains the Immunizations detail window used by the admissions office to view, enter, and update a student's immunization records.

Tables/Records: immune_rec, immune_table

inq_1

Contains the Undergraduate Inquiry Form entry screen. This screen is used to enter information about undergraduate student inquiries.

Tables/Records: acad_cal_rec, adm_rec, ctry_table, cty_table, ed_rec, ethnic_table, id_rec, major_table, profile_rec, ref_table, sch_rec, site_rec, st_table, suffix_table, tick_rec, title_table, trk_table, zip_table

intvwrecom

Contains the Interview/Recommendation detail window used by the admissions office to track information about an applicant's interviews and recommendations for admission.

Tables/Records: id_rec, intvwrecom_rec

longapp_1

Contains the first page of the Undergraduate Application Form used to enter undergraduate student applications for admission.

Tables/Records: cl_table, ctry_table, cty_table, deg_table, ethnic_table, hand_table, major_table, mrtl_table, priv_table, sess_table, st_table, suffix_table, title_table, visa_table, zip_table, adm_rec, id_rec, profile_rec, site_rec

longapp_2

Contains the second page of the Undergraduate Application Form used to enter undergraduate student applications for admission.

Tables/Records: adm_rec, ctc_rec, dec_table, id_rec, profile_rec, ref_table, tick_rec, trk_table

longapp_3

Contains the third page of the Undergraduate Application Form. This display-only screen shows document contact control information.

Tables/Records: adm_rec, ctc_rec, ed_rec, id_rec

ref_1

Contains the Referral Card screen. This screen can be used as an alternative to the inq_1 screen to enter information about inquiries.

Tables/Records: adm_rec, ctry_table, cty_table, ed_rec, id_rec, major_table, profile_rec, ref_table, sch_rec, sess_table, site_rec, st_table, suffix_table, title_table

sat_1

Contains the SAT Exam Form screen. This form can be used if an admissions office chooses to manually enter information received from ETS's hard-copy examination results forms (instead of using the *tpconvert* program to load this information from a tape/diskette).

Tables/Records: adm_rec, ctry_table, cty_table, denom_table, ed_rec, ethnic_table, exam_rec, exam_table, hand_table, id_rec, major_table, profile_rec, sch_rec, sess_table, site_rec, st_table, suffix_table, title_table, zip_table

shortapp_1

Contains the short Application Form screen which can be used as an alternative to the long application forms to enter information about applicants.

Tables/Records: adm_rec, ctry_table, cty_table, denom_table, ed_rec, id_rec, int_rec, int_table, major_table, mrtl_table, profile_rec, ref_table, sch_rec, sess_table, site_rec, st_table, suffix_table, title_table, zip_table

SECTION 7 - ADMISSIONS STATISTICS

Overview

Introduction

This section provides reference information about the Admissions Statistics (*admstats*) program. The Recruiting/Admissions product uses *admstats* to update the Current Enrollment Status and Previous Enrollment Status of a student in the Admissions record. The system creates an Enrollment Status record (and creates or updates an Admissions Statistics record) each time a student's enrollment status changes. This program tracks the enrollment statuses of students and produces information that can be used in admissions reporting.

The terms *Admission Status* and *Enrollment Status* are very often used interchangeably; both terms refer to a student's standing or position in the inquiry/acceptance/enrollment process (e.g., INQUIRED, APPLIED).

Program Features Detailed

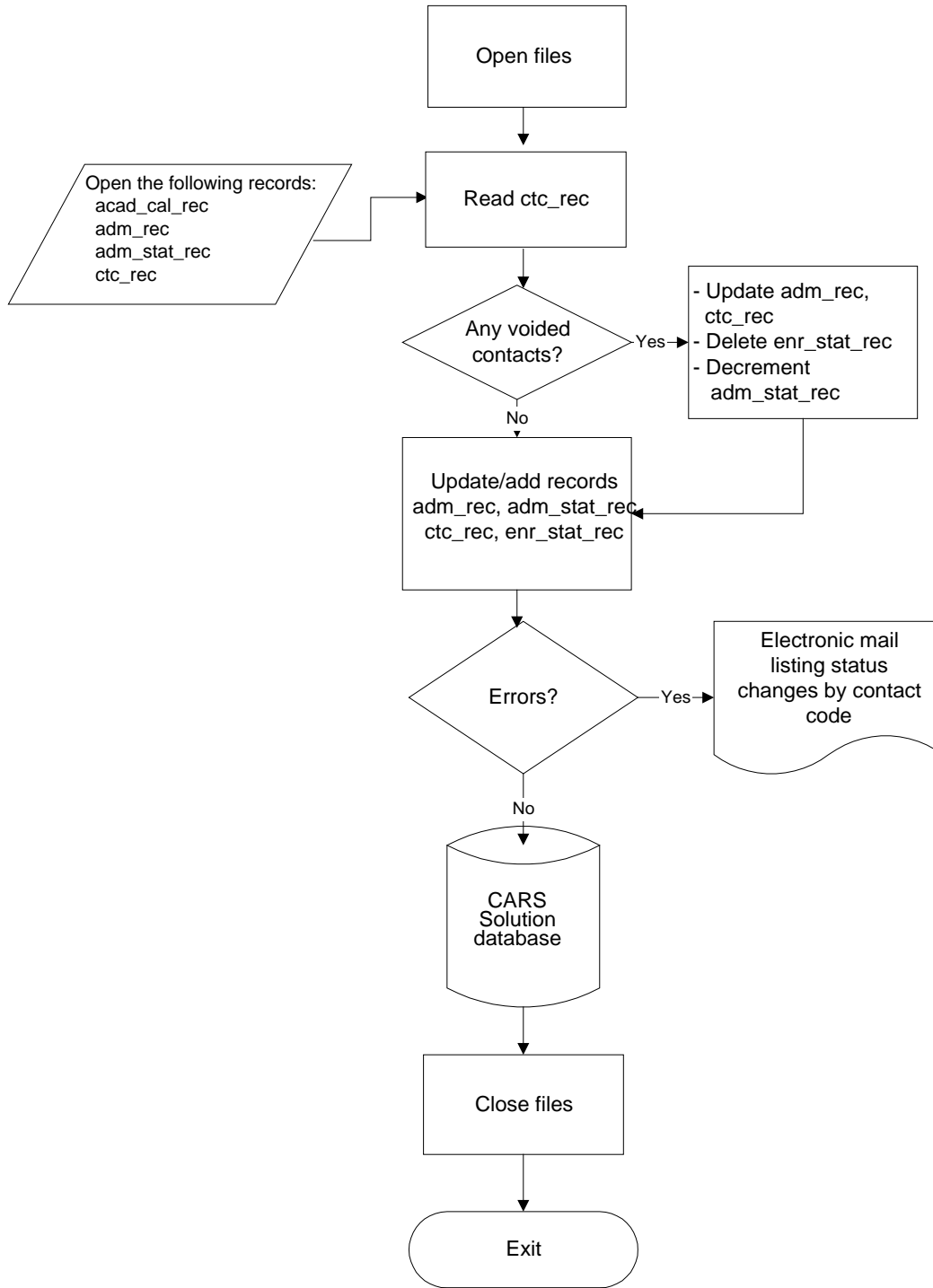
This section contains details about the following features of the *admstats* program:

- Process flow
- Parameters
- Program screens

Process Flow

Diagram

The following diagram shows the process flow in the *admstats* program.



Process Flow Description

The following describes the process flow in the *admstats* program.

1. The *admstats* program opens the necessary input files.
2. The program reads Contact records with the appropriate admissions Tickler code, storing the records for each ID number in a temporary buffer. In this buffer, the Contact records are sorted and processed in Completion Date (*ctc_rec.cmpl_date*) and Contact Number (*ctc_rec.ctc_no*) order. The program ignores Contact records that have been voided previously.
3. The program checks for any Contact records that have been voided since the last run of *admstats*. A contact is voided when a V appears in the Contact Status (*ctc_rec.stat*) field. This value can be entered using *admentry*.

The *admstats* program performs the voiding process on these Contact records by determining whether the voided contact caused an enrollment status change. If so, the program does the following:

- Deletes the Enrollment Status record (*enr_stat_rec*) that was created as a result of the change.
 - Decrements the counts in the proper Admissions Statistics records (*adm_stat_rec*) for the recorded change.
 - Places the value V into the Contact Admstat (*ctc_rec.admstat*) field.
 - Places an X into the Contact Admstat field for all contacts subsequent to the voided contact so they will be reprocessed.
 - Deletes all Enrollment Status records associated with these subsequent contacts.
 - Decrements the corresponding Admissions Statistics records.
 - Reprocesses Contact records subsequent to the first voided contact to maintain accurate admissions statistics.
4. If a voided contact did not cause an enrollment status change, or if another Contact record exists that could cause the same enrollment status change, the *admstats* voiding process voids that contact by placing a V into the Contact Admstat field. If other Contact records exist that could have caused the same enrollment status change as the contact that is being voided, *admstats* will not delete Enrollment Status records or decrement counts in the Admissions Statistics record.
 5. The *admstats* program skips contacts that have been previously completed, and processes only those that have been added since the last time the program was run. It also processes any contacts that have been marked for reprocessing after a contact has been voided. If a Contact record does not contain a valid Resource code (*ctc_rec.resrc*) as defined in the Contact table (*ctc_table*), *admstats* detects the error and sends notification to the operator via e-mail, and the contact is marked as invalid with an X in the Contact Admstat field. The next contact is then processed.
 6. The *admstats* program checks each new contact with a valid Resource code to see if it causes an enrollment status change by checking the Contact table (*ctc_table.enrstat*). If so, *admstats* determines whether the status change is valid by checking the Enrollment Sequence table. If valid, *admstats* flags the contact as complete (using a C in the Contact Admstat (*ctc_rec.admstat*) field), and updates the appropriate fields in the database. If the enrollment status change is invalid, *admstats* sends an error message to the operator's mail, and marks the Contact record as invalid with an X in the Contact Admstat field.
 7. The program updates the following fields in the Admissions record:
 - *adm_rec.enrstat* (Current enrollment status)
 - *adm_rec.cur_enr_date* (Contact Record date, *ctc_rec.cmpl_date*)
 - *adm_rec.prev_enrstat* (Previous enrollment status)
 - *adm_rec.prev_enr_date* (Date of old status, *adm_rec.cur_enr_date*)

- adm_rec.last_resrc (Last outgoing contact, where routing = 0)
- adm_rec.last_resrc_date (Date of last outgoing contact)
- adm_rec.last_resp (Last incoming contact, where routing = 1)
- adm_rec.last_resp_date (Date of last incoming contact)
- adm_rec.next_resrc (Next outgoing contact, where routing = 0)
- adm_rec.next_resrc_date (Date of next outgoing contact)
- adm_rec.next_resp (Next incoming contact, where routing = 1)
- adm_rec.next_resp_date (Date of next incoming contact)

8. The program updates the following fields in the Contact record:

- ctc_rec.admstat (Current Contact Status code)
- ctc_rec.enrstat (Current Enrollment Status after the contact is processed)

Note: The *admstats* program reviews all Contact records even if they are not being used to update the student's status (e.g., contacts used to create letters/labels or contacts used to track the arrival of documents). The *admstats* program updates the Enrollment Status field in the Contact record (ctc_rec.enrstat) with the student's current status (adm_rec.enrstat) at the time the *admstats* program reviews the Contact record. This enables you to determine what status a student had at the time a letter was sent, or when an incoming document (e.g., ESSAY), arrived in the admissions office.

The Admissions Status field in the Contact record works in conjunction with *admstats* and tells the program whether or not it needs to review the Contact record. If the ctc_rec.admstat field is not equal to the ctc_rec.stat field, the *admstats* program will review the Contact record. The valid values for ctc_rec.admstat field include:

- " " blank
- (E)xpected
- (C)ompleted
- (V)oided
- (X) Unsuccessful

When *admstats* reviews a contact with a status (ctc_rec.stat) of E, it updates the ctc_rec.admstat field to E as well. When *admstats* reviews a contact that has a status of C and a valid completion date, and successfully updates the student's admissions status, the ctc_rec.admstat field is updated to C. If a contact is voided by updating the contact status to V, the *admstats* program will update the ctc_rec.admstat field to V. If *admstats* is unsuccessful in updating the adm_rec.enrstat field, it will update the ctc_rec.admstat field with a value of X. The *admstats* program will keep trying to review and act upon this contact each time the program is run until it is successful in updating the student's admission status. At that time, it will update the ctc_rec.admstat field from X to C. The ctc_rec.admstat and ctc_rec.enrstat fields do not display on the Contact detail window.

9. The program adds a new Enrollment Status record using the following information:

- enr_stat_rec.enrstat_no (Unique serial number for each record)
- enr_stat_rec.id (ID number of student)
- enr_stat_rec.prog (Student's academic program)
- enr_stat_rec.enrstat (New enrollment status)
- enr_stat_rec.prev_code (Previous enrollment status)
- enr_stat_rec.sess (Session in adm_rec.plan_enr_sess)
- enr_stat_rec.yr (Year in adm_rec.plan_enr_yr)
- enr_stat_rec.beg_date (Contact's completion date)
- enr_stat_rec.add_tm (System's current time)

- enr_stat_rec.add_date (System's current date)
 - enr_stat_rec.tick (Contact Tickler code)
 - enr_stat_rec.init (Status change caused by Incoming or Outgoing contact)
10. The program updates running totals for enrollment statistics in the Admissions Statistics record. Statistics are based on the beginning date (adm_stat_rec.beg_date), the ending date (adm_rec.end_date), academic program, session, year, and enrollment sequence, as follows:
- adm_stat_rec.prog (Academic Program in adm_rec.prog)
 - adm_stat_rec.sess (Session in adm_rec.plan_enr_sess)
 - adm_stat_rec.yr (Year in adm_rec.plan_enr_yr)
 - adm_stat_rec.beg_date (Beginning date of week for this record)
 - adm_stat_rec.end_date (Ending date of week for this record)
 - adm_stat_rec.enrstat1 (Old enrollment status)
 - adm_stat_rec.enrstat2 (New enrollment status)
 - adm_stat_rec.cnt (Number of students changed from Status 1 to Status 2)
- Note:** Once *admstats* has verified the attempted status change is valid, an Enrollment Status record is created. Each Enrollment Status record can be viewed from the Enrollment Status detail window when using *admentry*. Enrollment Status records provide a history of all past statuses a student has had, and for which academic program, session and year. If the Contact record that was used to create an Enrollment Status record is voided, the Enrollment Status record will be deleted.
11. After all Contact records have been processed and related records updated, the files are closed and *admstats* exits.

Features of admstats

The *admstats* program offers the following features to add flexibility and minimize the chance of introducing errors on your database.

The Voiding Process

The voiding process of *admstats* allows the operator to cancel unwanted Contact records (ctc_rec). This process voids the contact in question and backs out information from any records created or updated by the voided contact. The voiding process treats the voided contact as if it had never been processed. This feature may be useful when Contact records have been entered out of order or by mistake. For example, if you accidentally denied a student's admission, you can void the DENIED contact, and add the ACCEPTED contact. To void a contact through *admentry*, enter V in the Contact Status field of a Contact record (ctc_rec.stat).

Repeatable Enrollment Statuses

The *admstats* program allows enrollment statuses to be repeated. The Enrollment Status record (enr_stat_rec) will allow duplicate enrollment statuses without producing an error. This feature enables statuses to be repeated over different sessions and years, and even in the same session and year. For example, if a student is accepted for the Fall term, then cancels and reapplies for the following Spring term, and is accepted again, two valid Enrollment Status records (enr_stat_rec) with an enrollment status of ACCEPTED will be created. Similarly, if the student is accepted for the Fall term, cancels and is re-accepted for the same term, two Enrollment Status records (enr_stat_rec) with an enrollment status of ACCEPTED will be created.

Updatable Sequences In Enrollment Sequence Table

Before *admstats* processes a Contact record, it finds the student's current enrollment status (adm_rec.enrstat), and the status the Contact record is attempting to use for the student's new status (ctc_table.enrstat). Once these two statuses are retrieved, they are compared

against the comparable entry in the Enrollment Sequence table to see if the status change is valid (enr_seq_table.upd_stat). If the Update Status field in the Enrollment Sequence table (enr_seq_table.upd_stat) is set to Y (for Yes), *admstats* considers this a valid status change, and will continue to process the contact, and produce the status change. If the Update Status field in the Enrollment Sequence table is set to N (for No), *admstats* will not create the new status, and will send an error message relating to the invalid status change.

The Enrollment Sequence table should have entries to compare every status added to the Enrollment Status table (enr_stat_table) to every other status. It is important to not only indicate which are valid status changes, but also which are invalid status changes. For example, going from INQUIRED to ACCEPTED would be an invalid status change as students must first apply before they can be accepted. Entries should also be made to the Enrollment Sequence table to make sure applicants are not accidentally moved backwards in the admissions flow (e.g., ACCEPTED to APPLIED). Therefore, if an admissions office has entered 10 different statuses listed in the Enrollment Status table, there should be 100 entries in the Enrollment Sequence table.

The following table provides an example of how the Enrollment Sequence table (enr_seq_table) might be set up.

Current Status	Next Status	Update Status
INQUIRED	INQUIRED	Y
INQUIRED	APPLIED	Y
INQUIRED	ACCEPTED	N
APPLIED	INQUIRED	N
ACCEPTED	INQUIRED	N

When the Enrollment Sequence table (enr_seq_table) uses this setup, the *admstats* program *will not* process contacts that would lead to a change in status to INQUIRED from APPLIED or ACCEPTED. Without these table entries, errors might be produced. A Y in the Update Status (enrseq_upd_stat) field indicates that an enrollment sequence is valid, and that *admstats* *will* process Contact records that cause a change of enrollment status, as indicated by the Enrollment Sequence record.

Entering Contacts on the Same Day

The *admstats* program will sort Contact records first by Completion Date (ctc_rec.cmpl_date), and then by Contact Number (ctc_rec.ctc_no). This allows any number of Contact records to be entered on the same day without restrictions. The *admstats* program will process Contact records in the order in which they were entered. Therefore, if you want to update an applicant's status immediately from INQUIRED to ACCEPTED, you must add the contacts in the logical order. You should do the following:

1. Add the contact INQUIRED.
2. Open another line to add the contact APPLIED.
3. Open another line to add the contact ACCEPTED.
4. Save the three contact entries.
5. Execute *admstats*.

Maintaining the Contact Table

The *admstats* program uses the Contact table (ctc_table.ctc and ctc_table.tick) to verify the validity of both the Resource code (ctc_rec.resrc) and the Admissions Tickler code (ctc_rec.tick) of a Contact record. This assures that *admstats* uses the correct Contact (ctc_table.ctc) in the Contact table. The *admstats* program must access the table in this manner, since the table may contain the same Resource code for different Tickler codes. For example, the contact (ctc_table.ctc) ACCEPTED may have two or more entries in the table. One entry may have a Tickler code (ctc_table.tick) of ADM, while another Contact table entry may have a Tickler code of ADMG. When *admstats* searches according to the Resource code and the Tickler code, it can find the appropriate table entry. Therefore, if

your admissions office processes both undergraduate and graduate applications, you must have two status contacts in the Contact table: one with the Tickler code of ADM and the other with the Tickler code of ADMG.

Database Input

The *admstats* program reads the following database tables and records to perform some validity checks before changing a student's status.

ctc_rec

Checks to see if any Contact records have been added or updated, therefore requiring processing.

acad_cal_rec

Checks the Academic Calendar record to verify that the student's planned enrollment program, session and year are listed. If an Academic Calendar record is not listed for the program, session and year, the *admstats* program will not process the student's contacts, and an error message will be sent. According to the logic of the *admstats* program, if there is no Academic Calendar record, then the program assumes the institution is not offering classes for that particular program, session and year, and therefore will not allow the processing of inquiries or applicants for that program, session, and year. The *admstats* program also uses the beginning date of classes listed in the Academic Calendar record (*acad_cal_rec.beg_date*) for the student's planned enrollment program, session and year, to determine which seven-day period to update in the Admissions Statistics record (*adm_stat_rec*).

Note: Admissions Statistics records contain both a Beginning Date and an Ending Date field (*adm_stat_rec.beg_date* and *adm_stat_rec.end_date*). The date ranges specified by these two fields are always seven days in length, and date ranges between records never overlap within a specific program. Admissions Statistics records allow you to easily track admissions statistics and to compare one year's numbers to another year's numbers. For example, menu users can easily generate a report of how many inquiries or applications were received during the current week of the current year, and how many inquiries or applications were received during the same week of past years. The *admstats* program calculates date ranges for Admissions Statistics records by first checking the beginning day of classes listed in the Academic Calendar record (*acad_cal_rec.beg_date*) for the program, year, and session in question. The *admstats* program will then start calculating seven-day intervals (beginning with the first day of classes) until it reaches the seven-day date range into which the contact's completion date falls (i.e., the contact used to update a student's status). If an Admissions Statistics record already exists for this date range, *admstats* updates the *adm_stat_rec.cnt* field by incrementing or decrementing the value by one (1). If an Admissions Statistics record does not exist for this date range, *admstats* will create the needed Admissions Statistics record and assign the *adm_stat_rec.cnt* field with an initial value of one (1).

To report data stored in the Admissions Statistics record, use the Status Sequences menu option on the Recruiting/Admissions: Statistical/Summary Analysis Reports menu. The Query by Form menu option Admissions Statistic can also enable menu users to query on counts for specific programs, sessions, years, and date ranges.

adm_rec

Checks to ensure the student has an Admissions record with the appropriate Program code (e.g., UNDG for undergraduate program or GRAD for graduate program). Typically, the UNDG Program codes works in conjunction with the ADM contact Tickler code, and the GRAD Program code works in conjunction with the ADMG Contact Tickler code. A link

between the program code and admissions Contact Tickler code exists in the Program table (prog_table.prog, prog_table.adm_tick).

ctc_table

Checks the Contact table to verify the Contact code is valid, and to determine what enrollment status (if any) is to be used to update the Admissions and other records.

enr_stat_table

Checks the Enrollment Status table to determine if the status listed in the ctc_table.enrstat field is a valid status.

enr_seq_table

Checks the Enrollment Sequence table to determine if the intended status change is a valid change.

Database Output

The *admstats* program updates the following database files:

- adm_rec
- adm_stat_rec
- ctc_rec
- enr_stat_rec

Locating Status Errors from admstats

Processing results are sent to the user in an e-mail message that lists status error messages, if any. Error messages sent by the *admstats* program should be checked and resolved on a daily basis. If errors are not resolved, the *admstats* program will continue to send the same error message for the contact in question each time *admstats* is run. Failure to resolve *admstats* errors on a regular basis can result in a large and unmanageable list of error messages.

Program Relationships

The *admstats* program interacts with *admentry*. If you set the value of the ENABLE_FEAT_AUTO_ADMSTATS macro to Y, then *admstats* will run automatically whenever a menu user causes a change in a student's Contact records (by adding a contact, deleting a contact, or updating the status of an existing contact (e.g., Completed to Voided)) and selects the Finish command.

Parameters

Introduction

The *admstats* program can be run automatically one student at a time when using *admentry*. The *admstats* program can also be run automatically for all students on a nightly basis via *cron*. If you do not want to wait for *cron* to run the *admstats* program overnight, you can run *admstats* using the Admissions Statistics menu option located on the Admissions Processing menu. With this menu option you can specify parameters that will affect the operation of *admstats* in a specified manner at the time of execution.

Parameter Syntax

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

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

```
admstats [-r] [-p prog] [-t tick_tick] [-i id_id] [-N]
```

Parameters

The following lists the parameters for running *admstats*.

Note: Although the C program does not specifically require any parameters to load, some parameters are required for successful processing. The following list flags the parameters that you must enter for successful processing as Required.

-r re-run

Optional - Used to rerun statistics for all Contact records that currently exist in the database for a specific Tickler code. This option is primarily used during implementation, and should not be invoked during normal operation. The -r parameter is not needed after Contact records have been voided, since the voiding process corrects the information in the database.

CAUTION: Before this parameter is passed, a setup procedure is required. It is recommended that you contact Jenzabar personnel regarding setup procedures before using the -r parameter. Otherwise, errors can occur that create inaccurate admissions statistics.

-p program code

Required - Allows a specific type of Admission record to be processed. Only those records with the specified code will be used by the program. The default is UNDG (for undergraduate).

Example: admstats -p UNDG

-t tickler code

Required - Specifies the Tickler code you want to process. The default is ADM (for undergraduate admissions).

Example: admstats -t ADM

-i id number

Required - Enables you to enter the ID number of the individual student for whom you want to run *admstats*. Alternatively, you can enter zero (0) to run *admstats* for all students with new or updated Contact records.

Example: admstats -i 123456

CAUTION: Do not use the -i option with the -r option.

-N Optional - Specifies that no mail is sent to the operator if the process is successful.

Program Screens

Purpose

Since *admstats* is a background process, it does not require any program screens. However, an Admissions Statistics PERFORM screen (located in the Admissions Query by Form menu) enables users to view the Admissions Statistics records (*adm_stat_rec*) that the *admstats* program creates and updates. The screen is view only, since the records displayed are maintained by the *admstats* program.

Access

The PERFORM screen file for *admstats* is located in the following directory path:
\$CARSPATH/modules/admit/screens/admstats

Screen File and Table/Record Usage

The Admissions Statistics PERFORM screen uses the *adm_stat_rec*.

SECTION 8 - ELECTRONIC APPLICATIONS

Overview

Introduction

This section provides reference information about the Electronic Applications (*elecapp*) program. The Recruiting/Admissions product uses *elecapp* to process applications that have been received via electronic sources such as the World Wide Web.

Electronic Application Terminology: Acceptance

Within Recruiting/Admissions, the term *acceptance* typically refers to a prospect's receiving an offer of admission to the institution's student body. However, within the *elecapp* program, *acceptance* means that an electronic application from a prospect has been received and that it has been determined to be legitimate. An application that is *accepted* within *elecapp* causes the temporary records created in the application process to be committed to the actual CX database for use within *admentry*. Acceptance into the database does *not* constitute acceptance as a student at the institution, but it does mean that the application proceeds through the admission decision process.

Electronic Application Terminology: Rejection

As with the term *acceptance*, the *elecapp* program assigns a special meaning to the word *rejection*. For the purposes of *elecapp*, *rejection* means that an electronic application from a prospect has been received, reviewed, and determined to be fictitious. When admissions office personnel reject an *elecapp* application, no records are committed to the CX database, and the application goes no further in the admission decision process.

Program Features Detailed

This section contains details about the following features of the *elecapp* program:

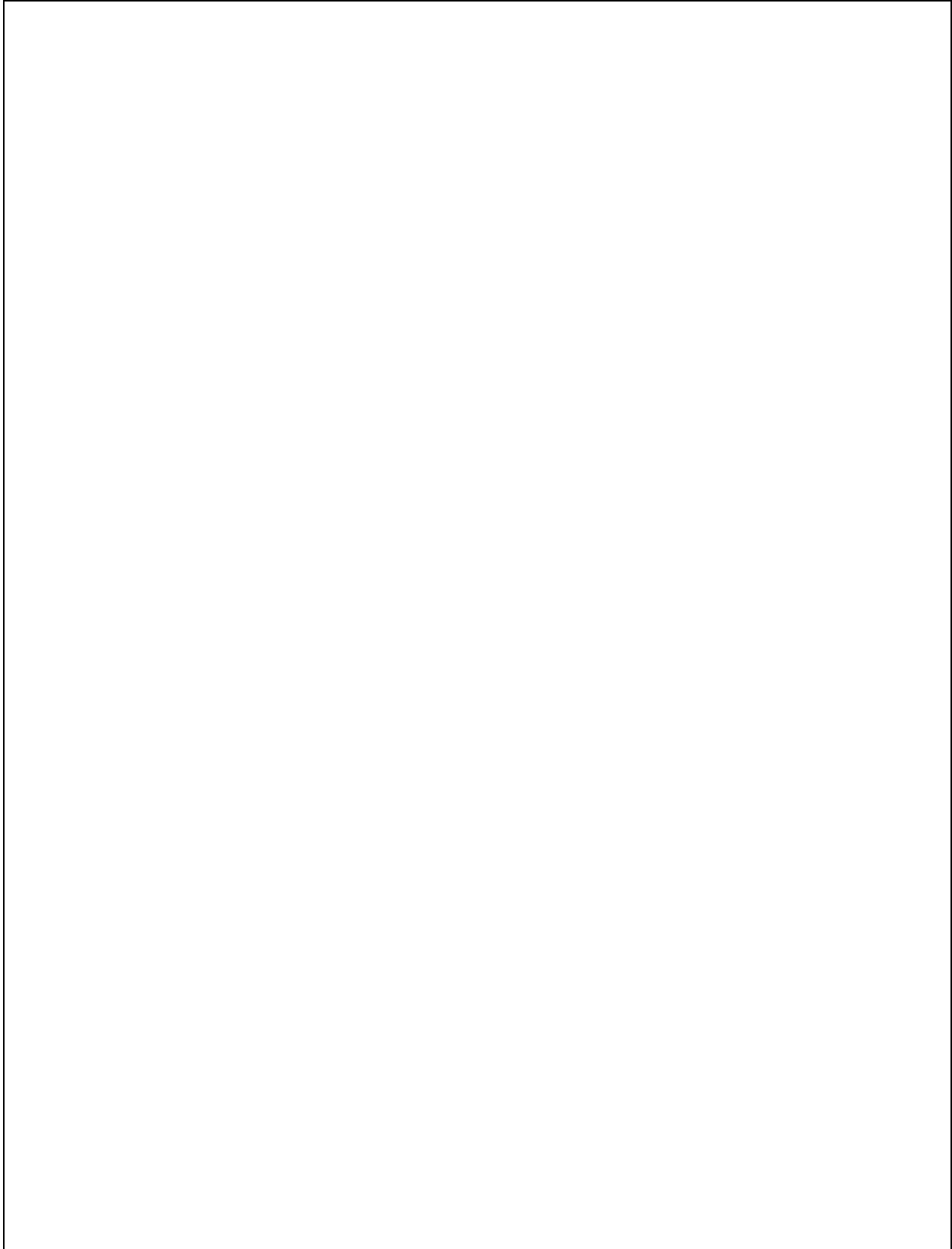
- Process flow, including diagram of *admit.cgi* script that creates input for *elecapp*
- Parameters
- Program screens

Process Flow

Diagram

The following diagram shows the process flow in the *elecapp* program. The first part of this two-part diagram shows how *elecapp* input is created, and the second part shows how *elecapp* processes the input.





Process Flow Description for the Web Application Scripts

The following describes the process flow in the Web application that provides input to the program. The Web application consists of five scripts.

1. The applicant accesses the institution's Web application and enters his/her birthdate and social security number.

Note: Foreign applicants enter their passport numbers instead of their social security numbers.

2. Does this combination of birthdate and social security number (or passport number) exist among the temporary application records maintained for *elecapp*?
 - If yes, the *admit.cgi* script displays the application if it has not been processed.
 - If no, the *admit.cgi* script displays a blank application, and the process proceeds to step 4.
3. Do the temporary records contain a *apptmp_rec.stat* of P (for partially completed)?
 - If yes, the process proceeds to step 4.
 - If no, the script returns a message stating that the institution is processing the application. The message also provides address, phone, and Web information for contacting the admissions office.
4. The applicant enters application information, and then does one of the following:
 - Submits the completed application.
 - Indicates the application is not complete, but submits the partial application.
 - Abandons the entered data.
5. Did the applicant abandon the entered data?
 - If yes, the session ends.
 - If no, go to step 6.
6. Did the application contain the necessary information to create an *apptmp_rec*?
 - If yes, the application data is stored within the temporary record files, either for updating by the student or for processing by the institution.
 - If no, the script displays a message indicating the incomplete requirements for submitting the application, and the process returns to step 4.

Scripts that Comprise the Web Application

The Web application consists of five scripts. All the scripts are in the `$CARSPATH/modules/admit/cgi` directory. The scripts and their purposes are as follows:

apply.cgi

Provides the steps by which the Web application is processed.

getrec.ph

Provides the routines required to read data from the temporary records (i.e., *apptmp_rec*, *app_edtmp_rec*, *app_accomptmp_rec*, *app_inttmp_rec*, and *app_invltmp_rec*). This script provides access to Web application information.

ldtable.ph

Provides the routines required to read data from the CX database. This script provides access to the required CX tables.

transact.ph

Provides the ability to delete and insert the temporary records within a transaction. This script protects the CX database from overwriting or destroying existing data.

validate.ph

Provides the functions used to validate data entered by an applicant. This script protects the CX database from erroneous data.

Process Flow Description for the Electronic Applications Program

The following describes the process flow in the *elecapp* program.

1. The *elecapp* program accesses the temporary records created via an electronic application process, and selects records for review based on the parameters passed by the user.
2. The user reviews the applicant's information and determines if it should be accepted or rejected.
3. Does the institution accept the applicant information?
 - If yes, go to step 4.
 - If no, go to step 11 to repeat the process for other applicants.
4. The *elecapp* program checks for matches between the data in the temporary file of *apptmp_recs*, and the *id_rec*. The program searches for matches on social security number (or passport number), comparing *apptmp_rec.ss_no* to *id_rec.ss_no*.
5. Do database records already exist for the applicant with the designated social security number?
 - If yes, the program displays the Duplicate ID Information screen. Under these conditions, the user must select one of the duplicate IDs detected, or cancel out of the *elecapp* process to resolve the duplication through other means (e.g., *admentry*). After resolution, go to step 8.
 - If no, go to step 6.
6. The program displays the Query Duplicate ID Information screen.
7. If desired, the user queries the database by name for an existing applicant for whom a social security number was not entered, and resolves any detected duplicate.

Note: Jenzabar recommends users perform this step to prevent the creation of duplicate IDs.
8. Does the user want to process the applicant's education records at this time?
 - If yes, the Applicant Education Information window appears with information about the applicant's school.

Note: An applicant can enter information about multiple schools, and for each school, an *app_edtmp_rec* is created. The *elecapp* program tracks the number of records for each applicant and indicates the count in the upper right corner of the window.
 - If no, go to step 10.
9. The *elecapp* program processes one or more Education records for the applicant by locating a valid CEEB number and school ID number for the school(s) named in the *app_edtmp_rec*, comparing *app_edtmp_rec.ceb* to *sch_rec.ceb*.
10. The user selects **Finish**, and the program updates or inserts the following records on the live CX database:
 - *admhist_rec* (Insert only)
 - *adm_rec* (Update or Insert)
 - *aa_rec* (Update or Insert)
 - *accomp_rec* (Insert only)
 - *ctc_rec* (Insert only)
 - *ed_rec* (Insert only)
 - *id_rec* (Insert only)
 - *int_rec* (Update or Insert)

- involve_rec (Insert only)
- profile_rec (Update or Insert)
- site_rec (Update or Insert)

Note: The program also inserts the id_rec.id in the apptmp_rec.id field, and the CEEB number and school ID in the app_edtmp_rec (if ed_rec information was processed).

11. Did the program select any more temporary records, based on the parameters passed?
 - If yes, go to step 2.
 - If no, the program exits.

Program Relationships

The elecapp program relates to the following programs:

- The *admentry* program displays information incorporated into the database from *elecapp*.
- The *elecapp* program will automatically execute the *admstats* program on a per ID basis (e.g., for the selected student only).

Tables and Records Used in the Program

The *elecapp* program uses a combination of common records, temporary records, and Recruiting/Admissions records as input, and updates or maintains common records and Recruiting/Admissions records.

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

Common records

The *elecapp* program creates or updates the following common records.

- aa_rec
- accomp_rec
- ctc_rec
- ed_rec
- id_rec
- int_rec
- involve_rec
- profile_rec
- site_rec

Recruiting/Admissions temporary records

The following are temporary records created or updated in the *admit.cgi* scripts or other electronic sources when an applicant submits an electronic application:

- app_aatmp_rec
- app_accomptmp_rec
- app_admtmp_rec
- apptmp_rec
- app_edtmp_rec
- app_idtmp_rec
- app_inttmp_rec
- app_invtmp_rec
- app_proftmp_rec
- app_sitetmp_rec

Note: The *elecapp* program also updates the apptmp_rec and the app_edtmp_rec.

Recruiting/Admissions records

The *elecapp* program creates or updates the following Recruiting/Admissions records, in addition to the common records listed above:

- admhist_rec
- adm_rec

Record Counts

The standard CX Web admissions application creates or updates the following quantities of each specified record:

- One adm_rec
- One ctc_rec
- One id_rec
- One profile_rec
- One site_rec
- One aa_rec
- Zero or more int_rec
- Zero or more involve_rec
- Zero or more ed_rec
- Zero or more accomp_rec
- One admhist_rec (if the application is for a later term or if the applicant is reapplying for another term)

Program Parameters

Introduction

CX contains parameters for executing the *elecapp* program. You can specify parameters to compile *elecapp* in a specified manner at the time of execution.

Parameters

The following lists the parameters for running *elecapp*.

-r *get_rejected*

Selects only previously rejected applicants.

-p *prog_code*

Selects only applicants with the specified Program code (e.g., UNDG).

-L *site_code*

Designates a Site code.

-b *begin_date*

Specifies the first date in the submitted date range for selecting applications.

-e *end_date*

Specifies the last date in the submitted date range for selecting applications.

-c *beg_time*

Specifies the beginning time based on a 24 hour clock (e.g., 9:00 a.m. = 900) in the submitted time range for selecting applications.

-d *end_time*

Specifies the ending time based on a 24 hour clock (e.g., 4:00 p.m. = 1600) in the submitted time range for selecting applications.

-a *app-source*

Enables the user to select specific applications by source of entry (e.g., Web or TS189).

-v *status*

Specifies the status of the electronic application.

-o *office added by*

Enables the user to identify the office responsible for adding information.

-t *ctc_tick_code*

Designates the Tickler code to be assigned to the initial Contact record created by *elecapp*.

-g *batch mode*

Processes *elecapp* in batch mode, as follows:

- Selects only those electronic applications with a (C)ompleted status and dates that are in the specified date range (-b and -e parameters).
- Inserts only those electronic applications that will not cause duplication, based on applicant social security number.
- Does not insert any ed_recs for the applicants.

-i *only partial applications*

Causes *elecapp* to select only those Temporary Application records with a status of Partial (i.e., apptmp_rec.stat = P).

CAUTION: Use extreme caution when passing this parameter to *elecapp*. As a general rule, your institution should only view partial applications, and not add them to your database. If you do add a partial application to your

database, the applicant will not be able to access it via the Web to complete it.

-h applications on hold

Causes elecapp to select only those Temporary Application records with a status of Hold (i.e., apptmp_rec.stat = H).

Program Screens

Purpose

The *elecapp* program has thirteen screens for viewing and processing electronic application forms.

Access

The screen files for *elecapp* are located in the following directory paths:
\$CARSPATH/modules/admit/progscr/elecapp

Screen Files and Table/Record Usage

The *elecapp* program screens appear in the following files and use the indicated tables and records.

aatmp

Contains the Applicant Other/Alternate Address Information window.

Tables/Records: app_apptmp_rec, ctry_table, st_table

accomptmp

Contains the Applicant Accomplishment Information window.

Tables/Records: app_accomptmp_rec, accomp_table

adm

Contains the Admission Information window.

Tables/Records: adm_rec, app_admtmp_rec

app

Contains the Electronic Application form.

Tables/Records: apptmp_rec, app_idtmp_rec

chg

Contains the Change Student Information window.

dupid

Contains the Duplicate IDs window.

Tables/Records: id_rec, profile_rec

ed

Contains the Applicant Education Information window.

Tables/Records: app_edtmp_rec

edtmp

Contains the Applicant Education Information window.

Tables/Records: ctry_table, deg_table, app_edtmp_rec, major_table, prog_table, st_table

inttmp

Contains the Applicant Interest Information window.

Tables/Records: app_inttmp_rec, int_table

invltmp

Contains the Applicant Involvement Information window.

Tables/Records: invl_table, app_invltmp_rec

proftmp

Contains the Applicant Profile Information window.

Tables/Records: app_proftmp_rec

qid

Contains the Query Duplicate Information window.

Tables/Records: id_rec, profile_rec

sch

Contains the School Information window.

Tables/Records: id_rec, sch_rec

SECTION 9 - CALL ENTRY

Overview

Introduction

This section provides reference information about the *calentry* program. The Recruiting/Admissions product uses *calentry* to track the efforts made by call entry personnel who contact students for your institution.

Program Features Detailed

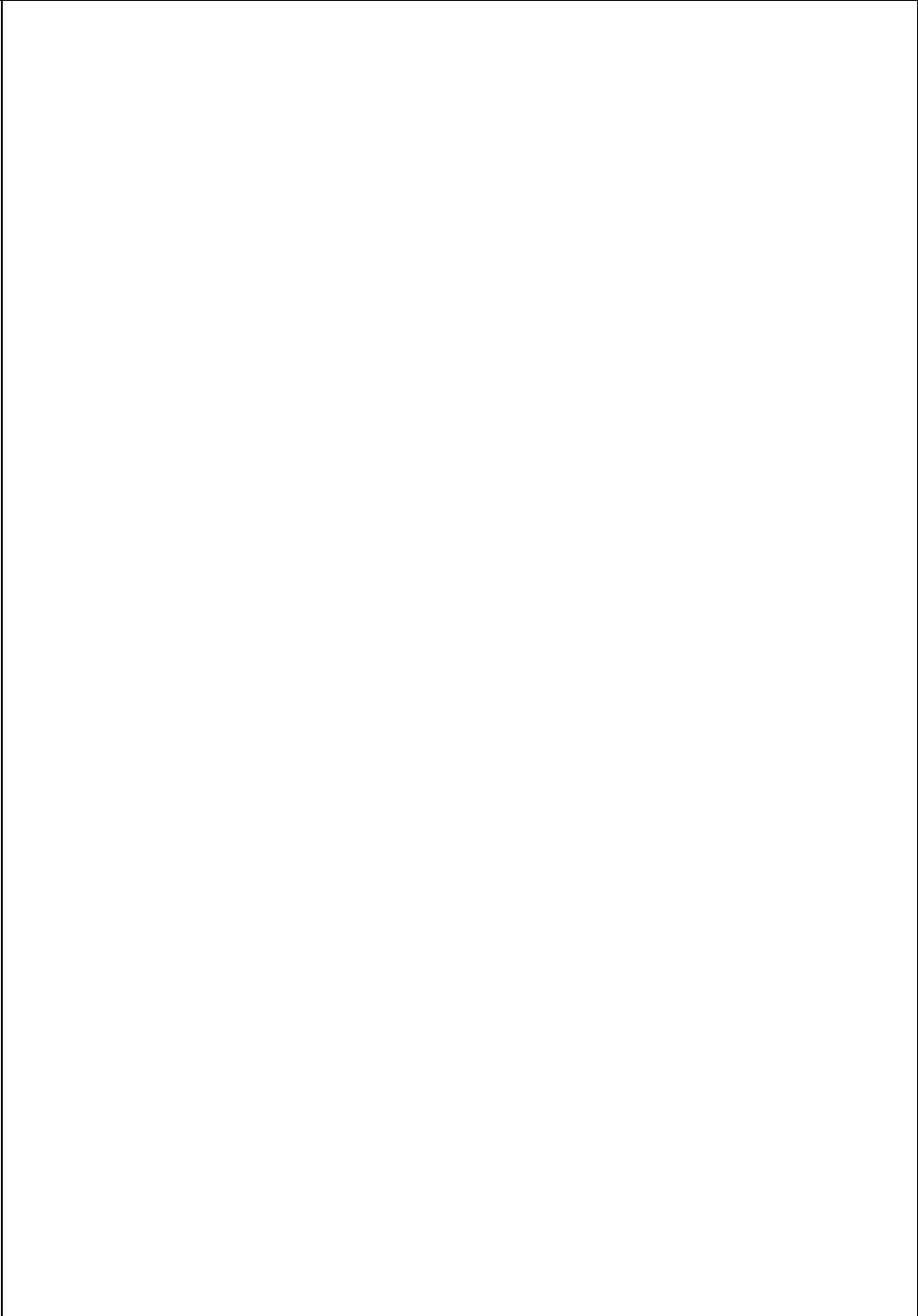
This section contains details about the following features of the *calentry* program:

- Process flow
- Parameters
- Program screens

Call Entry Process Flow

Diagram

The following diagram shows the flow of data in the *callentry* program.



Processing Flow Description

The following describes the processing flow in the *callentry* program.

1. The user enters call entry processing parameters.
2. The *callentry* program displays the Call Entry Form screen.
3. The user locates information about a prospect (either through directly entering the ID or performing a query). The screen displays all available call entry information about the prospect.
4. The user views or updates information about the prospect, based on the current contact.
5. Has the user updated information?
 - If yes, the *callentry* program updates the necessary records.
Note: For a list of records used in *callentry*, see *Tables and Records Used* in this section.
 - If no, go to step 6.
6. Does the user want to process other prospects?
 - If yes, go to step 3.
 - If no, the user exits the program.

Program Relationships

The *admentry* program (and other CX entry programs) uses data collected via the *callentry* process.

Related Processes

The *callentry* program relates to the Caller Time Entry process, a menu option that enables a caller to enter the number of hours and minutes worked. This information appears on Call Entry reports.

The *callentry* program uses the following tables and records.

- aa_rec
- aa_table
- accomp_rec
- accomp_table
- addree_rec
- adm_rec
- admhist_rec
- callresult_rec
- callresult_table
- callstu_rec
- cl_table
- ctc_rec
- ctc_table
- ctry_table
- dec_table
- deg_table
- ed_rec
- enr_stat_rec
- ethnic_table
- exam_rec
- exam_table
- hand_table

- hold_rec
- hold_table
- id_rec
- int_rec
- int_table
- invl_table
- involve_rec
- major_table
- ofc_table
- profile_rec
- prog_table
- ref_table
- rel_table
- relation_rec
- resec_rec
- sess_table
- site_rec
- st_table
- suffix_table
- tick_rec
- tick_table
- title_table
- zip_table

Call Entry Parameters

Introduction

CX contains parameters for executing *callentry*. You can specify parameters to compile *callentry* in a specified manner at the time of execution.

Parameter Syntax

You can display *callentry* parameters by entering the following: **callentry -**. The following is the correct usage for running the *callentry* program from the UNIX shell:

```
callentry [-d] [-p prog] [-T tick] [-o ofc_added_by] [-m menuname] [-f form_selected] [-t today] [-L site] [-P scr_path] [-a] [-F] [-M menu_title] [-D debug_level] [-S pause_level] [-w idtype_screen] [-q]
```

Parameters

The following lists the parameters for running *callentry*.

-d

Optional - Specifies that *callentry* should be run in Display-Only mode (i.e., *callentry* recognizes and responds only to Query commands).

-p prog

Optional - Overrides the default Program code defined in the program with the valid Program code specified. The default is UNDG (for undergraduate).

-T tick

Optional - Overrides the default Tickler code with the code you specify. This restricts the user to displaying and updating only Tickler records and contacts with the specified code. The default is ADM (for Admissions).

-o ofc_added_by

Optional - Overrides the default Office Added By code with one you specify. The default is ADM (for Admissions).

-f form_selected

Optional - Restricts the user to a specific form. If the **-f** parameter is passed with a valid form, *callentry* will not use a menu, but will display the specified form.

-t today

Optional - Overrides the default value of today for the effective date for changes.

-L site

Optional - Permits a Site code to be specified. The default is CARS.

-m menuname

Optional - Specifies the name of the form menu screen to use when running *callentry*. The default is callmenu.

-P scr_path

Optional - Specifies which screen path to use, overriding the default path of admit/callentry/.

-a

Optional - Passes a parameter to *callentry* that lets you automatically enter Query mode.

-F

Optional - Requires the user to attempt a query before being able to enter Insert mode.

-M menu_title

Optional - Lets you specify an alternate ring menu title, overriding the default.

-D *debug_level*

Optional - Lets you specify a higher debug value for printing debugging messages about the tables and files that are being updated while *callentry* is running. The higher the value, the more messages available for display. Options are 1, 3, 5, 7, and 9.

-S *pause_level*

Optional - Lets you specify a lower pause value for pauses while *callentry* is running. The lower the pause value, the more pauses. Options are 1-9.

-w *idtype_screen*

Optional - Lets you specify a path for the ID Type window, overriding the default path of Lib/libids/type/.

-q

Optional - Permits additional selection restrictions on a name query.

Program Screens

Purpose

The *callentry* program uses several screens for performing the following functions:

- Entering/updating call entry information
- Viewing call information (call entry results) by caller
- Viewing admissions information for previous years or sessions

Access

The screen files for *callentry* are located in the following directory path:

\$CARSPATH/modules/admit/progscr/callentry

Screen Files and Table/Record Usage

The *callentry* screens appear in the following files and use the indicated tables and records.

admhist

Contains the Prospect/Applicant History screen. This screen is maintained by the *admentry* program, and is not intended for updating in the *callentry* program.

Tables/Records: admhist_rec, id_rec

callent_1

Contains the entry screen for the Call Entry form.

Tables/Records: adm_rec, callstu_rec, cl_table, ctc_rec, ctc_table, ctry_table, ed_rec, ethnic_table, hand_table, id_rec, major_table, profile_rec, ref_table, sess_table, st_table, suffix_table, title_table, zip_table

callent_2

Contains the Document Status Information screen.

Tables/Records: adm_rec, ctc_rec, dec_table, ed_rec, id_rec

callres

Contains the Call Entry Results screen.

Tables/Records: callresult_rec, callresult_table, ctc_table, id_rec

ctc

Contains a special Contacts screen used for admissions. This screen contains a Time field in the contact record that can be used to schedule admission appointments for counselors and staff.

Tables/Records: ctc_rec, ctc_table, id_rec

SECTION 10 - EVENT ENTRY

Overview

Introduction

This section provides reference information about the Event Entry program (also known as Schedule Entry). The Recruiting/Admissions product uses Event Entry to enter and maintain information pertaining to an activity schedule.

Program Features Detailed

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

- Process flow
- Parameters
- Program screens

Process Flow

Diagram

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

Overview

Introduction

This section provides reference information about the Event Entry program (also known as Schedule Entry). The Recruiting/Admissions product uses Event Entry to enter and maintain information pertaining to an activity schedule.

Program Features Detailed

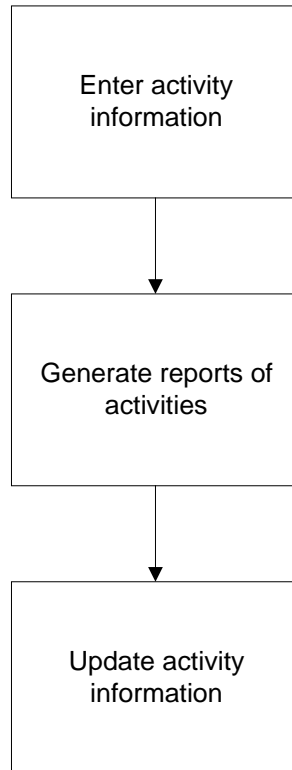
This section contains details about the following features of the Event Entry program:

- Process flow
- Parameters
- Program screens

Process Flow

Diagram

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



Data Flow Description

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

1. The user enters activity information using Event Entry screens.
2. The user obtains reports of scheduled activities using Event Entry report menu options.
3. The user updates event information as necessary.

Program Relationships

The following programs use Event Entry.

- *libentry*

Tables and Records Used

The *evntentry* program uses the following Common and Event Entry tables and records.

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

Common tables and records

- *aa_table*
- *ctry_table*

- cty_table
- ethnic_table
- id_rec
- profile_rec
- schd_rec
- st_table
- title_table
- zip_table

Event Entry tables and records

- None

Parameters

Introduction

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

Note: You can also specify compilation values with the includes for the Recruiting/Admissions product that affect the Event Entry program.

Parameter Syntax

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

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

```
evntentry [-d] [-m menuname] [-o ofc_added_by] [-f form_selected] [-t today] [-P  
scr_path] [-a] [-F] [-T menu_title] [-D debug_level] [-S pause_level] [-q] [-w  
idtype_screen]
```

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

Parameters

The following lists the parameters for running Event Entry.

-d

Optional - Specifies access to Event Entry in display-only mode.

-m *menuname*

Optional - Specifies the name of the menu screen you want to access directly, where *menuname* represents the menu name. The default value in Event Entry is *evmenu*.

Example: evntentry -m evmenu

-o *ofc_added_by*

Optional - Specifies the Office code for the office that is adding or updating the information. The default is NOFC.

Example: evntentry -o ADM

-f *form_selected*

Optional - Specifies the name of the form you want to access directly, where *form_selected* represents the form name.

Example: evntentry -f xxxx

-t *today*

Optional - Specifies the effective date for changes to data, where *today* represents the data you specify.

Example: evntentry -t 09/01/2000

-P *scr_path*

Optional - Specifies the path for the Event Entry screens, where *scr_path* represents the path where the Event Entry screens reside on your system. The default in Event Entry is *common/progscr/evntentry/*.

-a

Optional - Specifies that Event Entry will automatically enter Query mode.

-F

Optional - Specifies that Event Entry will force you to query the database for a record before you can enter Insert mode.

-T *menu_title*

Optional - Specifies a change to the ring menu title, where *menu_title* represents the title you specify.

Example: evtentry -T Event Entry

Note: This parameter applies only to character mode ring menus.

-D *debug_level*

Optional - Specifies the level of debug messages to be displayed, where *debug_level* represents the message level (1-9). To receive more messages, you must specify a higher level.

-S *pause_level*

Optional - Specifies the level of pauses for Event Entry, where *pause_level* represents the pause level (1-9). To have more pauses, you must specify a lower number.

-q

Optional - Specifies that you can select additional restrictions for querying an event's record. When you select the ID Lookup command in Query mode, you can specify that the search for an event's record be based on one of the following records:

- id_rec
- site_rec

-W *idtype_screen*

Optional - Specifies the path for the ID-Type window, where *idtype_screen* represents the path and filename of the ID-Type window. The default in Event Entry is Lib/libids/type.

Program Screens and Windows

Introduction

Event Entry has screens and windows for performing the following interactive functions:

- Entering or updating information pertaining to an event, including a description, location, sponsor, and scheduled date and time
- Entering report parameters to produce reports of event information

Access

The screen and window files for Event Entry are located in the following directory path:

- \$CARSPATH/modules/common/progscr/evntentry

Note: You can access windows from each program screen in Event Entry.

See the *CX System Reference Technical Manual* for information about common windows that appear in Event Entry.

Screen Files and Table/Record Usage

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

evntmenu

Contains the Event Entry - Schedule Entry Parameters screen.

Access: \$CARSPATH/modules/common/progscr/evntentry

Tables/Records: None

schd

Contains the Event Entry - Schedule Activities window.

Access: \$CARSPATH/modules/common/progscr/evntentry

Tables/Records:

- id_rec
- schd_rec

schedule_1

Contains the Event Entry - Schedule Entry screen.

Access: \$CARSPATH/modules/common/progscr/evntentry

Tables/Records:

- aa_table
- ctry_table
- cty_table
- ethnic_table
- id_rec
- profile_rec
- st_table
- title_table
- zip_table

SECTION 11 - LEAD ENTRY

Overview

Introduction

This section provides reference information about the *leadent* program. The Recruiting/Admissions product uses *leadent* to manage Lead records, Lead Tickler records, and Lead Contact records.

Program Features Detailed

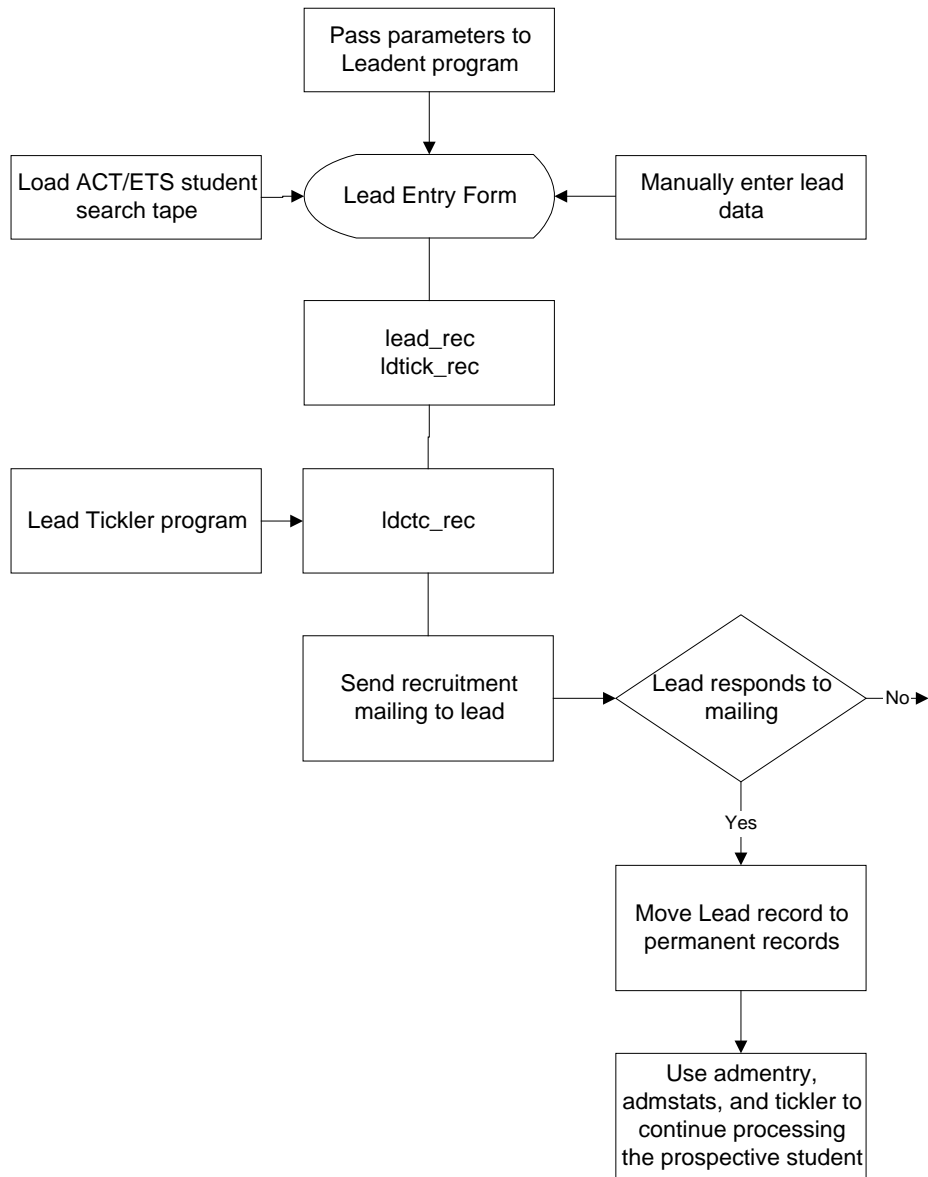
This section contains details about the following features of the *leadent* program:

- Process flow
- Parameters`
- Program screens
- Other lead record fields
- Lead entry parameters screen
- Program features

Lead Entry Process Flow

Diagram

The following diagram shows the flow of data in the *leadent* program.



Processing Flow Description

The following describes the processing flow in the *leadent* program.

1. The user enters Lead Entry processing parameters.
2. The Leadent program displays the Lead Entry Form screen.
3. The user locates data about an existing prospect (either through directly entering the ID or performing a query) or enters data for a new prospect. The screen displays all available lead entry information about the lead.
4. The user views or updates information about the lead, based on the current data.
5. Has the user updated information?
 - If yes, the *leadent* program updates the necessary records.
Note: For a list of records used in *leadent*, see *Tables and Records Used* in this section.
 - If no, go to step 6.
6. Does the user want to process other prospects?
 - If yes, go to step 3.
 - If no, the user exits the program.

Program Relationships

The *leadent* program uses the libleads library to query on Lead records. The *ldsndxinit* program populates the lead_rec.name_sndx field which makes phonetic name queries possible.

Related Processes

The *leadent* program uses the following tables and records.

- acad_cal_rec
- ctc_table
- ctry_table
- ethnic_table
- id_rec
- lead_rec
- ldctc_rec
- ldtick_rec
- major_table
- ofc_table
- prog_table
- ref_table
- sch_rec
- sess_table
- st_table
- tick_table
- title_table
- trk_table
- userid_table

Note: You must add the UNIX ID number and UNIX login name of each Lead Entry end-user to the User ID table.

- zip_table

Lead Entry Parameters

Introduction

CX contains parameters for executing the *leadent* program. You can specify parameters to compile *leadent* in a specified manner at the time of execution.

Parameter Syntax

You can display *leadent* parameters by entering the following command from your shell prompt: **leadent -.** This command will generate an e-mail listing the parameters used by the *leadent* program. The following is the correct usage for running the *leadent* program from the UNIX shell:

```
[-r adm_resrc] [-T adm_tick] [-t lead_tick] [-o lead_ofc_add_by] [-p adm_prog]
```

Parameters

The following lists the parameters for running *leadent*.

- r** Required - Defines the default Admissions Contact Resource code.
- T** Required - Defines the default Admissions Tickler code.
- t** Required - Defines the default Lead Tickler code.
- o** Required - Defines the default Office Added By code.
- p** Required - Defines the default Program code.

Program Screens

Purpose

The *leadent* program uses several screens for performing the following functions:

- View and update lead records that were loaded into the database via the data conversion process of an ACT or ETS student search electronic file/diskette.
- Manually add lead records to the database. The Lead Entry program will first require you to query the existing lead records before you can go into Insert Mode. This is done to help eliminate the creation of duplicate lead records. You can query by lead ID number or lead social security number, or by lead name.
- Add and update lead tickler records for leads.
- Add and update lead contact records for leads using the Lead Contacts detail window.
- Delete Lead records one at a time using the Lead Entry program if delete permissions have been granted to the end-user. When the Delete command is selected for an individual lead record, any lead contact records and/or lead tickler record for the lead will be deleted at the same time.
- Move/copy individual Lead records over to permanent records.

You can use the Lead Entry program in four basic modes:

- Command mode
- Query mode
- Insert mode
- Update mode

Access

The screen files for *leadent* are located in the following directory path:
\$CARSPATH/modules/admit/progscr/leadent

Screen Files and Table/Record Usage

The *leadent* screens appear in the following files and use the indicated tables and records.

ldctc

Contains the Lead Contact Detail window.

lead

Contains the Lead Entry screen.

qid

Contains the Query Duplicate Information window.

Other Lead Record Fields

Lead Record Fields Not Appearing on Lead Form

Although these database fields of the Lead record do not appear on the Lead Form, they can be added by the person who maintains CX at your institution, if it is felt they are needed.

lead_rec.add_uid

The UNIX identification number of the data entry person who added the Lead record to the database. The value in this field determines whose login name will appear on the Lead Form.

lead_rec.upd_uid

The UNIX identification number of the data entry person who last updated the Lead record.

lead_rec.valid

A logical Yes/No flag indicating if the Lead record is valid. The default value is Y. If a Lead record is determined to be invalid for some reason, (i.e., a duplicate lead record) the lead_rec.valid field can be updated with a value of N. Once a Lead record has been flagged as not being a valid Lead record, it does not appear in the Lead List screen when you perform a query of Lead records by name. The other alternative is to simply delete the invalid Lead record from the database.

lead_rec.ss_major1

The numeric code used by ACT or ETS (American College Testing or Educational Testing Services, respectfully), to indicate the lead's first choice of academic major. This field is populated when loading a Student Search tape/diskette purchased from ACT or ETS.

lead_rec.ss_major2

The numeric code used by ACT or ETS (American College Testing or Educational Testing Services, respectfully), to indicate the lead's second choice of academic major. This field is populated when loading a Student Search tape/diskette purchased from ACT or ETS.

lead_rec.name_sndx

A field populated by the ldsndxinit program to enable the ability of phonetic name queries of Lead records.

ldtick_rec.next_rvw_date

A date field telling the Lead Tickler program the next time it should review a student's Lead Contact records. This field is updated by the Leadent program, the Lead tickler program, and by the letter creation process, (ltrlead ace report).

ldtick_rec.last_rvw_date

A date field indicating the last time the Lead Tickler program reviewed a student's Lead Contact records.

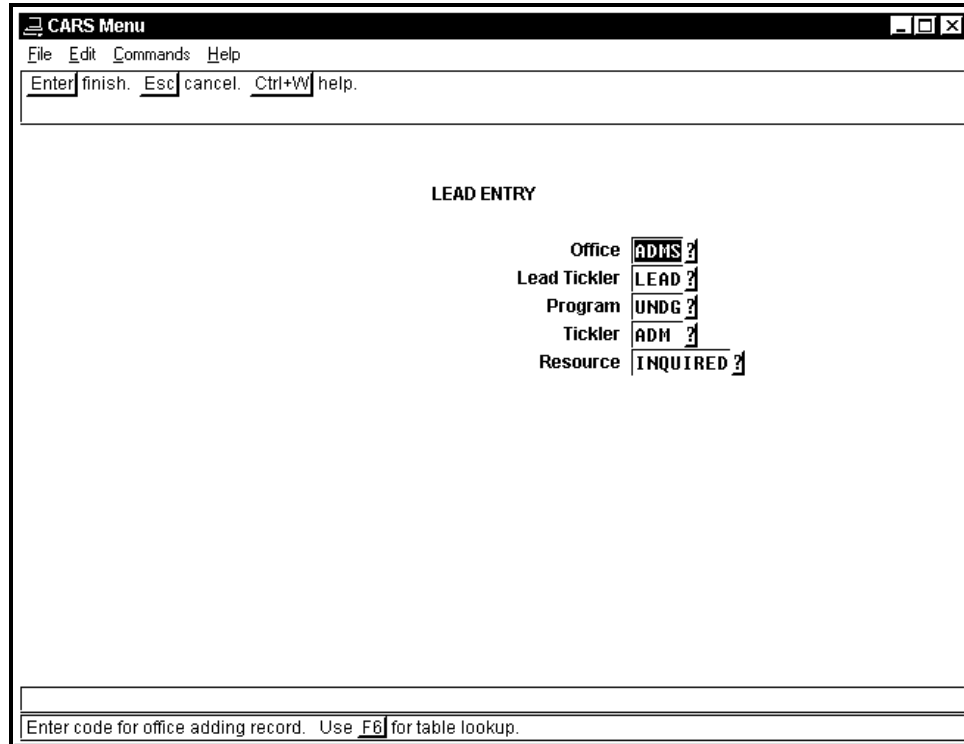
Lead Entry Parameters Screen

Purpose

The Lead Entry parameters screen enables you to pass variable settings to the Lead Entry program.

Example

Following is an example of the Lead Entry parameter screen.



Parameters Passed

The following lists the parameters that must be passed to the Lead Entry program.

Office (-o)

The Office code used to indicate the office where the end-user works, (i.e., ADMS). Use **Table Lookup** for a list of valid Office codes.

Lead Tickler (-t)

The Lead Tickler code that is assigned to any Lead Contact records created by the Lead Entry program, (i.e., LEAD). Use **Table Lookup** for a list of valid Lead Tickler codes.

Program (-p)

The Program code that is assigned to any Admission records when the Move command is used in the Lead Entry program, (i.e., UNDG for undergraduate students, GRAD for graduate students). Use **Table Lookup** for a list of valid Program codes.

CAUTION: If you have passed parameters to the Lead Entry program to process undergraduate leads, and then use the Move command for a graduate lead (i.e., where program code is GRAD), the lead will not be moved over to permanent

records. Therefore, if you are using Lead Entry to process both undergraduate and graduate leads, it is very important to process all undergraduate leads first, and then exit the Lead Entry program. Reload the Lead Entry program again, this time passing the parameters to process graduate leads. Do not process both undergraduate and graduate leads using the same program and tickler code parameters.

Tickler (-T)

The Tickler code that is assigned to the initial Contact record created by the moveleads informer when the Move command is used in the Lead Entry program, (i.e., ADM). Use **Table Lookup** for a list of valid Tickler codes.

Resource (-r)

The name of the initial Contact record created by the moveleads informer when the Move command is used in the Lead Entry program, (i.e., INQUIRED). Use **Table Lookup** for a list of valid Contact Resource codes.

Lead Entry Program Features

Introduction

The following lists the features of the Lead Entry program.

Data Consistency & Keystroke Saving Features

When you enter a gender specific title, the Lead Entry program automatically populates the Sex field (e.g., MR = M, MS = F, MISS = F, MRS = F, etc.). This feature works only when the Title table is properly populated.

When you enter a zip code, the Lead Entry program automatically populates the City and State fields. This feature works only when the Zip Code table is properly populated.

A planned year of enrollment cannot be entered for a lead if that enrollment year does not exist in the Academic Calendar record. This will help eliminate admstat errors should the lead be moved over to an admissions record. If this situation occurs, the data entry person will need to ask the Registrar's office to create an Academic Calendar record for the program/session/year in question before he/she can continue to enter the lead.

A completion date for a lead tickler record cannot be entered for a lead if an Academic Calendar record does not exist for the lead's planned session and year of enrollment. If this situation occurs, the data entry person will need to ask the Registrar's office to create an Academic Calendar record for the program/session/year in question before he/she can continue to enter the lead.

When you enter a CX ID number for the lead's school of attendance, the Lead Entry program automatically populates the CEEB (College Entrance Examination Board) Number field. Conversely, when you enter the school's CEEB number first, the Lead Entry program automatically populates the School ID field. If you do not know either the CX ID number or the CEEB number, perform a query by school name.

The moveleads informer will not move a lead record if there is no Academic Calendar record for the lead's planned program/session/year of enrollment. This is done to prevent admstat errors.

When you use the Move Leads command, CX searches the database for the admissions counselor in charge of recruiting student's from the lead's school, and automatically assigns that admissions counselor to the lead's new Admissions record.

When you use the Move Leads command, Lead Entry deactivates a student's Lead Tickler record and does not schedule any more Lead Contact records. The system also updates any existing Lead Contact records with a status of E (Expected) to a status of V (Voided).

When you update a student's Lead Tickler record; or insert, delete, or update a student's Lead Contact record, the Lead Entry program automatically updates the Next Review Date of a student's Lead Tickler record to the current date.

Data Tracking Features

When you manually enter data in a Lead record, Lead Entry records the user's UNIX ID number and displays the user's login name (if the User ID table is properly populated) and office location on the Lead Entry screen. This allows the Admissions office to identify the user responsible for the information entered in each Lead record. If a duplicate Lead record is created, the Admissions office can notify the user to correct the record.

The Lead Entry screen displays the date the user moved the lead to an ID record and the permanent CX ID number for the moved lead.

The UNIX ID number of the last person to update a Lead record is recorded for each Lead record that is manually updated. This information does not appear on the data entry screen.

The date a lead was moved over to an ID record is displayed on the Lead Entry screen. The permanent CX ID number for the moved lead is also displayed on the Lead Entry screen.

When exiting the program, Lead Entry sends e-mail to the user detailing the database transactions that occurred.

Duplicate Checking Features

Use the Lead Entry program to move individual Lead records to permanent records by selecting the Move command, a Query Duplicate Information screen appears. This screen enables you to perform an initial ID query by the lead's name. In this way you can verify that the lead does not have an existing ID record and eliminate the possibility of a duplicate ID record.

The Lead Entry program does not move a Lead record to an ID record if it finds a match by social security number.

Lead Tickler Program

To learn how to use the Lead Tickler program, see the documentation regarding the original Tickler program in the *Communications Management User Guide*, and the *Communications Management Technical Manual*. The set up and logic behind the Tickler and Lead Tickler programs are exactly the same. Only the records used by the two programs are different. The differences between the two programs are listed below.

Lead Tickler uses the lead record (`lead_rec`), whereas the Tickler program uses the ID record (`id_rec`).

Lead Tickler uses the lead tickler record (`ldtick_rec`), whereas the Tickler program uses the tickler record (`tick_rec`).

Lead Tickler uses the lead contact record (`ldctc_rec`), whereas the Tickler program uses the contact record (`ctc_rec`).

The Lead Tickler program is located in the `$CARSPATH/src/commgmt/ldtickler` directory.

Parameter Syntax

You can display *ldtickler* parameters by entering the following: **`ldtickler -.`**. The following is the correct usage for running the *ldtickler* program from the UNIX shell:

`[-t tick] [-d today] [-r review only] [-o dump only] [-a allow alteration of date for setup/testing]`

Parameters

The following lists the parameters for running *ldtickler*.

- d** today's date
- t** lead tickler code
- r** review lead tickler and lead contact records in batch (instead of interactively one student at a time)
- o** dump only
- a** allow alteration of date for setup and testing

Parameters used by the *ldtickler* program:

Lead Soundex Program

The Lead Soundex program is located in the \$CARSPATH/src/common/ldsndxinit directory. This program updates the lead_rec.name_sndx field. Once the name_sndx field is updated in the lead record the capability of doing phonetic name queries is possible, using the Libleads and Leadent programs. This program is run automatically each night when CRON runs the admprocess script.

Parameter Syntax

You can display *ldsndxinit* parameters by entering the following: **ldsndxinit -.** The following is the correct usage for running the *ldsndxinit* program from the UNIX shell:

[-u] [-q]

-u

Update all lead_rec.name_sndx fields according the value in lead_rec.fullname field.

-q

Do not send mail regarding the specific LEAD numbers effected.

SECTION 12 - MENUS, SCREENS, SCRIPTS, AND REPORTS

Overview

Introduction

This section provides reference information on the following features of the Recruiting/Admissions product:

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

Directory Locations

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

Menu source files

\$CARSPATH/menusrc/admit
\$CARSPATH/menusrc/admit/callentry
\$CARSPATH/menusrc/admit/callentry/reports
\$CARSPATH/menusrc/admit/commgmt
\$CARSPATH/menusrc/admit/elecapp
\$CARSPATH/menusrc/admit/elecapp/reports
\$CARSPATH/menusrc/admit/leads
\$CARSPATH/menusrc/admit/leads/commgmt
\$CARSPATH/menusrc/admit/leads/commgmt/ldtickler
\$CARSPATH/menusrc/admit/leads/leadrpts
\$CARSPATH/menusrc/admit/query
\$CARSPATH/menusrc/admit/reports
\$CARSPATH/menusrc/admit/reports/cnslrrpts
\$CARSPATH/menusrc/admit/reports/ctcrpts
\$CARSPATH/menusrc/admit/reports/profrpts
\$CARSPATH/menusrc/admit/reports/statrpts
\$CARSPATH/menusrc/admit/screens
\$CARSPATH/menusrc/admit/session
\$CARSPATH/menusrc/admit/tables
\$CARSPATH/menusrc/admit/tpconv

Menu option files

\$CARSPATH/menuopt/admit
\$CARSPATH/menuopt/admit/informers
\$CARSPATH/menuopt/admit/others
\$CARSPATH/menuopt/admit/programs
\$CARSPATH/menuopt/admit/reports
\$CARSPATH/menuopt/admit/screens
\$CARSPATH/menuopt/admit/scripts

PERFORM screens

\$CARSPATH/modules/admit/screens
\$CARSPATH/modules/common/screens

SQL statements

\$CARSPATH/modules/admit/informers
\$CARSPATH/modules/regist/informers

ACE reports

\$CARSPATH/modules/common/reports
\$CARSPATH/modules/admit/reports
\$CARSPATH/modules/admit/others

Communications Management Menu Options

Because letter creation and production is an important aspect of Recruiting/Admissions, access to Communications Management is available from a variety of Recruiting/Admissions menus. For more information about Communications Management menu options, see *Communications Management Technical Manual*.

Menu Structure

Introduction

The CX menu source (menusr) directory path contains definitions of the CX menu structure. Specifically, the \$CARSPATH/menusr/admit directory path contains definitions for all the Recruiting/Admissions menus.

Menu Options

Introduction

The CX menu structure is defined in the menu source (menusr) directory path. Specifically, the Recruiting/Admissions Main Menu is described in the \$CARSPATH/menusr/admit directory. The purpose of this subsection is to provide you with reference information about the processes referenced in each menusr file in the standard CX Recruiting/Admissions product.

Note: Refer to the preceding flowcharts for information about the macro definitions required to make certain optional menu options available.

Programs, Screens, Scripts, and Reports

The following is a list of Recruiting/Admissions menu options, the name of the actual process (program, screen, script, or report) each option executes, the name and location of the menu option file, and the parameters passed by the menu option. The access information is the location of the source information that, once installed, is in the installed path (e.g., \$BINPATH (programs), \$ARCPATH (ACE reports), \$INFPATH (SQL statements), \$OTHPATH (reports and some scripts), \$SCPPATH (C shell scripts), and \$FRMPATH (PERFORM screens)).

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.

Admissions Processing menu

Admissions/Service Entry

Accesses: \$CARSPATH/src/matric/mtentry (Program)

Menuopt File: \$CARSPATH/menuopt/matric/programs/mte.a

Parameters Passed:

- -m (menu name)
- -o (office running the program)
- -p *PROG (Program)*
- -L (Site)
- -a (Automatic Query mode. Passed if the macro ENABLE_FEAT_AUTOMODE is set to Y)
- -F (Forced Query mode. Passed if the macro ENABLE_FEAT_FORCEQUERY is set to Y)
- -D (3) (Debugging level)

Biographical - One

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

Menuopt File: \$CARSPATH/menuopt/regist/reports/bione

Parameters Passed:

- PP_ID (*Student ID*)
- PP_SESS (*Session*)
- PP_ACAD_YR (*Academic year*)
- PP_PROG (*Program*)
- -f (Formtype)

Services by Student

Accesses: \$CARSPATH/modules/matric/reports/svcstu (ACE report)

Menuopt File: \$CARSPATH/menuopt/matric/reports/svcstu

Parameters Passed:

- *PP_ID* (Student ID)
- *PP_SERVICE* (Service code)
- *PP_SERVICE* (Flag to select last Service record only (Y), or all Service records(N))
- *PP_SERVICE* (Flag to include only those services not completed)

Admissions Entry

Accesses: \$CARSPATH/src/admit/admentry (Program)

Menuopt File: \$CARSPATH/menuopt/admit/programs/adme

Parameters Passed:

- Include Church Records form option if the macro ENABLE_FEAT_CHURCH is set to Y
- Include Business Records form option if the macro ENABLE_FEAT_ADM_BUSINESS is set to Y
- -a (Automatic Query mode. Passed if the macro ENABLE_FEAT_AUTOMODE is set to Y)
- -F (Forced Query mode. Passed if the macro ENABLE_FEAT_FORCEQUERY is set to Y)
- -T ADM_PROG_TICK (Tickler code)
- -o ADM_PROG_OFFICE (Office running the program)
- -m ADM_PROG_MENU (Menu from which program is run)
- -L (Site)
- -D (3) (Debugging level)

Graduate Admission Entry

Accesses: \$CARSPATH/src/admit/admentry (Program)

Menuopt File: \$CARSPATH/menuopt/admit/programs/gadme

Parameters Passed:

- Show Church Records option if the macro ENABLE_FEAT_CHURCH is set to Y
- Show Business Records option if the macro ENABLE_FEAT_ADM_BUSINESS is set to Y
- -a Automode (if ENABLE_FEAT_AUTOMODE set to Y)
- -F Forced query (if ENABLE_FEAT_FORCEQUERY set to Y)
- -p PROG_GRAD (Program code (e.g., GRAD))
- -T TICK_ADMG (Tickler code)
- -o ADM_PROG_OFFICE (Office running the program)
- -m ADM_PROG_MENU (Form selection menu from which program is run)
- -L (Site)
- -D 3 (Debugging level)

Admissions Statuses

Accesses: \$CARSPATH/src/admit/admstats (Program)

Menuopt File: \$CARSPATH/menuopt/admit/programs/adms

Parameters Passed:

- -t ADM_PROG_TICK (Tickler code)
- -p ADM_PROG_PROGRAM (Program code)
- -i ID# COMMENT_ID_ZERO

Update Waitlist

Accesses: \$CARSPATH/modules/admit/reports/setwaitrnk (ACE Report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/setwaitrnk

Parameters Passed:

- -f formtype
- PP_PROG (Program)
- PP_MAJOR (Major)
- PP_ADM_SESS (Session)
- PP_ADM_YEAR (Year)

Update Waitlist by Major

Accesses: \$CARSPATH/modules/admit/reports/setmajrnk (ACE Report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/setmajrnk

Parameters Passed:

- -f formtype
- PP_PROG (Program)
- PP_MAJOR (Major)
- PP_ADM_SESS (Session)
- PP_ADM_YEAR (Year)

Student Services

Accesses: \$CARSPATH/modules/stuser/screens/stuser (Screen)

Menuopt File: \$CARSPATH/menuopt/stuser/screens/stuser

Parameters Passed: None

Display Financial Aid

Accesses: \$CARSPATH/src/finaid/faentry (Program)

Menuopt File: \$CARSPATH/menuopt/finaid/programs/fae.dl

Parameters Passed:

- -D (Display-only mode)
- -I (Default student accounts subsidiary)
- -L (Site)

Display Registrations

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

Menuopt File: \$CARSPATH/menuopt/regist/programs/regent.d

Parameters Passed:

- -p PROG (Program)
- -L Site
- -F F (Registration finish functionality)
- -m H C (Disabled main menu options)
- -e R (Disabled enrollment menu option)
- -r C B R L T A (Disabled registration menu options)
- -d (Display-only mode)

Interactive Degree Audit

Accesses: \$CARSPATH/src/degaudd/degaudd (Program)

Menuopt File: \$CARSPATH/menuopt/degau/programs/dgau

Parameters Passed:

- -L Site
- -s Session
- -y Academic year
- -p PROG (Program)
- -f Audit form
- -o Printer name

Add Pre-Transfer Courses

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

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

Parameters Passed:

- -L Site
- -p PROG (Program)
- -a (Enables the adding of transfer equivalency records)
- -m (Enables update of move flag for pre-transfer work)

Print Student Pre-Trans Crs

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

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

Parameters Passed:

- Student ID
- -f Formtype

Schedule Entry

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

Menuopt File: \$CARSPATH/menuopt/common/programs/evnte

Parameters Passed: None

Add Contacts/School Visit

Accesses: \$CARSPATH/modules/admit/informers/colldayctc (SQL statement)

Menuopt File: \$CARSPATH/menuopt/admit/informers/colldayctc

Parameters Passed:

- PP_ID (School ID)
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YEAR (Planned enrollment year)
- PP_YEAR (Year selection operand)
- PP_PROG (Program)
- PP_STATUS (Excluded status, if any)
- PP_ID (Counselor ID)
- PP_DATE_DUE (Contact due date)
- PP_TICK (Tickler code)
- PP_CTC_RESRC (Contact Resource code)

Communications Management Menu

Select by User Parameters

Accesses: \$CARSPATH/modules/admit/informers/leadctc (SQL statement)

Menuopt File: \$CARSPATH/menuopt/admit/informers/leadctc

Parameters Passed:

- Planned enrollment session
- School ID#
- Planned enrollment year
- Year condition operand
- Program
- Major #1
- Major #2
- Referral source
- Responded
- Add date
- Add date condition operand
- Move date
- Move date condition operand
- State
- Country
- Beginning zip code
- Ending zip code
- Sex
- Ethnic
- Lead contact received
- Contact not received
- Tickler
- Due date
- Lead contact time
- Correspondent ID number
- Schedule contact

Lead Letters/Labels Due Report

Accesses: \$CARSPATH/modules/admit/ reports/leadctcdue (ACE Report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/leadctcdue

Parameters Passed:

- Tickler (Contact Tickler Code)
- Due Date (Due Date of Expected Lead Contacts)
- Summary (Print Summary or Detailed Report)

Create All Letters/Labels

Accesses: \$CARSPATH/modules/common/scripts/lpsrun

Menuopt File: \$CARSPATH/menuopt/common/scripts/lpsrun

Parameters Passed:

- Tickler (Contact Record Tickler Code)
- Date (Date to be Printed and Expanded on the Letters)
- Date (Contact Due Date)

Create School Letters

Accesses: \$CARSPATH/modules/common/scripts/ltrrun.sch

Menuopt File: \$CARSPATH/menuopt/common/scripts/ltrrun.sch

Parameters Passed:

- Letters/Labels/Both/None
- Tickler (Contact Record Tickler Code)
- Resource (Name of Letter)
- Date (Date to be Printed and Expanded on the Letters)
- Due Date (Contact Due Date)
- Format (File Format)
- Bulk Mail Processing (Y/N - Use Bulk Mail Processing?)
- Relationship Code (School to Employee Relationship Code)
- Default Addressee (Default Addressee Title)

Print Admissions Letters

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

Menuopt File: \$CARSPATH/menuopt/utilities/programs/lps.adm

Parameters Passed: None

Tickler Menu

Tickler System Entry

Accesses: \$CARSPATH/src/commgmt/tickent (Program)

Menuopt File: \$CARSPATH/menuopt/commgmt/programs/ldtent.ADM

Parameters Passed:

- -t (Lead Tickler Code – Not Displayed)

Interactive Tickler

Accesses: \$CARSPATH/src/commgmt/ldtickler (Program)

Menuopt File: \$CARSPATH/menuopt/commgmt/programs/ldtick.ADM

Parameters Passed:

- -t (Lead Tickler Code – Not Displayed)

Schedule Tickler Review

Accesses: \$CARSPATH/src/commgmt/ldtickler (Program)

Menuopt File: \$CARSPATH/menuopt/commgmt/programs/ldtick.rADM

Parameters Passed:

- -t (Lead Tickler Code – Not Displayed)
- -r (Review all lead tickler records and lead contact records in batch)

Tickler Structure Report

Accesses: \$CARSPATH/src/commgmt/ldtickler (Program)

Menuopt File: \$CARSPATH/menuopt/commgmt/programs/ldtick.oADM

Parameters Passed:

- -t (Lead Tickler Code – Not Displayed)
- -o (Print output of lead tickler strategy as entered in tickler tables)

Create All Letters/Labels

Accesses: \$CARSPATH/modules/common/scripts/ltbrun (Script)

Menuopt File: \$CARSPATH/menuopt/admit/scripts/ltrrun.ld

Parameters Passed:

- ACE report (Not displayed - ltrlead)
- Letters/Labels/Both/None (Selection of letters, labels, both, or none)
- Resource (Lead Contact Resource code)
- Date (Date to be printed on letters using the WP_TODAY macro)
- Date (Due date of lead contacts with expected status)
- Format (Type of Word Processing Format)
- Bulk Mail Processing (Create .blk File to use with Mailers + 4 software)

Reports Menu

Contact Reports

Contacts for One Student

Accesses: \$CARSPATH/modules/admit/reports/onectc (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/onectc

Parameters Passed:

- -f Formtype
- PP_ID (Student ID)
- PP_PROG (Program)

Contacts for All Students

Accesses: \$CARSPATH/modules/admit/reports/allctc (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/allctc

Parameters Passed:

- -f Formtype
- PP_TICK (Tickler code)
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YR (Planned enrollment year)
- PP_PROG (Program)
- PP_NEW_PAGE (Page break after each ID)

Contacts by Prospect

Accesses: \$CARSPATH/modules/admit/others/ctcrpt (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/others/ctcrpt

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YR (Planned enrollment year)
- PP_PROG (Program)
- PP_COL (Select database table/record and field name)
- PP_PATTERN (Selection string)
- PP_VALUE (Selection value)
- PP_SORT_FIELD (Value on which to sort)
- PP_NEW_PAGE (Page break after each ID)

Contacts by Counselors

Accesses: \$CARSPATH/modules/admit/reports/cnslrctc (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/cnslrctc

Parameters Passed:

- -f Formtype
- *PP_TICK (Tickler code)*
- *PP_ADM_SESS (Planned enrollment session)*
- *PP_ADM_YR (Planned enrollment year)*
- *PP_PROG (Program)*
- *PP_DATE_BEG (Beginning of date range)*
- *PP_DATE_END (End of date range)*

Expected Contacts

Accesses: \$CARSPATH/modules/admit/others/ctcexpect (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/others/ctcexpect

Parameters Passed:

- -f Formtype
- *PP_ADM_SESS (Planned enrollment session)*
- *PP_ADM_YR (Planned enrollment year)*
- *PP_PROG (Program)*
- *PP_CTC (Contact code)*
- *PP_TITLE (Report title)*
- *PP_SORT_FIELD (Value on which to sort)*

One Contact - All Students

Accesses: \$CARSPATH/modules/admit/others/ctcall (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/others/ctcall

Parameters Passed:

- -f Formtype
- *PP_ADM_SESS (Planned enrollment session)*
- *PP_ADM_YR (Planned enrollment year)*
- *PP_PROG (Program)*
- *PP_CTC (Contact code)*
- *PP_TITLE (Report title)*
- *PP_SORT_FIELD (Value on which to sort)*

Not Received One Contact

Accesses: \$CARSPATH/modules/admit/others/ctcnot (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/others/ctcnot

Parameters Passed:

- -f Formtype
- *PP_ADM_SESS (Planned enrollment session)*
- *PP_ADM_YR (Planned enrollment year)*
- *PP_PROG (Program)*
- *PP_CTC (Contact code)*
- *PP_TITLE (Report title)*
- *PP_SORT_FIELD (Value on which to sort)*

One Contact - Not Another

Accesses: \$CARSPATH/modules/admit/others/ctccond (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/others/ctccond

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YR (Planned enrollment year)
- PP_PROG (Program)
- PP_CTC (Contact code received)
- PP_CTC (Contact code not received)
- PP_TITLE (Report title)
- PP_SORT_FIELD (Value on which to sort)

Eligible for Decision

Accesses: \$CARSPATH/modules/admit/reports/elig_dec (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/elig_dec

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YR (Planned enrollment year)
- PP_PROG (Program)
- PP_TICK (Tickler code)
- PP_STATUS (Enrollment status)
- Add Contact (Add Contact)
- PP_CTC_RESRC (Resource)

Not Eligible for Decision

Accesses: \$CARSPATH/modules/admit/reports/not_elig (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/not_elig

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YR (Planned enrollment year)
- PP_PROG (Program)
- PP_TICK (Tickler code)
- PP_STATUS (Enrollment status)
- Partial Transcripts (N, P)
- PP_CNLSR_ID (Counselor ID)

Competitive Schools

Accesses: \$CARSPATH/modules/admit/reports/cmptvschl

Menuopt File: \$CARSPATH/menuopt/admit/reports/cmptvschl

Parameters Passed:

- PP_PROG (Program)
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YR (Planned enrollment year)
- PP_TICK (Tickler code)
- PP_CTC (Contact code)
- PP_SUMMARY (Summary report)

Counselor Reports

Appointments - 1 Counselor

Accesses: \$CARSPATH/modules/admit/reports/schdcoun (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/schdcoun

- -f Formtype
- PP_CTC_RESRC (Contact resource; default is value in ADM_CTC_APPT_DEF macro)
- PP_ID (Counselor ID)
- PP_DATE_BEG (Beginning of date range)
- PP_DATE_END (End of date range)
- PP_PROG (Program)

Appointments - 3 Counselors

Accesses: \$CARSPATH/modules/admit/reports/schdcal (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/scripts/schdcal

Parameters Passed:

- -f Formtype
- PP_CTC_RESRC (Contact resource; default is value in ADM_CTC_APPT_DEF macro)
- PP_DATE_BEG (Beginning of date range)
- PP_DATE_END (End of date range)
- PP_ID (First counselor ID)
- PP_FIRST (First counselor name)
- PP_ID (Second counselor ID)
- PP_SECOND (Second counselor name)
- PP_ID (Third counselor ID)
- PP_THIRD (Third counselor name)

Activity Schedule

Accesses: \$CARSPATH/modules/common/reports/schdcal (ACE report)

Menuopt File: \$CARSPATH/menuopt/common/reports/schdcal

Parameters Passed:

- -f Formtype
- PP_ID (Counselor ID (or other individual with Schedule records))
- PP_DATE_BEG (Beginning of date range)
- PP_DATE_END (End of date range)

Students - Scheduled Visits

Accesses: \$CARSPATH/modules/admit/script/runcnslr (Script)

Menuopt File: \$CARSPATH/menuopt/admit/scripts/runcnslr

Parameters Passed:

- -f Formtype
- PP_DATE_BEG (Beginning of date range)
- PP_DATE_END (End of date range)
- PP_ID (Counselor ID)
- PP_ADM_YEAR (Planned enrollment year)
- PP_PROG (Program)
- PP_OUTPUT (Printer)

Counselor School/Prospect

Accesses: \$CARSPATH/modules/admit/reports/cnslrsch

Menuopt File: \$CARSPATH/menuopt/admit/reports/cnslrsch

Parameters Passed:

- *PP_PROG (Program)*
- *PP_ADM_SESS (Planned enrollment session)*
- *PP_ADM_YR (Planned enrollment year)*
- *PP_STATUS (Enrollment status)*
- *PP_SUMMARY (Summary report)*

Examination Scores

Accesses: \$CARSPATH/modules/admit/others/examrpt

Menuopt File: \$CARSPATH/menuopt/admit/others/examrpt

Parameters Passed:

- *PP_PROG (Program)*
- *PP_ADM_SESS (Planned enrollment session)*
- *PP_ADM_YR (Planned enrollment year)*
- *PP_STATUS (Enrollment status)*
- *PP_EXAM (Examination code)*
- *PP_SORT_FIELD (Value on which to sort)*

Scheduled Interviews

Accesses: \$CARSPATH/modules/admit/reports/interview

Menuopt File: \$CARSPATH/menuopt/admit/reports/interview

Parameters Passed:

- *PP_PROG (Program)*
- *PP_MAJOR (Major code)*
- *PP_STATUS (Enrollment status)*
- *Interviewer ID (Interviewer's ID number)*
- *PP_DATE_BEG (Beginning of date range)*
- *PP_DATE_END (End of date range)*

Student Visit Agenda

Accesses: \$CARSPATH/modules/admit/reports/stuvst

Menuopt File: \$CARSPATH/menuopt/admit/reports/stuvst

Parameters Passed:

- *PP_ID (Visitor ID)*
- *PP_PROG (Program)*
- *PP_DATE_BEG (Beginning of date range)*
- *PP_DATE_END (End of date range)*

Profile Reports

User Defined Profile Info

Accesses: \$CARSPATH/modules/admit/others/profile (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/others/profile

Parameters Passed:

- *-f Formtype*
- *PP_OUTPUT (Printer)*
- *PP_SORT_FIELD (Value on which to sort)*

- *PP_ADM_SESS* (Planned enrollment session)
- *PP_ADM_YEAR* (Planned enrollment year)
- *PP_YEAR* (Year selection operand)
- *PP_PROG* (Program)
- *PP_TICK* (Tickler code)
- *PP_ID* (Counselor ID)
- *PP_STATUS* (Enrollment status)
- *PP_MAJOR* (Major code)
- *PP_INT* (Interest code)
- *PP_ID* (School ID)
- *PP_ID* (Church ID)
- *PP_ZIP_BEG* (Beginning Zip code)
- *PP_ZIP_END* (Ending Zip code)
- *PP_BREAK* (Flag to cause the report to break when the sort value changes (valid codes are (P)age, (Y)es, and (N)o))

One Student's Profile

Accesses: \$CARSPATH/modules/admit/others/profone (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/others/profone

Parameters Passed:

- -f Formtype
- *PP_ID* (Student ID)
- *PP_PROG* (Program)
- *PP_TICK* (Tickler code)

Students by School

Accesses: \$CARSPATH/modules/admit/scripts/hsprosp (Script)

Menuopt File: \$CARSPATH/menuopt/admit/scripts/hsprosp

Parameters Passed:

- *PP_ID* (School ID)
- *PP_ADM_YR* (Planned enrollment year)
- *PP_PROG* (Program)
- *PP_OUTPUT* (Printer)
- Site (Default is CARSSITE)

School Profile

Accesses: \$CARSPATH/modules/admit/scripts/hsreport (Script)

Menuopt File: \$CARSPATH/menuopt/admit/scripts/hsreport

- *PP_ID* (School ID)
- *PP_SESS* (Enrollment session)
- *PP_ACAD_YR* (Academic year for identifying enrolled students)
- *PP_YEAR* (Planned enrollment year for identifying prospects)
- *PP_PROG* (Program)
- *PP_OUTPUT* (Printer)
- Site (Default is CARSSITE)

Students by Church

Accesses: \$CARSPATH/modules/admit/scripts/chprosp (Script)

Menuopt File: \$CARSPATH/menuopt/admit/scripts/chprosp

Parameters Passed:

- *PP_ID (Church ID)*
- *PP_ADM_YR (Planned enrollment year)*
- *PP_PROG (Program)*
- *PP_OUTPUT (Printer)*

Church Profile

Accesses: \$CARSPATH/modules/admit/script/chreport (Script)

Menuopt File: \$CARSPATH/menuopt/admit/scripts/chreport

- *PP_ID (Church ID)*
- *PP_SESS (Enrollment session)*
- *PP_ACAD_YR (Academic year for identifying enrolled students)*
- *PP_YEAR (Planned enrollment year for identifying prospects)*
- *PP_PROG (Program)*
- *PP_OUTPUT (Printer)*
- *Site (Default is CARSSITE)*

School Directory

Accesses: \$CARSPATH/modules/admit/others/schdir

Menuopt File: \$CARSPATH/menuopt/admit/others/schdir

- *City (City identifying school location)*
- *PP_STATE (State identifying school location)*
- *PP_ZIP_RAN (Zip code identifying school location)*
- *County (County identifying school location)*
- *PP_COUNTRY (Country identifying school location)*
- *Area Code (Area code identifying school location)*
- *PP_SCH_TYPE (School category, e.g., High School, 2 year college, etc.)*
- *Public/Private (School type)*
- *Denomination (Denomination)*
- *College Day Date Range (Beginning date of college visits)*
- *PP_CNCLR_ID (Admission Counselor's ID number)*
- *PP_SUMMARY (Summary report)*

Applicant Committee Review

Accesses: \$CARSPATH/modules/admit/reports/committee

Menuopt File: \$CARSPATH/menuopt/admit/others/committee

- *PP_ID (Applicant's ID number)*
- *PP_PROG (Program code)*
- *PP_ADM_SESS (Planned enrollment session)*
- *PP_ADM_YR (Planned enrollment year)*
- *PP_STATUS (Enrollment status)*

Statistical Reports

Counts, Subtotals

Accesses: \$CARSPATH/modules/admit/others/countrpt (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/others/countrpt

Parameters Passed:

- *-f Formtype*
- *PP_SORT_FIELD_PRIM (Primary sort field)*

- *PP_SORT_FIELD_SEC* (Secondary sort field)
- *PP_ADM_SESS* (Planned enrollment session)
- *PP_ADM_YR* (Planned enrollment year)
- *PP_YEAR* (Operand selection criteria for choosing years (i.e., =, <, or >))
- *PP_PROG* (Program)
- *PP_ID* (Counselor ID)
- *PP_STATUS* (Current admission status)
- *PP_MAJOR* (Major code)
- *PP_INT* (Interest code)
- *PP_EXAM* (Exam code)
- *PP_ID* (School ID)
- *PP_TYPE* (School type)
- *PP_ID* (Church ID)
- *PP_ZIP_BEG* (Beginning Zip code)
- *PP_ZIP_END* (Ending Zip code)
- *PP_NEW_PAGE* (Page break after each primary sort change)
- *PP_SUMMARY* (Summary report)

Enroll History - Years

Accesses: \$CARSPATH/modules/admit/reports/enrhistr (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/enrhistr

Parameters Passed:

- *PP_ADM_SESS* (Planned enrollment session)
- *PP_YEAR_BEG* (Beginning enrollment year)
- *PP_YEAR_END* (Ending enrollment year)
- *PP_DATE* (Admission status deadline date)
- *PP_PROG* (Program)
- *PP_SEX* (Selection by sex)
- *PP_ETHNIC* (Selection by Ethnic code)
- *PP_PRINT* (Report elements to print)
- *PP_SUMMARY* (Summary or detail report)

Enroll History - Percent

Accesses: \$CARSPATH/modules/admit/reports/enrhispct (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/enrhispct

Parameters Passed:

- *PP_ADM_SESS* (Planned enrollment session)
- *PP_YEAR_BEG* (Beginning enrollment year)
- *PP_YEAR_END* (Ending enrollment year)
- *PP_DATE* (Admission status deadline date)
- *PP_PROG* (Program)
- *PP_SEX* (Selection by sex)
- *PP_ETHNIC* (Selection by Ethnic code)
- *PP_FIRST* (First enrollment status to include)
- *PP_SECOND* (Second enrollment status to include)
- *PP_THIRD* (Third enrollment status to include)
- *PP_FOURTH* (Fourth enrollment status to include)

Inclusive Statuses

Accesses: \$CARSPATH/modules/admit/reports/curstat (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/curstat

Parameters Passed:

- -f Formtype
- PP_ADM_YEAR (Planned enrollment year)
- PP_ADM_SESS (Planned enrollment session)
- PP_PROG (Program)
- PP_DATE (Admission status deadline date)

Current Statuses

Accesses: \$CARSPATH/modules/admit/reports/admcursat (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/admcursat

Parameters Passed:

- -f Formtype
- PP_ADM_YEAR (Planned enrollment year)
- PP_ADM_SESS (Planned enrollment session)
- PP_PROG (Program)

Status Sequences

Accesses: \$CARSPATH/modules/admit/reports/seqcount (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/seqcount

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YEAR (Planned enrollment year)
- PP_PROG (Program)
- Admission status deadline date

Contact Resources

Accesses: \$CARSPATH/modules/admit/reports/resrcappl (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/resrcappl

Parameters Passed:

- -f Formtype
- PP_TICK (Tickler code)
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YEAR (Planned enrollment year)
- PP_DATE_BEG (Beginning of date range of contact resource)
- PP_DATE_END (End of date range of contact resource)

Reference Sources

Accesses: \$CARSPATH/modules/admit/reports/refsrc (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/refsrc

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YEAR (Planned enrollment year)
- PP_PROG (Program)
- PP_DATE (Admission record add date)
- PP_REF (Beginning Reference code)

- *PP_REF* (End Reference code)
- *PP_SUMMARY* (Summary or detail report)

Sources - Inclusive Statuses

Accesses: \$CARSPATH/modules/admit/reports/enrrefsrc (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/enrrefsrc

Parameters Passed:

- -f Formtype
- *PP_ADM_SESS* (Planned enrollment session)
- *PP_ADM_YEAR* (Planned enrollment year)
- *PP_PROG* (Program)
- *PP_DATE* (Beginning date of admission status)
- *PP_REF* (Beginning Reference code)
- *PP_REF* (End Reference code)
- *PP_FIRST* (First enrollment status to include)
- *PP_SECOND* (Second enrollment status to include)
- *PP_THIRD* (Third enrollment status to include)
- *PP_SUMMARY* (Summary or detail report)

Sources - Current Statuses

Accesses: \$CARSPATH/modules/admit/reports/refsrcenr (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/refsrcenr

Parameters Passed:

- -f Formtype
- *PP_ADM_SESS* (Planned enrollment session)
- *PP_ADM_YEAR* (Planned enrollment year)
- *PP_PROG* (Program)
- *PP_REF* (Beginning Reference code)
- *PP_REF* (End Reference code)
- *PP_FIRST* (First enrollment status to include)
- *PP_SECOND* (Second enrollment status to include)
- *PP_THIRD* (Third enrollment status to include)
- *PP_SUMMARY* (Summary or detail report)

Majors - Inclusive Statuses

Accesses: \$CARSPATH/modules/admit/reports/enrmajor (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/enrmajor

Parameters Passed:

- -f Formtype
- *PP_ADM_SESS* (Planned enrollment session)
- *PP_ADM_YEAR* (Planned enrollment year)
- *PP_PROG* (Program)
- *PP_DATE* (Beginning date of admission status)
- *PP_MAJOR* (First major for report)
- *PP_MAJOR* (Last major for report)
- *PP_FIRST* (First enrollment status to include)
- *PP_SECOND* (Second enrollment status to include)
- *PP_THIRD* (Third enrollment status to include)
- *PP_SUMMARY* (Summary or detail report)

Majors - Current Statuses

Accesses: \$CARSPATH/modules/admit/reports/major (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/major

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YEAR (Planned enrollment year)
- PP_PROG (Program)
- PP_MAJOR (First major for report)
- PP_MAJOR (Last major for report)
- PP_FIRST (First enrollment status to include)
- PP_SECOND (Second enrollment status to include)
- PP_THIRD (Third enrollment status to include)
- PP_SUMMARY (Summary or detail report)

Summary by Schools

Accesses: \$CARSPATH/modules/admit/reports/collsrc (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/collsrc

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YEAR (Planned enrollment year)
- PP_PROG (Program)
- PP_Status (Admissions status)

ACT Composite Scores

Accesses: \$CARSPATH/modules/admit/reports/actprof (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/actprof

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YEAR (Planned enrollment year)
- PP_PROG (Program)
- PP_EXAM (Type of ACT exam)

ACT Profile vs. HS Rank

Accesses: \$CARSPATH/modules/admit/reports/actprofrnk (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/actprofrnk

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YEAR (Planned enrollment year)
- PP_PROG (Program)
- PP_STATUS (Enrollment status)
- PP_SEX (Selection by sex)
- PP_DEC (Selection by Decision code)
- PP_DIV (Selection by college division)

SAT Profile vs. HS Rank

Accesses: \$CARSPATH/modules/admit/reports/satprofrnk (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/satprofrnk

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YEAR (Planned enrollment year)
- PP_PROG (Program)
- PP_STATUS (Enrollment status)
- PP_SEX (Selection by sex)
- PP_DEC (Selection by Decision code)
- PP_DIV (Selection by college division)

Geographical Distribution

Accesses: \$CARSPATH/modules/admit/reports/geography (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/geography

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YR (Planned enrollment year)
- PP_PROG (Program)
- PP_STATUS (If not using Matriculation module) (Current enrollment status)

Students by Sex and Age

Accesses: \$CARSPATH/modules/admit/reports/sexage (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/sexage

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YR (Planned enrollment year)
- PP_PROG (Program)
- PP_MAJOR (First major for report)
- PP_MAJOR (Last major for report)
- PP_STATUS (If not using Matriculation module) (Current enrollment status)

Session Processing Menu

Create Program Enrollment

Accesses: \$CARSPATH/modules/regist/informers/progenr (SQL statement)

Menuopt File: \$CARSPATH/menuopt/regist/informers/progenr

Parameters Passed:

- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YR (Planned enrollment year)
- PP_STATUS (Current enrollment status)
- PP_SESS (Probable graduation session)
- PP_PROG (Program)
- PP_CLASS (Students' classification (e.g., first time freshman))
- PP_CAT (Catalog)

Students Not Registered

Accesses: \$CARSPATH/modules/admit/reports/resnreg (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/resnreg

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YR (Planned enrollment year)
- PP_PROG (Program)
- PP_STATUS (Current enrollment status)

Update to Final Status

Accesses: \$CARSPATH/modules/admit/informers/regctc (SQL statement)

Menuopt File: \$CARSPATH/menuopt/admit/informers/regctc

Parameters Passed:

- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YR (Planned enrollment year)

Add Required Documents

Accesses: \$CARSPATH/modules/admit/informers/reqadmdoc

Menuopt File: \$CARSPATH/menuopt/admit/informers/reqadmdoc

Parameters Passed:

- PP_PROG (Program)
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YR (Planned enrollment year)
- PP_TICK (Tickler code)
- PP_STATUS (Enrollment status)
- Document Category

Set Counselor Territories

Accesses: \$CARSPATH/modules/admit/informers/setcnslr

Menuopt File: \$CARSPATH/menuopt/admit/informers/setcnslr

Parameters Passed:

- city
- PP_ST
- PP_AI

Reassign Counselors

Accesses: \$CARSPATH/modules/admit/informers/updcnslr

Menuopt File: \$CARSPATH/menuopt/admit/informers/updcnslr

Parameters Passed:

- Program
- Year (two fields 1) > = < and 2) Year)
- Sessions (four fields Session Codes 1 through 4)
- Past Admission Counselor ID
- New Admission Counselor ID

Update Age of Students

Accesses: \$CARSPATH/modules/regist/informers/infage
Menuopt File: \$CARSPATH/menuopt/regist/informers/infage
Parameters Passed: None

Query by Form

ID/Profile/Relationship

Accesses: \$CARSPATH/modules/common/screens/idprofil_q (PERFORM screen)
Menuopt File: \$CARSPATH/menuopt/common/screens/idprofil_q
Parameters Passed: None

Note: Displays the id_rec, profile_rec, and relation_rec

Admissions

Accesses: \$CARSPATH/modules/admit/screens/admit_q (PERFORM screen)
Menuopt File: \$CARSPATH/menuopt/admit/screens/admit_q
Parameters Passed: None

Note: Displays the id_rec and adm_rec.

Admissions Statistics

Accesses: \$CARSPATH/modules/admit/screens/admstats (PERFORM screen)
Menuopt File: \$CARSPATH/menuopt/admit/screens/admstats
Parameters Passed: None

Note: Displays the adm_stat_rec.

Enrollment Status

Accesses: \$CARSPATH/modules/admit/screens/enrstat (PERFORM screen)
Menuopt File: \$CARSPATH/menuopt/admit/screens/enrstat
Parameters Passed: None

Note: Displays the id_rec, adm_rec, and enr_stat_rec.

Statuses

Accesses: \$CARSPATH/modules/admit/screens/statuses (PERFORM screen)
Menuopt File: \$CARSPATH/menuopt/admit/screens/statuses
Parameters Passed: None

Note: Displays the id_rec, adm_rec, prog_enr_rec, stu_acad_rec, ctc_rec, hold_rec, and hold_act_rec.

Schools

Accesses: \$CARSPATH/modules/common/screens/school_q (PERFORM screen)
Menuopt File: \$CARSPATH/menuopt/common/screens/school_q
Parameters Passed: None

Note: Displays the id_rec, ed_rec, relation_rec, and sch_rec.

Churches

Accesses: \$CARSPATH/modules/common/screens/church_q (PERFORM screen)

Menuopt File: \$CARSPATH/menuopt/common/screens/church_q

Parameters Passed: None

Note: Displays the id_rec, church_rec, relation_rec, and profile_rec.

Lead Processing Menu

Lead Entry

Accesses: \$CARSPATH/src/admit/leadent (Program)

Menuopt File: \$CARSPATH/menuopt/admit/programs/leade

Parameters Passed:

- -o Office Code
- -t Lead Contact Tickler Code
- -p Program Code
- -T Contact Tickler Code
- -r Contact Resource Code

Move Leads to Inquired

Accesses: \$CARSPATH/modules/admit/informers/moveleads

Menuopt File: \$CARSPATH/menuopt/admit/informers/moveleads

Parameters Passed:

- ID# (Lead Number of respondent or zero for all respondents)
- Program (Program code to assign to student's admissions record)
- Tickler (Tickler code to assign to student's initial contact record)
- Resource (Name of initial contact resource)

Remove Specific Leads

Accesses: \$CARSPATH/modules/admit/informers/rmleads

Menuopt File: \$CARSPATH/menuopt/admit/informers/rmleads

Parameters Passed:

- Date Added (Delete lead records where add_date <= date entered)
- Date Moved (Delete lead records where move_date <= date entered)

Communications Management Menu

Select by User Parameters

Accesses: \$CARSPATH/modules/admit/informers/leadctc (SQL statement)

Menuopt File: \$CARSPATH/menuopt/admit/informers/leadctc

Parameters Passed:

- Planned enrollment session
- School ID#
- Planned enrollment year
- Year condition operand
- Program
- Major #1
- Major #2

- Referral source
- Responded
- Add date
- Add date condition operand
- Move date
- Move date condition operand
- State
- Country
- Beginning zip code
- Ending zip code
- Sex
- Ethnic
- Lead contact received
- Contact not received
- Tickler
- Due date
- Lead contact time
- Correspondent ID number
- Schedule contact

Lead Letters/Labels Due

Accesses: \$CARSPATH/modules/admit/ reports/leadctcdu (ACE Report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/leadctcdu

Parameters Passed:

- Tickler (Lead Contact Tickler Code)
- Due Date (Due Date of Expected Lead Contacts)
- Summary (Print Summary or Detailed Report)

Create Lead Letters/Labels

Accesses: \$CARSPATH/modules/common/scripts/ltrun (Script)

Menuopt File: \$CARSPATH/menuopt/admit/scripts/ltrun.ld

Parameters Passed:

- ACE report (Not displayed - ltrlead)
- Letters/Labels/Both/None (Selection of letters, labels, both, or none)
- Resource (Lead Contact Resource code)
- Date (Date to be printed on letters using the WP_TODAY macro)
- Date (Due date of lead contacts with expected status)
- Format (Type of Word Processing Format)
- Bulk Mail Processing (Create .blk File to use with Mailers + 4 software)

Print Lead Letters/Labels

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

Menuopt File: \$CARSPATH/menuopt/utilities/programs/lps.ld

Parameters Passed: None

Lead Tickler Menu

Lead Tickler System Entry

Accesses: \$CARSPATH/src/commgmt/tickent (Program)

Menuopt File: \$CARSPATH/menuopt/commgmt/programs/ldtent.ADM

Parameters Passed:

- -t (Lead Tickler Code – Not Displayed)

Interactive Lead Tickler

Accesses: \$CARSPATH/src/commgmt/ldtickler (Program)

Menuopt File: \$CARSPATH/menuopt/commgmt/programs/ldtick.ADM

Parameters Passed:

- -t (Lead Tickler Code – Not Displayed)

Lead Tickler Review

Accesses: \$CARSPATH/src/commgmt/ldtickler (Program)

Menuopt File: \$CARSPATH/menuopt/commgmt/programs/ldtick.rADM

Parameters Passed:

- -t (Lead Tickler Code – Not Displayed)
- -r (Review all lead tickler records and lead contact records in batch)

Tickler Structure Report

Accesses: \$CARSPATH/src/commgmt/ldtickler (Program)

Menuopt File: \$CARSPATH/menuopt/commgmt/programs/ldtick.oADM

Parameters Passed:

- -t (Lead Tickler Code – Not Displayed)
- -o (Print output of lead tickler strategy as entered in tickler tables)

Lead Reports

Lead Roster by Parameters

Accesses: \$CARSPATH/modules/admit/others/leadrpt (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/others/leadrpt

Parameters Passed:

- *PP_STATE* (State of Leads' Address)
- *PP_ZIP_RAN* (Zip Code Range of Leads' Address)
- *PP_COUNTRY* (Country of Leads' Address)
- *PP_PROG* (Leads' Intended Academic Program, i.e. UNDG)
- *PP_MAJOR* (Leads' First Choice of Major)
- *PP_MAJOR2* (Leads' Second Choice of Major)
- *PP_ADM_SESS* (Leads' Planned enrollment session)
- *PP_ADM_YEAR* (Leads' Planned enrollment year)
- *PP_REF* (Leads' source of referral)
- *PP_SCHOOL_ID* (Leads' school of attendance)
- *PP_SEX* (Leads' gender)
- *PP_ETHNIC* (Leads' ethnic code)
- *PP_RESPONDED* (Lead respondents vs. non-respondents)
- *PP_MOVE_DATE* (Date lead record was moved to permanent records)
- *PP_ADD_DATE* (Date lead record was added to database)
- *PP_SORT_FIELD_PRIM* (Primary sort field. Enables you to order the data on the report output)

Leads by Major

Accesses: \$CARSPATH/modules/admit/reports/leadmaj (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/leadmaj

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_YEAR_BEG (First year to select)
- PP_YEAR_END (Last year to select)

Leads by Contact Resource

Accesses: \$CARSPATH/modules/admit/reports/leadresrc (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/leadresrc

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_YEAR_BEG (First year to select)
- PP_YEAR_END (Last year to select)

Leads by School

Accesses: \$CARSPATH/modules/admit/reports/leadceeb (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/leadceeb

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_YEAR_BEG (First year to select)
- PP_YEAR_END (Last year to select)

Leads by Sex

Accesses: \$CARSPATH/modules/admit/reports/leadsex (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/leadsex

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_YEAR_BEG (First year to select)
- PP_YEAR_END (Last year to select)

Leads by Reference Source

Accesses: \$CARSPATH/modules/admit/reports/leadsrc (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/leadsrc

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_YEAR_BEG (First year to select)
- PP_YEAR_END (Last year to select)

Leads by State

Accesses: \$CARSPATH/modules/admit/reports/leadst (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/leadst

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_YEAR_BEG (First year to select)
- PP_YEAR_END (Last year to select)

Leads by Zip

Accesses: \$CARSPATH/modules/admit/reports/leadzip (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/leadzip

Parameters Passed:

- -f Formtype
- PP_ADM_SESS (Planned enrollment session)
- PP_YEAR_BEG (First year to select)
- PP_YEAR_END (Last year to select)

Data Conversion

ACT Exam Convert/Add

Accesses: \$CARSPATH/modules/admit/scripts/tpconv (Runs act and act2_awk reports) (Script)

Menuopt File: \$CARSPATH/menuopt/admit/scripts/act

Parameters Passed:

- Output type (printer, file, or more)

ETS SAT Exam Convert/Add

Accesses: \$CARSPATH/modules/admit/scripts/tpconv (Runs ets and ets2_awk scripts) (Script)

Menuopt File: \$CARSPATH/menuopt/admit/scripts/ets

Parameters Passed:

- Output type (printer, file, or more)

SSN Import Report – DUP

Accesses: \$CARSPATH/modules/admit/reports/import_dob; import_ssn; import_zip

Menuopt File: \$CARSPATH/menuopt/admit/reports/ import_dob; import_ssn; import_zip

Parameters Passed:

- PP_FMT (Format)
- PP_YR (Year)
- PP_SITE (Site)
- PP_STAT (Status)

SSN Import Report – Blank

Accesses: \$CARSPATH/modules/admit/reports/import_blk

Menuopt File: \$CARSPATH/menuopt/admit/reports/import_blk

Parameters Passed:

- PP_FMT (Format)
- PP_YR (Year)
- PP_SITE (Site)

- *PP_STAT (Status)*

Correct Duplicate Imports

Accesses: \$CARSPATH/src/admit/admimport

Menuopt File: \$CARSPATH/menuopt/admit/programs/import

Parameters Passed:

- *PP_FMT (Format)*
- *PP_YR (Year)*
- *PP_OFF (Office)*

Create Import File

Accesses: \$CARSPATH/src/admit/admimport-g

Menuopt File: \$CARSPATH/menuopt/admit/programs/import-bg

Parameters Passed:

- *PP_FMT (format)*
- *PP_YR (Year)*

Import Format Table Report

Accesses: \$CARSPATH/modules/admit/reports/timportfmt

Menuopt File: \$CARSPATH/menuopt/admit/reports/timportfmt

Import Format Table

Accesses: \$CARSPATH/modules/admit/screens/timportfmt

Menuopt File: \$CARSPATH/menuopt/admit/screens/timportfmt

Parameters Passed:

- *PP_FMT (Format Code)*
- *PP_TXT (Description)*
- *PP_YR (Format Year)*
- *PP_LNAME_IND (Last Name Beginning Position)*
- *PP_LNAME_LEN (Last Name Length)*
- *PP_FNAME_IND (First Name Beginning Position)*
- *PP_FNAME_LEN (First Name Length)*
- *PP_MI_IND (Middle Initial Beginning Position)*
- *PP_MI_LEN (Middle Initial Length)*
- *PP_SSNO_IND (SSN Beginning Position)*
- *PP_SSNO_LEN (SSN Length)*
- *PP_BIRTHDATE_IND (Birthdate Beginning Position)*
- *PP_BIRTHDATE_LEN (Birthdate Length)*
- *PP_ZIP_IND (ZIP Code Beginning Position)*
- *PP_ZIP_LEN (ZIP Code Length)*
- *PP_ACTIVE (Active Date)*
- *PP_INACTIVE (Inactive Date)*

Import Maintenance

Accesses: \$CARSPATH/modules/admit/programs/import_wash

Menuopt File: \$CARSPATH/menusrc/admit/programs/import_wash

Parameters Passed:

- *PP_FMT (format)*

- PP_YR (Year)

Call Entry Menu

Call Entry

Accesses: \$CARSPATH/src/admit/callentry (Program)

Menuopt File: \$CARSPATH/menuopt/admit/programs/calle

Parameters Passed:

- PP_PROG (Program)
- PP_TICK (Tickler code)

Caller Time Entry

Accesses: \$CARSPATH/modules/admit/screens/callstat (PERFORM screen)

Menuopt File: \$CARSPATH/menuopt/admit/screens/callstat

Parameters Passed: None

Call Entry Result Table

Accesses: \$CARSPATH/modules/admit/screens/tcallresult (PERFORM screen)

Menuopt File: \$CARSPATH/menuopt/admit/screens/tcallresult

Parameters Passed: None

Call Entry Reports

Call Contact Schedule

Accesses: \$CARSPATH/modules/admit/reports/callctcs (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/callctcs

Parameters Passed:

- -f Formtype
- PP_DATE (Contact due date)
- PP_ADM_SESS (Planned enrollment session)
- PP_ADM_YR (Planned enrollment year)
- PP_PROG (Program)
- PP_TICK (Tickler code)
- PP_CTC_RESRC (Contact Resource code)
- PP_ID (Caller's ID)

Call Entry Information

Accesses: \$CARSPATH/modules/admit/reports/callinfo (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/callinfo

Parameters Passed:

- -f Formtype
- PP_DATE_BEG (Beginning of date range)
- PP_DATE_END (End of date range)
- PP_PROG (Program)

Calls Completed

Accesses: \$CARSPATH/modules/admit/reports/callmktctc (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/callmktctc

Parameters Passed:

- -f Formtype
- *PP_TICK (Tickler code)*
- *PP_DATE_BEG (Beginning of date range)*
- *PP_DATE_END (End of date range)*
- *PP_ID (Caller's ID)*

Caller Time Entry

Accesses: \$CARSPATH/modules/admit/reports/calltime (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/calltime

Parameters Passed:

- -f Formtype
- *PP_DATE_BEG (Beginning of date range)*
- *PP_DATE_END (End of date range)*
- *PP_ID (Caller's ID)*

Weekly Stats by Caller

Accesses: \$CARSPATH/modules/admit/reports/callstat (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/callstat

Parameters Passed:

- -f Formtype
- *PP_DATE_BEG (Beginning of date range)*
- *PP_DATE_END (End of date range)*
- *PP_TICK (Tickler code)*

Weekly Stats by Contact

Accesses: \$CARSPATH/modules/admit/reports/callstat2 (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/callstat2

Parameters Passed:

- -f Formtype
- *PP_DATE_BEG (Beginning of date range)*
- *PP_DATE_END (End of date range)*
- *PP_TICK (Tickler code)*

Daily Stats by Caller

Accesses: \$CARSPATH/modules/admit/reports/callstatd (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/callstatd

Parameters Passed:

- -f Formtype
- *PP_DATE (Report date)*
- *PP_TICK (Tickler code)*

Call Result Table Report

Accesses: \$CARSPATH/modules/admit/reports/tcallres (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/tcallres

Parameters Passed: None

Call Entry Contact Table Report

Accesses: \$CARSPATH/modules/admit/reports/tcalltmctc (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/tcalltmctc

Parameters Passed:

- -f Formtype
- PP_TICK_BEG (First Tickler code to select)
- PP_TICK_END (Last Tickler code to select)

Elec. Applications Menu

Review New Elec. Apps

Accesses: \$CARSPATH/src/admit/elecapp (Program)

Menuopt File: \$CARSPATH/menuopt/admit/programs/elecapp

Parameters Passed:

- PP_TICK (Tickler code)
- PP_CTC_RESRC (Contact resource code)
- PP_STATUS (Contact status code)
- PP_PROG (Program)
- PP_DATE_BEG (Beginning of date range)
- PP_DATE_END (End of date range)
- PP_TIME_BEG (Beginning time)
- PP_TIME_END (Ending time)
- PP_SOURCE (Application source)
- PP_OFFICE (Office code)
- PP_SITE (Site)

Note: The Site parameter does not appear on the parameter screen. The value for this parameter is passed in the background, using the value of the end user's CARSSITE environment variable.

Review Rejected Elec Apps

Accesses: \$CARSPATH/src/admit/elecapp (Program)

Menuopt File: \$CARSPATH/menuopt/admit/programs/elecapp

Parameters Passed:

- PP_TICK (Tickler code)
- PP_CTC_RESRC (Contact resource code)
- PP_STATUS (Contact status code)
- PP_PROG (Program)
- PP_DATE_BEG (Beginning of date range)
- PP_DATE_END (End of date range)
- PP_TIME_BEG (Beginning time)
- PP_TIME_END (Ending time)
- PP_SOURCE (Application source)
- PP_OFFICE (Office code)
- PP_SITE (Site)
- -r (Select rejected applications only)

Note: The Site parameter does not appear on the parameter screen. The value for this parameter is passed in the background, using the value of the end user's CARSSITE environment variable.

Review Partial Elec Apps

Accesses: \$CARSPATH/src/admit/elecapp (Program)

Menuopt File: \$CARSPATH/menuopt/admit/programs/elecappi

Parameters Passed:

- *PP_TICK* (Tickler code)
- *PP_CTC_RESRC* (Contact resource code)
- *PP_STATUS* (Contact status code)
- *PP_PROG* (Program)
- *PP_DATE_BEG* (Beginning of date range)
- *PP_DATE_END* (End of date range)
- *PP_TIME_BEG* (Beginning time)
- *PP_TIME_END* (Ending time)
- *PP_SOURCE* (Application source)
- *PP_OFFICE* (Office code)
- *PP_SITE* (Site)
- *-i* (Select incomplete applications only)

Note: The Site parameter does not appear on the parameter screen. The value for this parameter is passed in the background, using the value of the end user's CARSSITE environment variable.

Review Elec Apps on Hold

Accesses: \$CARSPATH/src/admit/elecapp (Program)

Menuopt File: \$CARSPATH/menuopt/admit/programs/elecapph

Parameters Passed:

- *PP_TICK* (Tickler code)
- *PP_CTC_RESRC* (Contact resource code)
- *PP_STATUS* (Contact status code)
- *PP_PROG* (Program)
- *PP_DATE_BEG* (Beginning of date range)
- *PP_DATE_END* (End of date range)
- *PP_TIME_BEG* (Beginning time)
- *PP_TIME_END* (Ending time)
- *PP_SOURCE* (Application source)
- *PP_OFFICE* (Office code)
- *PP_SITE* (Site)
- *-h* (Select applications on hold)

Note: The Site parameter does not appear on the parameter screen. The value for this parameter is passed in the background, using the value of the end user's CARSSITE environment variable.

Batch Elec Apps

Accesses: \$CARSPATH/src/admit/elecapp (Program)

Menuopt File: \$CARSPATH/menuopt/admit/programs/elecappg

Parameters Passed:

- *PP_TICK* (Tickler code)
- *PP_CTC_RESRC* (Contact resource code)
- *PP_STATUS* (Contact status code)
- *PP_PROG* (Program)

- *PP_DATE_BEG* (Beginning of date range)
- *PP_DATE_END* (End of date range)
- *PP_TIME_BEG* (Beginning time)
- *PP_TIME_END* (Ending time)
- *PP_SOURCE* (Application source)
- *PP_OFFICE* (Office code)
- *PP_SITE* (Site)
- -g (Process in batch mode)

Note: The Site parameter does not appear on the parameter screen. The value for this parameter is passed in the background, using the value of the end user's CARSSITE environment variable.

Elec App Reports

Electronic Application

Accesses: \$CARSPATH/modules/admit/reports/elecapp (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/elecapp

Parameters Passed:

- -f Formtype
- *PP_DATE_BEG* (Beginning of date range)
- *PP_DATE_END* (End of date range)
- *PP_TIME_BEG* (Beginning time)
- *PP_TIME_END* (Ending time)
- *PP_SOURCE* (Application source)
- *PP_RESERVE* (Application Reserve code)
- *PP_STATUS* (Application status)
- *PP_PROG* (Program)
- *PP_APPTMP_NO* (Application number)
- *PP_SITE* (Site)

Note: The Site parameter does not appear on the parameter screen. The value for this parameter is passed in the background, using the value of the end user's CARSSITE environment variable.

Elecapp Status

Accesses: \$CARSPATH/modules/admit/reports/elecapp (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/elecstat

Parameters Passed:

- -f Formtype
- *PP_DATE_BEG* (Beginning of date range)
- *PP_DATE_END* (End of date range)
- *PP_TIME_BEG* (Beginning time)
- *PP_TIME_END* (Ending time)
- *PP_SOURCE* (Application source)
- *PP_RESERVE* (Application Reserve code)
- *PP_STATUS* (Application status)
- *PP_PROG* (Program)
- *PP_SITE* (Site)

Note: The Site parameter does not appear on the parameter screen. The value for this parameter is passed in the background, using the value of the end user's CARSSITE environment variable.

Eleccapp DUP SSN

Accesses: \$CARSPATH/modules/admit/reports/elecscsn (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/elecscsn

Parameters Passed:

- -f Formtype
- PP_DATE_BEG (Beginning of date range)
- PP_DATE_END (End of date range)
- PP_TIME_BEG (Beginning time)
- PP_TIME_END (Ending time)
- PP_SOURCE (Application source)
- PP_RESERVE (Application Reserve code)
- PP_STATUS (Application status)
- PP_PROG (Program)
- PP_SITE (Site)

Note: The Site parameter does not appear on the parameter screen. The value for this parameter is passed in the background, using the value of the end user's CARSSITE environment variable.

Eleccapp Education Records

Accesses: \$CARSPATH/modules/admit/reports/elecscdtmp (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/elecscdtmp

Parameters Passed:

- -f Formtype
- PP_DATE_BEG (Beginning of date range)
- PP_DATE_END (End of date range)
- PP_TIME_BEG (Beginning time)
- PP_TIME_END (Ending time)
- PP_SOURCE (Application source)
- PP_RESERVE (Application Reserve code)
- PP_STATUS (Application status)
- PP_PROG (Program)
- PP_APPTMP_NO (Application number)
- PP_SITE (Site)

Note: The Site parameter does not appear on the parameter screen. The value for this parameter is passed in the background, using the value of the end user's CARSSITE environment variable.

Table Maintenance Menu

Admissions (A-Z) Menu

Advertising Medium

Accesses: \$CARSPATH/modules/admit/screens/tadvmedium

Menuopt File: \$CARSPATH/menuopt/admit/screens/tadvmedium

Parameters Passed: None

Advertising Medium Report

Accesses: \$CARSPATH/modules/admit/reports/tadvmedium
Menuopt File: \$CARSPATH/menuopt/admit/reports/tadvmedium
Parameters Passed: None

Advertisement Record

Accesses: \$CARSPATH/modules/admit/screens/adv
Menuopt File: \$CARSPATH/menuopt/admit/screens/adv
Parameters Passed: None

Advertisement Report

Accesses: \$CARSPATH/modules/admit/reports/adv
Menuopt File: \$CARSPATH/menuopt/admit/reports/adv
Parameters Passed:

- *PP_OFFICE* (Office code)
- *PP_DATE_BEG* (Beginning of date range)
- *PP_DATE_END* (End of date range)

Application Documents

Accesses: \$CARSPATH/modules/admit/screens/treqadmdoc
Menuopt File: \$CARSPATH/menuopt/admit/screens/treqadmdoc
Parameters Passed: None

Application Documents Rpt

Accesses: \$CARSPATH/modules/admit/reports/treqadmdoc
Menuopt File: \$CARSPATH/menuopt/admit/reports/treqadmdoc
Parameters Passed:

- *PP_PROG* (Program)
- *Category*

Decision

Accesses: \$CARSPATH/modules/admit/screens/tdec (PERFORM screen)
Menuopt File: \$CARSPATH/menuopt/admit/screens/tdec
Parameters Passed: None

Decision Report

Accesses: \$CARSPATH/modules/admit/reports/tdec (ACE report)
Menuopt File: \$CARSPATH/menuopt/admit/reports/tdec
Parameters Passed: None

Enrollment Status

Accesses: \$CARSPATH/modules/admit/screens/tenrstat (PERFORM screen)
Menuopt File: \$CARSPATH/menuopt/admit/screens/tenrstat
Parameters Passed: None

Enrollment Status Report

Accesses: \$CARSPATH/modules/admit/reports/tenrstat (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/tenrstat

Parameters Passed: None

Enrollment Sequence

Accesses: \$CARSPATH/modules/admit/screens/tenrseq (PERFORM screen)

Menuopt File: \$CARSPATH/menuopt/admit/screens/tenrseq

Parameters Passed: None

Enrollment Sequence Report

Accesses: \$CARSPATH/modules/admit/reports/tenrseq (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/tenrseq

Parameters Passed: None

Major

Accesses: \$CARSPATH/modules/admit/screens/tmajor

Menuopt File: \$CARSPATH/menuopt/admit/screens/tmajor

Parameters Passed: None

Major Report

Accesses: \$CARSPATH/modules/common/reports/tmajor

Menuopt File: \$CARSPATH/menuopt/common/reports/tmajor

Parameters Passed: None

Reference

Accesses: \$CARSPATH/modules/admit/screens/tref (PERFORM screen)

Menuopt File: \$CARSPATH/menuopt/admit/screens/tref

Parameters Passed: None

Reference Report

Accesses: \$CARSPATH/modules/admit/reports/tref (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/tref

Parameters Passed: None

Admissions Tape Conversion (A - Z)

ETS Country

Accesses: \$CARSPATH/modules/admit/screens/tleadctry (PERFORM screen)

Menuopt File: \$CARSPATH/menuopt/admit/screens/tleadctry

Parameters Passed: None

ETS Country Report

Accesses: \$CARSPATH/modules/admit/reports/tleadctry (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/tleadctry

Parameters Passed: None

ACT/ETS Denomination

Accesses: \$CARSPATH/modules/admit/screens/tleaddenom (PERFORM screen)

Menuopt File: \$CARSPATH/menuopt/admit/screens/tleaddenom

Parameters Passed: None

ACT/ETS Denomination Rep.

Accesses: \$CARSPATH/modules/admit/reports/tleaddenom (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/reports/tleaddenom

Parameters Passed: None

ACT/ETS Ethnic

Accesses: \$CARSPATH/modules/admit/screens/tleadethnc (PERFORM screen)

Menuopt File: \$CARSPATH/menuopt/admit/screens/tleadethnc

Parameters Passed: None

ACT/ETS Ethnic Report

Accesses: \$CARSPATH/modules/admit/reports/tleadethnc (ACE report)

Menuopt File: \$CARSPATH/menuopt/admit/report/tleadethnc

Parameters Passed: None

ACT/ETS Major

Accesses: \$CARSPATH/modules/admit/screens/tleadmaj

Menuopt File: \$CARSPATH/menuopt/admit/screens/tleadmaj

Parameters Passed: None

ACT/ETS Major Report

Accesses: \$CARSPATH/modules/admit/reports/tleadmaj

Menuopt File: \$CARSPATH/menuopt/admit/reports/tleadmaj

Parameters Passed: None

Major Report

Accesses: \$CARSPATH/modules/common/reports/tmajor

Menuopt File: \$CARSPATH/menuopt/common/reports/tmajor

Parameters Passed: None

PERFORM (Table Maintenance) Screens

Introduction

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

PERFORM Screens

The following PERFORM screens are available in Recruiting/Admissions.

Note: In the following list, descriptions of PERFORM screens include:

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

Menu access instructions assume you are beginning from the Recruiting/Admissions Main Menu.

Admissions Info

Not currently in use in CX, enables you to view or enter admissions information through a PERFORM screen rather than a program screen.

Menu Access: None

File: admit

Tables/Records Used:

- adm_rec
- cl_table
- ctc_table
- ctry_table
- dec_table
- enr_stat_table
- id_rec
- major_table
- prog_table
- ref_table
- sess_table
- st_table
- title_table

ACT/ETS Denomination Table

Lets you maintain the Denomination codes received through ACT, ETS student search, and examination tapes. The codes are used in the tape conversion process.

Menu Access: Table Maintenance: Admissions Tape (A-Z)

File: tleaddenom

Tables/Records Used:

- denom_table
- leaddenom_table

ACT/ETS Ethnic Table

Lets you maintain the Ethnic codes received through ACT, ETS student search, and examination tapes. The codes are used in the tape conversion process.

Menu Access: Table Maintenance: Admissions Tape (A-Z)

File: tleadethnc

Tables/Records Used:

- ethnic_table
- leadethnic_table

ACT/ETS Major Table

Lets you maintain the Lead Major codes received through ACT/ETS student search and examination tapes. These codes are used in the tape conversion process.

Menu Access: Table Maintenance: Admissions Tape (A-Z)

File: tleadmaj

Tables/Records Used:

- leadmaj_table
- major_table

Admission Record

Lets you view admissions, program enrollment, student academic, contact, and hold information in Query by Form.

Menu Access: Query by Form Menu: Statuses

File: statuses

Tables/Records Used:

- acad_stat_table
- adm_rec
- ctc_rec
- ctc_table
- enr_stat_table
- hold_act_table
- hold_rec
- hold_table
- id_rec
- major_table
- ofc_table
- prog_enr_rec
- stu_acad_rec
- tick_table

Admissions Record Query

Lets you view admissions information in Query by Form.

Menu Access: Query by Form Menu: Admissions

File: admit_q

Tables/Records Used:

- adm_rec
- cl_table
- ctc_table
- ctry_table
- dec_table
- enr_stat_table
- id_rec
- major_table
- prog_table
- ref_table

- sess_table
- st_table
- title_table

Admissions Statistic Record

Lets you view Admission Statistic records using Query by Form.

Menu Access: Query by Form Menu: Admissions Statistic

File: admstats

Tables/Records Used:

- adm_stat_rec

Admission Status Contact

Used within Communications Management, lets you create a single Contact record for an applicant.

Menu Access: Communications Management: Create One Contact Record

File: statusctc

Tables/Records Used:

- ctc_rec
- ctry_table
- id_rec
- st_table
- title_table

Note: For more information, see *Communications Management Technical Manual*.

Advertisement Record

Used to track all advertising efforts.

Menu Access: Table Maintenance: Admissions (A-Z) Menu: Advertisement Record

File: adv

Tables/Records Used:

- adv_rec
- id_rec
- adv_medium_table
- ctry_table
- ofc_table
- ref_table
- st_table
- title_table

Advertising Medium Record

Used to maintain codes for different advertising mediums used to promote ????. These codes are used to populate the Advertising Medium field within the Advertisement record.

Menu Access: Table Maintenance: Admissions (A-Z) Menu: Advertising Medium

File: tadvmedium

Tables/Records Used:

- adv_medium_table

Application Documents

Used to populate the Application Document table, Required Contact table, Required Examination table, and the Required Recommendation table.

Menu Access: Table Maintenance: Admissions (A-Z) Menu: Application Documents

File: reqadmdoc

Tables/Records Used:

- reqadmdoc_table
- prog_table
- reqctc_table
- ctc_table
- tick_table
- reqrecom_table
- reqexam_table
- exam_table

Call Entry Result Table

Lets you maintain the Call Entry Call Result codes.

Menu Access: Call Entry: Reports Menu: Call Result Table

File: tcallres

Tables/Records Used:

- callresult_table

Caller Time Entry

Lets you enter and update caller's hours and minutes worked.

Menu Access: Call Entry Menu: Caller Time Entry

File: callstat

Tables/Records Used:

- id_rec
- call_rec

Decision Table

Lets you maintain the Decision table.

Note: The Active Date and Inactive Date fields are used only if the macro ENABLE_FEAT_END_DATE is set to Y.

Menu Access: Table Maintenance: Admissions (A-Z)

File: tdec

Tables/Records Used:

- acad_stat_table
- dec_table

Enrollment Status

Lets you view individual Enrollment Status records using Query by Form.

Menu Access: Query by Form Menu: Enrollment Status

File: enrstat

Tables/Records Used:

- adm_rec
- cl_table
- ctc_table
- ctry_table
- enr_stat_rec
- enr_stat_table
- id_rec
- major_table

- prog_table
- ref_table
- sess_table
- st_table
- title_table

Enrollment Sequence Table

Lets you maintain the Enrollment Sequence table.

Note: The Active Date and Inactive Date fields are used only if the macro ENABLE_FEAT_END_DATE is set to Y.

Menu Access: Table Maintenance: Admissions (A-Z)

File: tenrseq

Tables/Records Used:

- enr_seq_table
- enr_stat_table

Enrollment Status Table

Lets you maintain the Enrollment Status table.

Note: The Active Date and Inactive Date fields are used only if the macro ENABLE_FEAT_END_DATE is set to Y.

Menu Access: Table Maintenance: Admissions

File: tenrstat

Tables/Records Used:

- enr_stat_table

ETS Country Table

Lets you maintain the codes in the ETS Country table. These codes are used in the ETS Student Search and SAT tape conversion process.

Menu Access: Table Maintenance: Admissions Tape (A-Z)

File: tleadctry

Tables/Records Used:

- ctry_table
- leadctry_table

Lead Entry

Lets you view, add, and update lead and lead contact information.

Menu Access: Lead Processing: Lead Entry

File: leade

Tables/Records Used:

- ctc_table
- ctry_table
- ethnic_table
- id_rec
- ldctc_rec
- lead_rec
- major_table
- ref_table
- sch_table
- sess_table

- st_table
- title_table

Major Table

Allows the Admissions office to update the Letter Text field and the Department field only.

Menu Access: Table Maintenance: Admissions (A-Z) Menu: Major

File: tmajor

Tables/Records Used:

- major_table
- dept_table

Reference Table

Lets you maintain the sources of references for students.

Menu Access: Table Maintenance: Admissions

File: tref

Tables/Records Used:

- ref_table

SQL Statements

Introduction

The Recruiting/Admissions product uses SQL statements that perform queries and then add, delete or update database records. The statements are located in the directory path \$CARSPATH/modules/admit/informers.

SQL Statements

The following lists the SQL statements provided with Recruiting/Admissions.

Admission Document Tracking Contacts - All Students

Adds admission document tracking records in a batch process for all students with a specified contact tickler, academic program, intended session and year of enrollment, admission status, and document category. This option automatically adds incoming document Contact records with expected contact statuses, Exam records with zero scores, and Recommendation records with expected statuses. This option is useful for professional schools who use a data conversion script to load data file provided by an admissions service (i.e., AMCAS for medical schools), and then need to add Document records after the data file has been loaded into the CX database.

Note: The Application Document tables need to be defined before this SQL statement can be run.

Menu Access: Session Processing: Add Document Contacts

File: admdocctc

Tables/Records Used:

- adm_rec
- ctc_rec
- ctc_table
- exam_rec
- intvwrecom_rec
- reqadmdoc_table
- reqexam_table
- reqrecom_table

Admission Document Tracking Contacts - One Student

Adds admission document tracking contacts for one student for a specified program and tickler.

Note: This SQL statement may need to be customized for each site. If you customize it, check for duplications before the first addition.

Menu Access: This SQL statement does not run from a menu option. It automatically executes when the Add App Doc field (adm_rec.adm_doc) on the Application screen is set to Y.

File: admdocone

Tables/Records Used:

- adm_rec
- ctc_rec
- ctc_table

Contact Records for Prospects at High Schools/Colleges Scheduled for a Recruitment Trip

Adds expected Contact records for all prospects who currently attend a high school or college matching a specified school ID number, have been scheduled for a recruitment trip (using the Schedule Entry program), and have not already been denied admission. This option does not add a Contact record if a student already has an expected contact by the same contact and tickler code, has been flagged as incorrectly addressed, or has been flagged as being deceased.

Note: The prospect must have an id_rec, adm_rec, and ed_rec to have a Contact record added.

Menu Access: Admissions Processing: Add Contacts/School Visit

File: colldayctc

Tables/Records Used:

- adm_rec
- ctc_rec
- ed_rec
- id_rec
- sch_rec

Contact Records for Prospects with Missing Documents

Queries the database for applicants with missing documents and adds an expected contact used to create letters with the *Itrincapp* or Itrmistran ACE report. This option does not add a Contact record if a student already has an expected contact by the same contact and tickler code, has been flagged as incorrectly addressed, or has been flagged as being deceased.

Menu Access: None

File: incapp

Tables/Records Used:

- adm_rec
- ctc_rec
- ctc_table
- ed_rec
- exam_rec
- intvwrecom_rec

Lead Contacts for Prospects

Adds Lead Contact records with an expected contact status for all prospects meeting the specified search criteria. This option does not add a Contact record if a student already has an expected lead contact by the same contact and tickler code, or has been flagged as an invalid lead.

Menu Access: Lead Processing Menu: Select by User Parameters

File: leadctc

Tables/Records Used:

- lead_rec
- leadctc_rec

Move Leads to Admissions File

Copies prospect data from the Lead record to the ID record, Profile record, Admissions record and other pertinent records.

Menu Access: Lead Processing Menu: Move Leads to Inquired

File: moveleads

Tables/Records Used:

- acad_cal_rec
- adm_rec
- ed_rec
- id_rec
- lead_rec
- profile_rec
- site_rec
- tick_rec

Pre-Application Documents

Queries the database for inquirers who have had any type of document sent to the admissions office before sending an application. If found, it creates an expected Contact record that can be used with the *ltrpreapp* ACE report to create a letter to the inquirer, listing documents that have been received. This option does not add a Contact record if a student already has an expected contact by the same contact and tickler code, has been flagged as incorrectly addressed, or has been flagged as being deceased.

Menu Access: Communications Management: Select Pre-Applicants

File: preapp

Tables/Records Used:

- adm_rec
- ctc_rec
- ctc_table
- ed_rec
- exam_rec
- intvwrecom_rec

Contact Records for Final Status

Depending on the Admissions office at your college/university, admission statuses such as ENROLLED (enrolled in classes) or WITHDREW (withdrew application from consideration) or DENIED (denied admissions) or DECLINED (declined offer of acceptance) can all be considered final statuses. At the end of each recruitment term, it is advisable that no students be left with a status that is in the middle of the Admissions Funnel, (i.e., APPLIED). Students who never applied for admission can be left with an INQUIRED status. This menu option adds Contact records used by the ADMSTATS program to update students to a final admission status tracked by the admissions office. For example, your admissions office may use statuses ACCEPTED, DEPOSIT and ENROLLED. When this menu option is run an ENROLLED Contact record is added in batch (with a completed contact status), for all students who have a current admission status of DEPOSIT and are enrolled in the college/university. At the same time this menu option can add a WITHDREW contact in batch (with a completed contact status), for all students who have a current admission status of DEPOSIT and are not enrolled in the college/university. This menu option should be run approximately one week after the beginning of each quarter/semester. After this menu option has run the ADMSTATS program should be run to update the students' current status from DEPOSIT to ENROLLED. Reports which can be reviewed before and after this menu option is run include Students Not Registered, Current Statuses, and Counts & Subtotals.

Note: Make sure the macros ADM_STAT_CONF, ADM_STAT_WITHDREW, ADM_STAT_ENR and ADM_STAT_NOSHOW are defined and installed, and the regctc informer and menuopt files are reinstalled before this informer is run.

Menu Access: Session Processing: Update to Final Status

File: regctc

Tables/Records Used:

- adm_rec

- ctc_rec
- prog_table
- stu_acad_rec

Admission Document Tracking Contacts - One Student

Adds admission document tracking records for one student for a specified academic program and contact tickler. This option automatically adds incoming document Contact records with expected contact statuses, Exam records with zero scores, and Recommendation records with expected statuses.

Note: The Application Document tables need to be defined before this SQL statement can be run.

Menu Access: This SQL statement does not run from a menu option. It automatically executes when the Add App Doc field (adm_rec.adm_doc) on the Application screen is set to Y, and a document category code (adm_rec.doc_ctgry) has been selected.

File: reqadmdoc

Tables/Records Used:

- adm_rec
- ctc_rec
- ctc_table
- exam_rec
- intvwrecom_rec
- reqadmdoc_table
- reqctc_table
- reqexam_table
- reqrecom_table

Remove Leads

Selectively removes Lead records, Lead Tickler records, and Lead Contact records based on the lead_rec.add_date or the lead_rec.move_date. By entering a date in the Add Date parameter and leaving the Move Date parameter blank, you delete records of leads that never responded to your recruitment efforts, and keep the leads who did respond. By entering a date in the Move Date parameter and leaving the Add Date parameter blank, you delete records for leads that did respond to your recruitment efforts, but not delete records for leads that did not respond. Entering a date in both the Add Date and Move Date parameters deletes records for both respondents and non-respondents at the same time. Both date parameters delete Lead records where the date value is on or before the date the Lead records were added/moved.

Menu Access: Lead Processing Menu: Remove Specific Leads

File: rmlleads

Tables/Records Used:

- ldctc_rec
- ldtick_rec
- lead_rec

Set Counselor Territories

Selects schools by various parameters and updates the sch_rec.cnslr_id field to indicate which admissions counselor is responsible for recruiting students from the selected schools.

Menu Access: Admissions Session Processing Menu: Set Counselor Territories

File: setcnslr

Tables/Records Used:

- id_rec
- sch_rec

Contact Records for Prospects

Adds expected Contact records for prospects who meet the specified criteria. This option does not add a Contact record if a student already has an expected contact by the same contact and tickler code, is flagged as incorrectly addressed, or is flagged as being deceased.

Menu Access: Communications Management: Select by User Parameters

File: studctc

Tables/Records Used:

- accomp_rec
- adm_rec
- ctc_rec
- ed_rec
- id_rec
- int_rec
- involve_rec
- profile_rec

Reassign Counselors

Updates the adm_rec.cnslr_id field in order to reassign students with the ID number of a new admission counselor who has taken over a territory of a past admission counselor. Caution should be used when running this menu option. Carefully choose which Admission records you want to update, and (possibly just as important) which records you don't want to update. For example, you may only want to update the Admission Counselor ID for all undergraduate students who are enrolling for the fall term of 2000, and then update the Admission counselor ID field for all terms after the fall 2000 term. In this scenario the you need to run the Reassign Counselors menu option twice. The first time the user passes the undergraduate Program code for the program parameter, "= 2000" for the year parameter, the fall Term code in the first session parameter (leaving the other three session parameters blank, and then entering the ID numbers of the past and new admission counselors. The second time the user passes the same undergraduate Program code for the program parameter, ">= 2001" for the year parameter, all the Session codes for each of the terms within an academic year in the four session parameters, and the ID numbers of the past and new admissions counselor. If your school has more than four sessions within an academic year, you need to run this menu option a third time for any remaining sessions. If an admission counselor is assigned to both undergraduate and graduate students, you need to run this menu option again to reassign the counselor for graduate students.

Menu Access: Session Processing Menu: Reassign Counselors

File: updcnslr

Tables/Records Used:

- adm_rec

Csh Scripts

Introduction

Recruiting/Admissions uses Csh scripts to automate the processing of information. Csh scripts are UNIX-based program statements that can execute a series of SQL scripts or reports. They can also perform other types of functions, including updating files. The Recruiting/Admissions Csh scripts are located in the \$CARSPATH/modules/admit/scripts directory path.

Csh Scripts

The following list associates a Recruiting/Admissions menu option with the corresponding Csh script and provides a description of the script.

Note: In the following list, 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 statements used, if applicable
- A list of tables used, if applicable

admprocess

Runs the following processes (some processes are disabled in the standard CX system; check the comments in the *admprocess* file for instructions on enabling processes or selecting the processes that are applicable to your institution):

- *infage* (SQL script that updates the Age field in Profile records)
- *admstats* (program that updates the status of applicants based on contacts received and sent)
- *tickler* (program that schedules contacts)
- *lpsctcs* (ACE report that displays letters or labels ready for creation)
- *lpsrun* (program that creates all expected letters and labels)
- *admstats* (program that updates the status of applicants based on contacts received and sent; the *admprocess* script runs this process again to update statuses based on letters just created by *lpsctcs* and *lpsrun*)
- *setwaitrnk* (ACE report that updates waiting lists for admission by program)
- *setmajrnk* (ACE report that updates multiple waiting lists for admission by major within a program)

admstats

Runs the *admstats* program that updates the status of applicants based on contacts received and sent.

chprosp

Executes the following two church reports:

- *chprof* (ACE report that provides general profile information about a church)
- *chprosp* (ACE report that lists prospects who are members of a specific church)

Menu Access: Profile Reports: Students by Church

chreport

Executes the following three church reports:

- *chprof* (ACE report that provides general profile information about a church)
- *chprosp* (ACE report that lists prospects who are members of a specific church)
- *chrecgrads* (ACE report that lists recent graduates of your institution who attended a specific church)

Menu Access: Profile Reports: Church Profile

hsprosp

Executes the following two high school reports:

- *hsprof* (ACE report that provides general profile information about a high school)
- *hsprosp* (ACE report that lists prospects who attend a specific high school)

Menu Access: Profile Reports: Students by School

hsreport

Executes the following high school reports:

- *hsprof* (ACE report that provides general profile information about a high school)
- *hslead* (ACE report that lists leads who attend a specific high school)
- *hsprosp* (ACE report that lists prospects who attend a specific high school)
- *hscurrent* (ACE report that lists current students at your institution who attended a specific high school)
- *hsrecgrads* (ACE report that lists recent graduates of your institution who attended a specific high school)
- *hsfaculty* (ACE report that lists faculty at a specified high school who attended your institution)

Menu Access: Profile Reports: School Profile

runcnslr

Creates profile sheets for admissions counselors by running the *runcnslr* report.

Menu Access: Counselor Reports: Students - Scheduled Visits

tpconv

Calls the *tpconvert* program to report on and convert American College Testing (ACT) and Educational Testing Service (ETS/SAT) data.

Menu Access: All Tape Conversion options from the Recruiting/Admissions Main menu

Reports

Introduction

CX contains ACE reports (as well as other types of reports) for automated reporting of Recruiting/Admissions database information. All the ACE reports reside in either of the following directory paths:

- \$CARSPATH/modules/admit/reports
- \$CARSPATH/modules/admit/others

This section provides reference information about the reports. In most cases, report information is sorted by the report title, that is, the name that prints at the top of the report. However, in some cases (e.g., where the report does not produce a single hard copy output), the listed title is a description of the output. In these instances, the report title appears in parentheses.

ACE Reports for Letter Preparation

CX also uses ACE reports to extract data for letters. For example, the *ltradmit* ACE report selects data from admissions records and matches the data with macros (e.g., the *ltradmit* report extracts names and Major codes and matches them with the macros WP_FULLNAME and WP_MAJOR respectively. The macros are then used to customize form letters).

CX offers a variety of these data-extracting ACE reports, which reside in the \$CARSPATH/modules/admit/reports directory along with other traditional reports. Because they are not typical printed reports (but are used instead for letter production), more information about these reports appears separately in *ACE Reports for Letter Creation* in this manual, and also in *Communications Management Technical Manual*.

The following ACE reports in the /reports directory extract data for letters:

- *ltradmbulk*
- *ltradmit*
- *ltrenclose*
- *ltrhsg*
- *ltrincapp*
- *lrintconf*
- *ltrlead*
- *lrmistran*
- *ltrnoltr*
- *ltrparent*
- *ltrparent2*
- *ltrpreapp*
- *ltrschlabl*
- *ltrschvst*
- *ltrstat*
- *ltrstat2*
- *ltrstat_pa*
- *ltrstatg*
- *ltrtrans*

Printed ACE Reports in the Reports Directory

The following lists the Recruiting/Admissions ACE reports stored in \$CARSPATH/modules/admit/reports. Some reports listed do not appear on the CX menu system because they are used only in Csh scripts, or because they represent a report alternative not currently defined through a menu option.

A Counselor's Schools and Prospects

Produces a list of schools assigned to a specified counselor and all the students from each school for a specified program, sessions, year and admission status.

Menu Access: Reports Menu: Counselor Reports: Counselor School/Prospect

File: \$CARSPATH/modules/admit/reports/cnslrsch

ACT Composite Score Report

Produces an ACT exam profile report.

Menu Access: Reports Menu: Statistical Reports: ACT Profile vs. HS Rank

File: \$CARSPATH/modules/admit/reports/actprofrnk

ACT/ETS Denomination Table Report

Lists the values in the leaddenom_table.

Menu Access: Table Maintenance: Admissions Tape Conversion A-Z: ACT/ETS Denomination Rep.

File: \$CARSPATH/modules/admit/reports/tleaddenom

ACT/ETS Ethnic Table Report

Lists the values in the leadethnc_table.

Menu Access: Table Maintenance: Admissions Tape Conversion A-Z: ACT/ETS Ethnic Report

File: \$CARSPATH/modules/admit/reports/tleadethnc

ACT/ETS Major Table Report

Lists the values in the leadmaj_table.

Menu Access: Table Maintenance: Admissions Tape Conversion A-Z: ACT/ETS Major Report

File: \$CARSPATH/modules/admit/reports/tleadmaj

Admissions Electronic Application

Prints an electronically submitted application.

Menu Access: Electronic Applications: Elec App Reports: Elecapp Application

File: \$CARSPATH/modules/admit/reports/elecapp

Admissions Electronic Application: Temporary Education Records Only Report

Produces a report of the *elecapp* Education records by either date range or specific application number.

Menu Access: Reports Menu: Elec App Reports - Elecapp Education Records

File: \$CARSPATH/modules/admit/reports/elecedtmp

Admissions Statistics Report

Prints major totals in summarized or non-summarized form, based on enr_stat_rec values. Produces results only for institutions using *admstats*.

Menu Access: Reports Menu: Statistical Reports: Enroll History - Years

File: \$CARSPATH/modules/admit/reports/enrhistyr

Admissions Status Sequences

Maintains counts of the number of applicants who change from one status to another.

Menu Access: Reports Menu: Statistical Reports: Status Sequences

File: \$CARSPATH/modules/admit/reports/seqcount

Advertising Medium Report

Lists the values in the advmedium_table.

Menu Access: Table Maintenance Menu: Admissions (A-Z) menu: Advertising Medium Report

File: \$CARSPATH/modules/admit/reports/tadvmedium

Advertising Record Report

Lists the values in the adv_rec.

Menu Access: Table Maintenance Menu: Admissions (A-Z) menu: Advertisement Report

File: \$CARSPATH/modules/admit/reports/adv

Applicant Committee Review

Produces an executive summary sheet you can use to review applicant data during admission committee review meetings.

Menu Access: Reports Menu: Profile Reports: Applicant Committee Review

File: \$CARSPATH/modules/admit/reports/committee

Calendar for Admissions Counselors

Produces a schedule calendar for a given recruiter.

File: \$CARSPATH/modules/admit/reports/schdcal

Call Contact Schedule

Produces a list of calls due.

Menu Access: Call Entry: Reports Menu: Call Contact Schedule

File: \$CARSPATH/modules/admit/reports/callctcs

Call Entry Contact Table Report

Prints the values in the Contact table that relate to call entry.

Menu Access: Call Entry: Reports Menu: Call Entry Contact Table Report

File: \$CARSPATH/modules/admit/reports/tcalltmctc

Call Entry Information Report

Produces a report of call entry activity.

Menu Access: Call Entry: Reports Menu: Call Entry Information

File: \$CARSPATH/modules/admit/reports/callinfo

Caller Time Report

Produces a report of the recorded amount of time worked by a caller for a specific date range.

Menu Access: Call Entry: Reports Menu: Caller Time Entry

File: \$CARSPATH/modules/admit/reports/calltime

Calls Completed

Produces a report of the outcome of call entry calls.

Menu Access: Call Entry: Reports Menu: Calls Completed

File: \$CARSPATH/modules/admit/reports/callmktctc

(Church Profile Report - Part 1)

Produces part of a profile listing for churches.

File: \$CARSPATH/modules/admit/reports/chprof

(Church Profile Report - Part 2)

Produces part of a profile listing for churches.

File: \$CARSPATH/modules/admit/reports/chprosp

(Church Profile Report - Part 3)

Produces part of a profile listing for churches.

File: \$CARSPATH/modules/admit/reports/chcurrent

(Church Profile Report - Part 4)

Produces part of a profile listing for churches.

File: \$CARSPATH/modules/admit/reports/chrecgrads

Composite Score Report

Produces an exam freshman profile.

Menu Access: Reports Menu: Statistical Reports: ACT Composite Scores

File: \$CARSPATH/modules/admit/reports/actprof

Contact Report for All Students

Produces list of all admissions contacts with selected recruits.

Menu Access: Reports Menu: Contact Reports: Contacts for All Students

File: \$CARSPATH/modules/admit/reports/allctc

Contact Report for One Student

Prints a list of all admissions contacts with a specified prospect. Since the report extracts information by ID, the prospect must have an id_rec on the database.

Menu Access: Reports Menu: Contact Reports: Contacts for One Student

File: \$CARSPATH/modules/admit/reports/onetctc

Contact Resource Summary Report

Prints a summary of resources and responses.

Menu Access: Reports Menu: Statistical Reports: Contact Resources

File: \$CARSPATH/modules/admit/reports/resrcappl

Contacts Grouped by Counselor

Produces a list of all contacts made within a specified period, sorted by counselor.

Menu Access: Reports Menu: Contact Reports: Contacts by Counselors

File: \$CARSPATH/modules/admit/reports/cnslrctc

(Counselor Reports)

Runs the *profch* and *profhs* reports. The *runcnslr* script executes this report.

File: \$CARSPATH/modules/admit/reports/runcnslr

Current Admissions Statuses

Produces a list of prospects and their statuses.

Menu Access: Reports Menu: Statistical Reports: Current Statuses

File: \$CARSPATH/modules/admit/reports/admcurstat

Daily Stats by Caller

Produces a report of daily call activity, sorted by caller ID.

Menu Access: Call Entry: Reports Menu: Daily Stats by Caller

File: \$CARSPATH/modules/admit/reports/callstatd

Decision Table Report

Lists the entries in the dec_table.

Menu Access: Table Maintenance: Admissions A-Z: Decision Report

File: \$CARSPATH/modules/admit/reports/tdec

Elecapp Duplicate SSN Report

Lists duplicate existing ID information before adding information from an electronic application to the database. Information is included where the id_rec.ss_no is equivalent to apptmp_rec.ss_no.

Menu Access: Reports Menu: Elec App Reports - Elecapp DUP SSN

File: \$CARSPATH/modules/admit/reports/elecassn

Elecapp Status Report

Shows the status related to each apptmp_rec on the system.

Menu Access: Reports Menu: Elec App Reports: Elecapp Status

File: \$CARSPATH/modules/admit/reports/elecstat

Enrollment Sequence Table Report

Lists the entries in the enr_seq_table.

Menu Access: Table Maintenance: Admissions A-Z: Enrollment Sequence Report

File: \$CARSPATH/modules/admit/reports/tenrseq

Enrollment Status History Report

Prints major totals in summarized or non-summarized form, based on enr_stat_rec values. Produces results only for institutions using *admstats*.

Menu Access: Reports Menu: Statistical Reports: Enroll History - Percent

File: \$CARSPATH/modules/admit/reports/enrhistpct

Enrollment Status Table Report

Lists the entries in the enr_stat_table.

Menu Access: Table Maintenance: Admissions A-Z: Enrollment Status Report

File: \$CARSPATH/modules/admit/reports/tenrstat

ETS Country Table Report

Lists the entries in the leadctry_table.

Menu Access: Table Maintenance: Admissions Tape Conversion A-Z: ETS Country Report

File: \$CARSPATH/modules/admit/reports/tleadctry

Geographical Distribution

Produces a count of applicants from each state (or country).

Menu Access: Reports Menu: Statistical Reports: Geographical Distribution

File: \$CARSPATH/modules/admit/reports/geography

(High School Profile Report - Part 1)

Produces part of a profile listing for high schools.

File: \$CARSPATH/modules/admit/reports/hsprof

(High School Profile Report - Part 2)

Produces part of a profile listing for high schools.

File: \$CARSPATH/modules/admit/reports/hsprosp

(High School Profile Report - Part 3)

Produces part of a profile listing for high schools.

File: \$CARSPATH/modules/admit/reports/hscurrent

(High School Profile Report - Part 4)

Produces part of a profile listing for high schools.

File: \$CARSPATH/modules/admit/reports/hsfaculty

(High School Profile Report - Part 5)

Produces part of a profile listing for high schools.

File: \$CARSPATH/modules/admit/reports/hsrecgrads

(High School Profile Report - Part 6)

Produces part of a profile listing for high schools.

File: \$CARSPATH/modules/admit/reports/hslead

Inclusive Admissions Statuses

Produces a total count of each type of application status (e.g., not accepted or reserved with deposit).

Menu Access: Reports Menu: Statistical Reports: Inclusive Statuses

File: \$CARSPATH/modules/admit/reports/curstat

Intended Majors by Applicants

Summarizes majors for applicants, sorted by current enrollment status.

File: \$CARSPATH/modules/admit/reports/majsum

Interview Report

Produces a list students' interview records. This report is used to print out a schedule of interviews for a selected time period.

Menu Access: Reports Menu: Counselor Reports: Scheduled Interviews

File: \$CARSPATH/modules/admit/reports/interview

Lead Contact Action Report

This report lists expected lead contact records that are to be used for letter/label creation.

Menu Access: Lead Processing: Lead Communications Management Menu: Lead Letters/Labels Due

File: \$CARSPATH/modules/admit/reports/leadctcdue

Leads by Contact Resource Report

This report prints the counts and percentages of how many leads responded and did not respond to a recruitment mailing. The counts and percentages are broken down by name of mailing.

Menu Access: Lead Reports menu: Leads By Contact Resource

File: \$CARSPATH/modules/admit/reports/leadresrc

Leads by Major Report

This report prints the counts and percentages of how many leads responded and did not respond to recruitment mailing(s). The counts and percentages are broken down by first choice major.

Menu Access: Lead Reports menu: Leads By Major

File: \$CARSPATH/modules/admit/reports/leadmaj

Leads by Reference Source Report

This report prints the counts and percentages of how many leads responded and did not respond to recruitment mailings. The counts and percentages are broken down by how your institution acquired names and addresses of leads, (ACT vs. ETS Student Search tapes).

Menu Access: Lead Reports menu: Leads By Reference Source

File: \$CARSPATH/modules/admit/reports/leadsrc

Leads by School Report

This report prints the counts and percentages of how many leads responded and did not respond to a recruitment mailing. The counts and percentages are broken down by school of attendance.

Menu Access: Lead Reports menu: Leads By School

File: \$CARSPATH/modules/admit/reports/leadceeb

Leads by Sex Report

This report prints the counts and percentages of how many leads responded and did not respond to a recruitment mailing. The counts and percentages are broken down by student gender.

Menu Access: Lead Reports menu: Leads By Sex

File: \$CARSPATH/modules/admit/reports/leadsex

Leads by State Report

This report prints the counts and percentages of how many leads responded and did not respond to recruitment mailings. The counts and percentages are broken down by state.

Menu Access: Lead Reports menu: Leads By State

File: \$CARSPATH/modules/admit/reports/leadst

Leads by Zip Report

This report prints the counts and percentages of how many leads responded and did not respond to recruitment mailings. The counts and percentages are broken down by zip code.

Menu Access: Lead Reports menu: Leads By Zip

File: \$CARSPATH/modules/admit/reports/leadzip

Major by Current Statuses

Prints major totals by current admissions enrollment status.

Menu Access: Reports Menu: Statistical Reports: Majors - Current Statuses

File: \$CARSPATH/modules/admit/reports/major

Major by Inclusive Statuses (or Major Report if not summarized)

Prints major totals in summarized or non-summarized form, based on enr_stat_rec values. Produces results only for institutions using *admstats*.

Menu Access: Reports Menu: Statistical Reports: Inclusive Statuses

File: \$CARSPATH/modules/admit/reports/enrmajor

Make Waiting List Current

Updates the admissions wait list by majors.

Menu Access: Admissions Processing: Update Waitlist by Major

File: \$CARSPATH/modules/admit/reports/setmajrnk

Make Waiting List Current

Updates the admissions wait list regardless of major.

Menu Access: Admissions Processing: Update Waitlist

File: \$CARSPATH/modules/admit/reports/setwaitrnk

Other Schools Competing for Applicants

Produces a list of all schools that students chose to attend instead of your institution. This report also prints a count of the number of students lost to each competitive school.

Menu Access: Reports Menu: Contact Reports: Competitive Schools

File: \$CARSPATH/modules/admit/reports/cmptvschl

Profile Info for Students by Church

Produces a counselor recruit card. The *runcnslr* report runs this report.

File: \$CARSPATH/modules/admit/reports/profch

Profile Info for Students by School

Produces all information for all recruits for a particular high school.

File: \$CARSPATH/modules/admit/reports/profhs

Recruit Profiles

Produces a counselor recruit card by student major.

File: \$CARSPATH/modules/admit/reports/profone

Recruit Profiles

Produces a counselor recruit card.

File: \$CARSPATH/modules/admit/reports/profcnslr

Recruit Profiles

Produces a counselor recruit card by student interests.

File: \$CARSPATH/modules/admit/reports/profint

Recruit Profiles

Produces a counselor recruit card by student major.

File: \$CARSPATH/modules/admit/reports/profmaj

Recruit Profiles

Produces a counselor recruit card by student home zip code.

File: \$CARSPATH/modules/admit/reports/profzip

Recruit Status Summary Report

Produces a summary of prospects' statuses.

File: \$CARSPATH/modules/admit/reports/enrstat

Recruit Status Report by State

Produces a report of recruits by state and enrollment status.

File: \$CARSPATH/modules/admit/reports/recstate

Reference Table Report

Lists the entries in the *ref_table*.

Menu Access: Table Maintenance: Admissions A-Z: Reference Report

File: \$CARSPATH/modules/admit/reports/tref

Required Application Documents

Lists the entries in the *reqadmdoc_table*.

Menu Access: Table Maintenance Menu: Admissions (A-Z) menu: Application Documents Rpt

File: \$CARSPATH/modules/admit/reports/treqadmdoc

Responses by Contact Resources

Produces a tally of responses by contact type.

Menu Access: Lead Processing: Reports Menu: Leads by Contact Resource

File: \$CARSPATH/modules/admit/reports/leadresrc

Responses by Major

Produces a report of responses from leads, including percentages, sorted by major.

Menu Access: Lead Processing: Reports Menu: Leads by Major

File: \$CARSPATH/modules/admit/reports/leadmaj

Responses by Sex

Produces a report of responses from leads, including percentages, sorted by sex.

Menu Access: Lead Processing: Reports Menu: Leads by Sex

File: \$CARSPATH/modules/admit/reports/leadsex

Responses by School

Produces a report of responses from leads, sorted by school.

Menu Access: Lead Processing: Reports Menu: Leads by School

File: \$CARSPATH/modules/admit/reports/leadceeb

Responses by Source of Reference

Produces a report of responses from leads, sorted by reference source.

Menu Access: Lead Processing: Reports Menu: Leads by Reference

File: \$CARSPATH/modules/admit/reports/leadsrc

Responses by State

Produces a report of responses from leads, sorted by state.

Menu Access: Lead Processing: Reports Menu: Leads by State

File: \$CARSPATH/modules/admit/reports/leadst

Responses by Zip Code

Produces a report of responses from leads, sorted by zip code.

Menu Access: Lead Processing: Reports Menu: Leads by Zip

File: \$CARSPATH/modules/admit/reports/leadzip

Results Table - Call Entry

Lists the table entries in the callresult_table.

Menu Access: Call Entry: Reports Menu: Call Result Table Report

File: \$CARSPATH/modules/admit/reports/tcallres

SAT Composite Score Report

Produces an SAT exam profile report.

Menu Access: Reports Menu: Statistical Reports: SAT Profile vs. HS Rank

File: \$CARSPATH/modules/admit/reports/satprofrnk

School Summary Report

Reports the source of applicants for the given session and year.

Menu Access: Reports Menu: Statistical Reports: Summary by Schools

File: \$CARSPATH/modules/admit/reports/collsrc

Source of Reference by Current Statuses

Prints source of reference totals (either summarized or non-summarized).

Menu Access: Reports menu: Statistical Reports: Current Statuses

File: \$CARSPATH/modules/admit/reports/refsrcenr

Source of Reference by Inclusive Statuses

Prints source of reference totals in summarized or non-summarized form, based on enr_stat_rec values. Produces results only for institutions using *admstats*.

Menu Access: Reports Menu: Statistical Reports: Inclusive Statuses

File: \$CARSPATH/modules/admit/reports/enrrefsr

Source of Reference Summary Report

Prints source of reference totals (either summarized or non-summarized).

Menu Access: Reports Menu: Statistical Reports: Reference Sources

File: \$CARSPATH/modules/admit/reports/refsr

Student Visit Agenda

This report lists an agenda for a student visit to campus.

Menu Access: Reports Menu: Counselor Reports: Student Visit Agenda

File: \$CARSPATH/modules/admit/reports/stuvst

Students by Age and Sex - Full and Part Time

Summarizes sex and age information for both full- and part-time students.

Menu Access: Reports Menu: Statistical Reports: Students by Sex and Age

File: \$CARSPATH/modules/admit/reports/sexage

Students Eligible for Admissions Committee Review

Prints a list of recruits who are eligible for decision.

Menu Access: Contact Reports: Eligible for Decision

File: \$CARSPATH/modules/admit/reports/elig_dec

Students Eligible for Admissions Committee Review

Prints a list of recruits who are eligible for decision. Not currently in use.

File: \$CARSPATH/modules/admit/reports/appcomplt

Students Eligible for Admissions Committee Review

Produces a list of recruits who are eligible for decision. These recruits have a Contact record of Complete, but do not have a Contact record of Decision. Not currently in use.

File: \$CARSPATH/modules/admit/reports/appcomp_pa

Students Not Eligible for Admissions Committee Review

Produces a list of recruits who are not eligible for decision because of missing documents. The report shows the missing documents, the students' names, and the students' telephone numbers.

Menu Access: Reports Menu: Contact Reports: Not Eligible for Decision

File: \$CARSPATH/modules/admit/reports/not_elig

Student Who Have Not Registered

Prints a list of applicants who have reserved a space in an incoming class but who have not yet registered.

Menu Access: Session Processing: Students Not Registered

File: \$CARSPATH/modules/admit/reports/resnreg

Weekly Statistics by Caller

Produces a call entry call statistics report, sorted by Caller ID.

Menu Access: Call Entry: Reports Menu: Weekly Stats by Caller

File: \$CARSPATH/modules/admit/reports/callstat

Weekly Statistics by Contact

Produces a call entry call statistics report, sorted by contact.

Menu Access: Call Entry: Reports Menu: Weekly Stats by Contact

File: \$CARSPATH/modules/admit/reports/callstat2

Printed Reports in the others Directory

The following lists the Recruiting/Admissions reports stored in \$CARSPATH/modules/admit/others.

Admissions Count Report

Produces a list of admissions candidates based on any of multiple selection criteria.

Menu Access: Reports Menu: Statistical Reports: Counts, Subtotals

File: \$CARSPATH/modules/admit/others/countrpt

Contact Report for Individuals

Produces a list of prospects who have received contacts, based on any of multiple selection criteria.

Menu Access: Reports Menu: Contact Reports: Contacts by Prospect

File: \$CARSPATH/modules/admit/others/ctcrpt

Examination Score Report

Produces a list of student exam scores for a specified exam code, and for a specified group of students: program, session, year, and current status. This report can be ordered by name or exam score.

Menu Access: Reports Menu: Counselor Reports: Examination Scores

File: \$CARSPATH/modules/admit/others/examrpt

(Expected Contacts Report)

Produces a list of all students for whom one or more contact has an Expected status. Has a user-defined title.

Menu Access: Reports Menu: Contact Reports: Expected Contacts

File: \$CARSPATH/modules/admit/others/ctcexpect

Lead Roster By Parameters Report

The Lead Roster By Parameters Report generates lists of leads by various selection criteria. Enter values in the search parameters to select the leads you want to print on a report, and leave the remaining selection parameters blank. Use the final parameter to order the data on the report, (i.e., by city or state). By default, this report also sorts data by lead name after it sorts by the value entered in the Primary Sort Field. The beginning of the report summarizes the parameters passed. If permanent records exist for the lead, the permanent ID number also prints.

Menu Access: Lead Reports menu: Lead Roster By Parameters

File: \$CARSPATH/modules/admit/others/leadrpt

(Not Received a Contact Report)

Produces a list of all students who have not received a specified contact. Has a user-defined title.

Menu Access: Reports Menu: Contact Reports: Not Received One Contact

File: \$CARSPATH/modules/admit/others/ctcnot

(One Contact for All Students Report)

Produces a list of all students for whom a specific contact was created, based on any of multiple selection criteria. Has a user-defined title.

Menu Access: Reports Menu: Contact Reports: One Contact - All Students

File: \$CARSPATH/modules/admit/others/ctcall

(One Contact but Not Another Report)

Produces a list of all students who have one specific contact, but who do not have another specific contact. Has a user-defined title.

Menu Access: Reports Menu: Contact Reports: One Contact - Not Another

File: \$CARSPATH/modules/admit/others/ctccond

Recruit Profile Report

Produces a list of prospects, including address, admissions, school, accomplishments, and contact information, selected on any of multiple criteria.

Menu Access: Reports Menu: Profile Reports: User Defined Profile Info

File: \$CARSPATH/modules/admit/others/profile

Recruit Profile Report

Produces profile information, including address, admissions, school, accomplishments, interests, and contact information, selected by ID.

Menu Access: Reports Menu: Profile Reports: One Student's Profile

File: \$CARSPATH/modules/admit/others/profone

School Directory

Produce a list of all high schools or all colleges based on various selection criteria.

Menu Access: Reports Menu: Profile Reports: School Directory

File: \$CARSPATH/modules/admit/others/schdir

Special Purpose Reports in the others Directory

Several reports stored in the \$CARSPATH/modules/admit/others directory serve special purposes (primarily for conversion from ACT or ETS data). These reports, listed by the menu options they support, are as follows:

ACT Exam Convert/Add

- act
- act2_awk

ACT Exam Data Report

- act1_awk
- act_rep

ACT College Convert/Add

- hscol_awk
- actcol

ACT SS Convert/Add

- actss

- actss2_awk

ACT SS Data Report

- actss1_awk
- actss_rep

ETS College Convert/Add

- hscol_awk
- etscol

ETS SAT Exam Convert/Add

- ets
- ets2_awk

ETS SAT Exam Data Report

- ets1_awk
- ets_rep

ETS SS Convert/Add

- etsss
- etsss2_awk

ETS SS Data Report

- etsss1_awk
- etsss_rep

High School Convert/Add

- hscol_awk
- hstape

ACE Reports for Letter Creation

Introduction

CX contains ACE reports that are used to create letters and labels.

ACE reports used for letter/label creation appear in the \$CARSPATH/modules/admit/reports directory, and their names always begin with the letters *ltr*. These ACE reports select student information from the database, and associate it with macros that can then be plugged into form letters. For example, a student's first name can be plugged into a form letter by using the macro WP_FIRSTNAME. The ACE report selects the student's name from the id_rec.fullname field, and associates it with the WP_FIRSTNAME macro in the format section of the ACE report. Each ACE report has a different purpose, and therefore selects different pieces of student data to associate with different macros. This subsection lists the ACE reports most commonly used in admissions offices, including their general purpose, the macros they create, and the data element associated with each macro.

Note: Examples that appear in association with a macro show the exact format in which the data is extracted. For example, in some cases a year appears as four digits (e.g., 1997), while in other cases, an academic year appears as four digits - hyphen - two digits (e.g., 1997-98). The date formats differ because users want to display years differently depending on their usage within the letters.

The ltradmit ACE Report

The *ltradmit* ACE report is the general purpose ACE report used to create most letters for an admissions office. It creates the following macros:

WP_TODAY

Current date as passed in the Create Admissions Letters menu option (e.g., April 4, 1997).

WP_LABEL

Student's address.

WP_SALUT

Salutation for letters to the student (e.g., Dear Ms. Huggins or Dear Amanda).

WP_FULLNAME

Student's full name (e.g., Amanda Huggins).

WP_FIRSTNAME

Student's first name (e.g., Amanda).

WP_LASTNAME

Student's last name (e.g., Huggins).

WP_LTR_NAME

Location of letter (e.g., /usr/carsi/wp/admissions/FileCabinet/letters/ACPTLTR.ltr).

WP_DUE_DATE

Due date of contact used to create the letter (e.g., 04/04/97).

WP_APPT_TIME

Time entered in Contact record used to create letter (e.g., 9:00 a.m.).

WP_FAC_BLDG

Student's academic advisor's building address (e.g., Curtiss Hall).

WP_FAC_NAME

Name of student's academic advisor (John Mendel).

WP_FAC_ROOM

Student's academic advisor's office number (e.g., 222).

WP_FAC_TITLE

Student's academic advisor's title (e.g., Dr.).

WP_MAJOR_CODE

Student's intended Major code (e.g., BIO).

WP_MAJOR1

Full name of student's major (e.g., Biology).

WP_PLAN_ENR_SESS

Student's intended session of enrollment (e.g., Fall).

WP_PLAN_ENR_YEAR

Student's intended year of enrollment (e.g., 1997).

WP_ACAD_YEAR

Student's intended academic year of enrollment (e.g., 1997-98).

WP_CLASS_YEAR

Student's expected year of graduation from your institution (e.g., 2001).

WP_DECISION

Student's Decision code (e.g., FULL).

WP_DECISION_DATE

Date of student's decision (e.g., 04/04/97).

WP_RANK

Student's rank on a waitlist for admission (e.g., 5).

WP_CUR_STAT

Student's current admissions status (e.g., INQUIRED).

WP_DEPT_LTR

Student's department (e.g., LAS for Liberal Arts and Sciences).

WP_REF_SOURCE

Student's Referral code (e.g., ALUM).

WP_REF_NAME

Full name of the person who referred the student to your institution (e.g., David Skinner).

WP_REF_LABEL

Address label for the reference person.

WP_REF_LASTNAME

Title and last name of the reference person (e.g., Mr. Skinner).

WP_COUNSELOR_ID

ID number of student's admission counselor (e.g., 12345).

WP_COUNSELOR_NAME

Name of student's admission counselor (e.g., John Smith).

The Itrenclose ACE Report

The *Itrenclose* ACE report is used to create letters with enclosures to prospective students. The *Itrenclose* ACE report selects data from the `id_rec`, `ctc_rec`, `adm_rec`, `acad`, and `_cal_rec`.

WP_LTR_NAME

Location of letter (e.g., usr/carsi/wp/admissions/FileCabinet/letters/INTCNF.ltr).

WP_LABEL

Student's address.

WP_TODAY

Current date as passed in the Create Admissions Letters menu option (e.g., April 4, 1997).

WP_SALUT

Salutation for student (e.g., Dear Ms. Huggins or Dear Amanda).

WP_FULLNAME

Student's full name (e.g., Amanda Huggins).

WP_LASTNAME

Student's last name (e.g., Huggins).

WP_FIRSTNAME

Student's first name (e.g., Amanda).

WP_ID

Lead number of the Lead record (e.g., 2345).

WP_CUR_STAT

Student's current admissions status (e.g., INQUIRED).

WP_MAJOR_CODE

Student's intended Major code (e.g., BIO).

WP_MAJOR1

Full name of student's major (e.g., Biology).

WP_PLAN_ENR_YEAR

Student's intended year of enrollment (e.g., 1997).

WP_PLAN_ENR_SESS

Student's intended session of enrollment (e.g., Fall).

WP_SESS

The Session code for the student's intended session of enrollment (e.g., FA).

WP_REF_NAME

Full name of person who referred student to your institution (e.g., David Skinner).

WP_DECISION

Student's Decision code (e.g., FULL).

WP_REF_SOURCE

Student's Referral code (e.g., ALUM).

WP_LTR_NAME

Location of letter (e.g., /usr/carsi/wp/admissions/FileCabinet/letters/SCHVST.ltr).

WP_RANK

Student's rank on a waitlist for admission (e.g., 5).

WP_DEGREE

Degree code of the student (e.g., BA).

WP_DEPT_LTR

Student's department (e.g., LAS for Liberal Arts and Sciences).

WP_CLASS_YEAR

Student's expected year of graduation from your institution (e.g., 2001).

WP_CORR_ID

ID number of correspondent entered for the Lead Contact record (e.g., 12345).

WP_CORR_NAME

Name linked to the correspondent ID of the Lead Contact record (e.g., Jack Sprat).

WP_COUNSELOR_ID

ID number of student's admission counselor (e.g., 12345).

WP_COUNSELOR_NAME

Name of student's admission counselor (e.g., John Smith).

WP_ACAD_YEAR

Student's intended academic year of enrollment (e.g., 1997-98).

WP_APPT_TIME

Time entered for the lead contact record (e.g. 10:00 a.m.).

WP_REF_LABEL

Address label for reference person.

WP_REF_LASTNAME

Title and last name of the reference person (e.g., Mr. Skinner).

WP_ENCLOSURE

The title of the pamphlet or brochure that you wish to include with the letter (e.g., Greek Living Brochure).

The Itrlead ACE Report

The *Itrlead* ACE report is used to create letters that are sent to leads, and selects data from a student's lead_rec and ldctc_rec.

WP_LABEL

Student's address.

WP_LTR_NAME

Location of letter (e.g., /usr/carsi/wp/admissions/FileCabinet/letters/LEADLT.Itr).

WP_MAJOR_CODE

Major code of student's first choice of academic major (e.g., Biology).

WP_MAJOR1

Full name of student's first choice of academic major (e.g., Biology).

WP_MAJOR2

Full name of student's second choice of academic major (e.g., Chemistry).

WP_REF_SOURCE

Student's Referral code (e.g., ETSS).

WP_FIRSTNAME

Student's first name (e.g., Amanda).

WP_FULLNAME

Student's full name (e.g., Amanda Huggins).

WP_LASTNAME

Student's last name (e.g., Huggins).

WP_SEX

Student's gender (e.g., M or F).

WP_TODAY

Current date as passed in the Create Admissions Letters menu option (e.g., April 4, 1997).

WP_PLAN_ENR_SESS

Planned session of enrollment for the student (e.g., Fall).

WP_PLAN_ENR_YEAR

Planned year of enrollment for the student (e.g., 1999).

WP_ETHNIC

Ethnic code of the student (e.g., WH).

WP_PROG

Program code of the student (e.g., UNDG).

WP_ID

Lead number of the lead record (e.g., 2345).

WP_ADDRLINE1

First line of student's address (e.g., 23 County Road 115).

WP_ADDRLINE2

Second line of student's address.

WP_ADDRLINE3

Third line of student's address.

WP_CITY

The city of the student's residence (e.g., Santa Fe).

WP_STATE

The state of the student's residence (e.g., NM).

WP_ZIP

The zip code of the student's residence (e.g., 87501).

WP_COUNTRY_TEXT

The country of the student's residence (e.g., USA).

WP_EMAIL

The student's e-mail address.

WP_SALUT

The salutation for the student (e.g., Ms. Huggins or Amanda). If a title has been entered in a student's lead record, a formal salutation will be created (i.e., Ms. Huggins). If a title has not been entered in a student's lead record, an informal salutation will be created (i.e., Amanda). If you always want to use an informal salutation for all students, use the macro WP_FIRSTNAME, (i.e., Dear WP_FIRSTNAME; instead of Dear WP_SALUT;).

WP_SCHOOL_NAME

Name of school the student attends (e.g., Ridgmont High School).

WP_CORR_ID

ID number of correspondent entered for the lead contact record (e.g., 12345).

WP_CORR_NAME

Name linked to the correspondent ID of the lead contact record (e.g., Jack Sprat).

WP_APPT_TIME

Time entered for the lead contact record (e.g. 10:00 a.m.).

WP_DUE_DATE

Due date of the lead contact record (e.g., January 1, 1998).

The Itrintconf ACE Report

The *Itrintconf* ACE report is used to create interview confirmation letters. It also creates an Interview record (intvwrecom_rec). The *Itrintconf* ACE report selects data from the ctc_rec, id_rec, adm_rec, and fac_rec.

WP_TODAY

Current date as passed in the Create Admissions Letters menu option (e.g., April 4, 1997).

WP_LABEL

Student's address.

WP_SALUT

Salutation for student (e.g., Dear Ms. Huggins or Dear Amanda).

WP_FULLNAME

Student's full name (e.g., Amanda Huggins).

WP_FIRSTNAME

Student's first name (e.g., Amanda).

WP_LASTNAME

Student's last name (e.g., Huggins).

WP_LTR_NAME

Location of letter (e.g., /usr/carsi/wp/admissions/FileCabinet/letters/INTCNF.ltr).

WP_INTV_NAME

Name of interviewer (e.g., Lisa Robertson).

WP_FAC_TITLE

Title of interviewer (e.g., Dr.).

WP_INTV_DATE

Date of interview/contact's completion date (e.g., May 5, 1997).

WP_INTV_TIME

Time of interview/contact's time (e.g., 9:00 a.m.).

WP_INTV_ROOM

Interviewer's office address if faculty (e.g., 222 Curtiss Hall).

WP_PROG

Student's program (e.g., UNDG).

WP_MAJOR1

Full name of student's major (e.g., Biology).

WP_MAJOR_CODE

Student's intended Major code (e.g., BIO).

WP_PLAN_ENR_SESS

Student's intended session of enrollment (e.g., Fall).

WP_PLAN_ENR_YEAR

Student's intended year of enrollment (e.g., 1997).

WP_COUNSELOR_ID

ID number of student's admission counselor (e.g., 12345).

WP_COUNSELOR_NAME

Name of student's admission counselor (e.g., John Smith).

The Itrmistran ACE Report

The *Itrmistran* ACE report is used to create letters telling incoming students what missing documents they must submit before they can start classes (e.g., a full/final transcript). The *Itrmistran* ACE report selects data from the *ctc_rec*, *id_rec*, *adm_rec*, *ed_rec*, *exam_rec*, and *intvwrecom_rec*.

WP_TODAY

Current date as passed in the Create Admissions Letters menu option (e.g., April 4, 1997).

WP_LABEL

Student's address.

WP_SALUT

Salutation for student (e.g., Dear Ms. Huggins or Dear Amanda).

WP_FULLNAME

Student's full name (e.g., Amanda Huggins).

WP_FIRSTNAME

Student's first name (e.g., Amanda).

WP_LASTNAME

Student's last name (e.g., Huggins).

WP_LTR_NAME

Location of letter (e.g., /usr/carsi/wp/admissions/FileCabinet/letters/MISTRN.ltr).

WP_EXAMS

Names of missing test scores (e.g., Scores from Scholastic Aptitude Test).

WP_RECS

Names of missing recommendations (e.g., Recommendation from Bob Smith).

WP_TRANS

Missing and partial transcript names (e.g., Transcript from University of Cincinnati).

WP_OTHER_DOCS

Names of other missing documents (e.g., Application Essay).

WP_MAJOR1

Full name of student's major (e.g., Biology).

WP_MAJOR_CODE

Student's intended Major code (e.g., BIO).

WP_PLAN_ENR_SESS

Student's intended session of enrollment (e.g., Fall).

WP_PLAN_ENR_YEAR

Student's intended year of enrollment (e.g., 1997).

WP_CUR_STAT

Student's current admissions status (e.g., APPLIED).

WP_COUNSELOR_ID

ID number of student's admission counselor (e.g., 12345).

WP_COUNSELOR_NAME

Name of student's admission counselor (e.g., John Smith).

The *ltrparent* ACE Report

The *ltrparent* report is used to create letters to parents of students at your institution. The *ltrparent* report selects data from the relation_rec, ctc_rec, adr_rec, id_rec, adm_rec, prog_enr_rec, and fac_rec.

WP_DUE_DATE

Due date of contact used to create the letter (e.g., 04/04/97).

WP_COUNSELOR_ID

ID number of student's admission counselor (e.g., 12345).

WP_COUNSELOR_NAME
Name of student's admission counselor (e.g., John Smith).

WP_CLASS_YEAR
Student's expected year of graduation from your institution (e.g., 2001).

WP_FAC_NAME
Name of student's academic advisor (e.g., John Mendel).

WP_FAC_BLDG
Student's academic advisor's building address (e.g., Curtiss Hall).

WP_FAC_ROOM
Student's academic advisor's office number (e.g., 222).

WP_FAC_TITLE
Student's academic advisor's title (e.g., Dr.).

WP_MAJOR1
Full name of student's major (e.g., Biology).

WP_PLAN_ENR_SESS
Student's intended session of enrollment (e.g., Fall).

WP_PLAN_ENR_YEAR
Student's intended year of enrollment (e.g., 1997).

WP_CUR_STAT
Student's current admissions status (e.g., INQUIRED).

WP_MAJOR_CODE
Student's intended Major code (e.g., BIO).

WP_PROG
Program code of the student (e.g., UNDG).

WP_DEGREE
Degree code of the student (e.g., BA).

WP_SEX
Student's gender (e.g., M or F).

WP_ETHNIC
Ethnic code of the student (e.g., WH).

WP_DECISION
Student's Decision code (e.g., FULL).

WP_REF_SOURCE
Student's Referral code (e.g., ALUM).

WP_ID
Lead number of the lead record (e.g., 2345).

WP_PARENT_ID
Student's parent's ID number (e.g., 5039590).

WP_APPT_TIME

Time entered in Contact record used to create letter (e.g., 9:00 a.m.).

The Itrpreapp ACE Report

The *Itrpreapp* ACE report is used to create letters telling inquiries what documents have arrived into the admissions office in advance of their application for admission. The *Itrpreapp* ACE report selects data from the `ctc_rec`, `id_rec`, `adm_rec`, `ed_rec`, `exam_rec`, and `intvwrecom_rec`.

WP_TODAY

Current date as passed in the Create Admissions Letters menu option (e.g., April 4, 1997).

WP_LABEL

Student's address.

WP_SALUT

Salutation for student (e.g., Dear Ms. Huggins or Dear Amanda).

WP_FULLNAME

Student's full name (e.g., Amanda Huggins).

WP_FIRSTNAME

Student's first name (e.g., Amanda).

WP_LASTNAME

Student's last name (e.g., Huggins).

WP_LTR_NAME

Location of letter (e.g., `/usr/carsi/wp/admissions/FileCabinet/letters/PREAPP.ltr`).

WP_EXAMS

Names of test scores received (e.g., Scores from Scholastic Aptitude Test).

WP_RECS

Names of recommendations received (e.g., Recommendation from Bob Smith).

WP_TRANS

Names of transcripts received (e.g., Transcript from University of Cincinnati).

WP_OTHER_DOCS

Names of other documents received (e.g., Application Essay).

WP_MAJOR1

Full name of student's major (e.g., Biology).

WP_MAJOR_CODE

Student's intended Major code (e.g., BIO).

WP_PLAN_ENR_SESS

Student's intended session of enrollment (e.g., Fall).

WP_PLAN_ENR_YEAR

Student's intended year of enrollment (e.g., 1997).

WP_CUR_STAT

Student's current admissions status (e.g., APPLIED).

WP_COUNSELOR_ID

ID number of student's admission counselor (e.g., 12345).

WP_COUNSELOR_NAME

Name of student's admission counselor (e.g., John Smith).

The Itrschlabl ACE Report

The *Itrschlabl* ACE report is used to create letters and labels for mailings to high schools and other colleges. The *Itrschlabl* ACE report selects data from the `ctc_rec`, `id_rec`, and `sch_rec`.

WP_TODAY

Current date as passed in the Create Admissions Letters menu option (e.g., April 4, 1997).

WP_LABEL

School's address.

WP_SALUT

Salutation for school's correspondent (e.g., Dear Dr. Skinner).

WP_FULLNAME

School's name (e.g., University of Cincinnati).

WP_FIRSTNAME

First name of school's correspondent (e.g., Amanda).

WP_LASTNAME

Last name of School's correspondent (e.g., Skinner).

WP_LTR_NAME

Location of letter (e.g., `/usr/carsi/wp/admissions/FileCabinet/letters/SCHCAT.ltr`).

The Itrincapp ACE Report

The *Itrincapp* ACE report is used to create letters telling applicants what documents are missing from their application files. The *Itrincapp* ACE report selects data from the `ctc_rec`, `id_rec`, `adm_rec`, `ed_rec`, `exam_rec`, and `intvwrecom_rec`.

WP_TODAY

Current date as passed in the Create Admissions Letters menu option (e.g., April 4, 1997).

WP_LABEL

Student's address.

WP_SALUT

Salutation for student (e.g., Dear Ms. Huggins or Dear Amanda).

WP_FULLNAME

Student's full name (e.g., Amanda Huggins).

WP_FIRSTNAME

Student's first name (e.g., Amanda).

WP_LASTNAME

Student's last name (e.g., Huggins).

WP_LTR_NAME

Location of letter (e.g., `/usr/carsi/wp/admissions/FileCabinet/letters/INCAPP.ltr`).

WP_EXAMS

Names of missing test scores (e.g., Scores from Scholastic Aptitude Test).

WP_RECS

Names of missing recommendations (e.g., Recommendation from Bob Smith).

WP_TRANS

Names of missing transcripts (e.g., Transcript from University of Cincinnati).

WP_OTHER_DOCS

Names of other missing documents (e.g., Application Essay).

WP_MAJOR1

Full name of student's major (e.g., Biology).

WP_MAJOR_CODE

Student's intended Major code (e.g., BIO).

WP_PLAN_ENR_SESS

Student's intended session of enrollment (e.g., Fall).

WP_PLAN_ENR_YEAR

Student's intended year of enrollment (e.g., 1997).

WP_CUR_STAT

Student's current admissions status (e.g., APPLIED).

WP_COUNSELOR_ID

ID number of student's admission counselor (e.g., 12345).

WP_COUNSELOR_NAME

Name of student's admission counselor (e.g., John Smith).

The Itrschvst ACE Report

The *Itrschvst* ACE report is used to create letters telling students the date and time of an upcoming visit to their school by an admissions representative. The *Itrschvst* ACE report selects data from the *ctc_rec*, *id_rec*, *adm_rec*, *fac_rec*, and *schd_rec*.

WP_TODAY

Current date as passed in the Create Admissions Letters menu option (e.g., April 4, 1997).

WP_LABEL

Student's address.

WP_SALUT

Salutation for student (e.g., Dear Ms. Huggins or Dear Amanda).

WP_FULLNAME

Student's full name (e.g., Amanda Huggins).

WP_FIRSTNAME

Student's first name (e.g., Amanda).

WP_LASTNAME

Student's last name (e.g., Huggins).

WP_LTR_NAME

Location of letter (e.g., /usr/carsi/wp/admissions/FileCabinet/letters/SCHVST.ltr).

WP_SCHOOL_NAME

Name of school for upcoming visit.

WP_SCHEDULE_ID

ID number of person visiting the school.

WP_PLACE_ID

ID number of school for upcoming visit.

WP_SCHEDULE_DATE

Date of visit to school.

WP_SCHEDULE_TIME

Time of visit to school.

WP_PLACE_CODE

Location type of the upcoming visit (e.g., HS).

WP_SCHEDULE_STAT

Scheduling status (e.g., C for confirmed or T for tentative).

WP_DUE_DATE

Due date of contact used to create the letter (e.g., 04/04/97).

WP_APPT_TIME

Time entered in Contact record used to create letter (e.g., 9:00 a.m.).

WP_FAC_BLDG

Student's academic advisor's building address (e.g., Curtiss Hall).

WP_FAC_NAME

Name of student's academic advisor (e.g., John Mendel).

WP_FAC_ROOM

Student's academic advisor's office number (e.g., 222).

WP_FAC_TITLE

Student's academic advisor's title (e.g., Dr.).

WP_MAJOR1

Full name of student's major (e.g., Biology).

WP_PLAN_ENR_SESS

Student's intended session of enrollment (e.g., Fall).

WP_PLAN_ENR_YEAR

Student's intended year of enrollment (e.g., 1997).

WP_CLASS_YEAR

Student's expected year of graduation from your institution (e.g., 2001).

WP_DECISION

Student's Decision code (e.g., FULL).

WP_REF_SOURCE

Student's Referral code (e.g., ALUM).

WP_REF_NAME

Full name of person who referred student to your institution (e.g., David Skinner).

WP_REF_LABEL

Address label for reference person.

WP_REF_LASTNAME

Title and last name of reference person (e.g., Mr. Skinner).

WP_COUNSELOR_ID

ID number of person visiting the school (e.g., 12345).

WP_COUNSELOR_NAME

Name of student's admission counselor (e.g., John Smith).

SECTION 13 - CUSTOMIZING THE RECRUITING/ADMISSIONS PROCESSES

Overview

Introduction

This section provides procedures for setting and installing the features of the Recruiting/Admissions product. It includes the following procedures:

- Assessing institution needs for the module
- Deciding on admission statuses and status sequences
- Determining Contact codes for all statuses, letters, and miscellaneous documents
- Completing common and Recruiting/Admissions tables
- Reviewing and modifying Recruiting/Admissions macros
- Setting up the *admprocess* script to run each night using *cron*
- Creating ID records and School records for high schools and colleges, usually through purchasing data files from ETS (Educational Testing Service)
- Determining which admissions office personnel to include in the *admtemp* permissions group
- Setting up additional programs, if necessary

Other Implementation Tasks

In addition, to the above tasks, you must also complete tasks for which documentation exists in other Jenzabar resources. The following list contains additional tasks you must complete, and the Jenzabar resource to use:

Modify the Inquiry screen to match your inquiry card

The curriculum materials from the Jenzabar course *Database Tools and Utilities*

Modify the Application screen to match your institution's application

The curriculum materials from the Jenzabar course *Database Tools and Utilities*

Create counselor IDs and assign them to territories

The following resources:

- The curriculum materials from the Jenzabar course *System Administration* (creating SQL statements)
- The Jenzabar manual *Getting Started User Guide*
- The Jenzabar manuals *Implementation and Maintenance / CX System Reference Technical Manuals*

Compose letters

The following resources:

- The Jenzabar manual *Communications Management User Guide*
- The Jenzabar manual *Communications Management Technical Manual*

Note: For more information, see the *Setting Up Recruiting/Admissions Letters* section in this manual about creating letters specifically for Recruiting/Admissions.

Basic Information

This section contains detailed procedures specific to the Recruiting/Admissions product. For information on performing basic procedures, including using the MAKE processor and reinstalling options, refer to the following resources:

- *Database Tools and Utilities* course notebook
- *CX System Reference Technical Manual*

Assessing the Recruiting/Admissions Setup

Introduction

CX provides several ways to implement the options of the Recruiting/Admissions product. For example, you can decide to automate your letter scheduling by using Tickler if desired, or perform this function manually. You can also determine if admissions office employees need to view Financial Aid, Student Services, or Registration information. In addition, you can elect to use Schedule Entry, Call Entry, and Electronic Application processing (this last option is available as a separate product). After assessing the needs of your institution, you can change the default settings, enable macros, and re-install the product to save your changes.

Before You Begin

The macro files that control many of the features in Recruiting/Admissions are maintained under the CX Revision Control System (RCS). You must therefore use MAKE processing commands to check out macro files and to check them back in before performing reinstalls that activate your changes. For more information about the MAKE processor, see *Jenzabar CX Technical Manual*.

Macro Files

To help you make policy decisions controlled by CX macros, review the comments in the following macro files and set the values according to your needs. You can also refer to the *Recruiting/Admissions Macros And Includes* section of this manual for additional explanations of the macros that control Recruiting/Admissions processing.

Note: The macro files listed below appear in order of importance to the Recruiting/Admissions product.

- \$CARSPATH/macros/custom/admissions
- \$CARSPATH/macros/custom/periodic
- \$CARSPATH/macros/custom/common
- \$CARSPATH/macros/custom/web
- \$CARSPATH/macros/custom/tables
- \$CARSPATH/macros/custom/student
- \$CARSPATH/macros/custom/matric

Reinstalls After Macro Changes

After you change macro settings and check in the file, you must reinstall each changed macro file using a MAKE processor command in the following format:

```
make reinstall F=<filename>
```

Example: Enter the following command at the UNIX prompt to reinstall the *admissions* macro file:

```
make reinstall F=admissions
```

Note: Alternatively, you can check in and reinstall the files with the following command:

```
make cii F=admissions
```

You must also use the following commands to perform the necessary reinstalls after changing and reinstalling your macro files. These commands reinstall all the aspects of Recruiting/Admissions customized with macros.

```
cd $CARSPATH/include/applic
```

```
make reinstall F=admit
```

```
cd src/admit  
make reinstall F=ALL
```

```
cd $CARSPATH/menusrc/admit  
make reinstall F=ALL
```

```
cd $CARSPATH/menuopt/admit  
make reinstall F=ALL
```

```
cd $CARSPATH/modules/admit  
make reinstall F=ALL
```

Reviewing and Modifying Data in Tables and Records

Introduction

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

Procedure

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

1. For each Recruiting/Admissions table, review the codes supplied with CX's distributed database. Determine whether or not the codes meet the needs of your institution. Make updates as appropriate. For some table values, you must consider how you have defined certain macro values. For example, if you define the macro INQUIRY_DEFAULT_CTC with the value INQUIRED, then the contact INQUIRED must exist in the Contact table (ctc_table). Similarly, if you define the macro ADM_CL_DEF (default admission classification) with the value FF (for first time freshman), then the code FF must exist in the Classification table (cl_table).
2. Review the institution's records converted from the previous Recruiting/Admissions system. Determine whether or not the records need to be updated to meet the needs of the CX users and reports. Make updates as appropriate.

Table and Record Information

For more information about the tables and records, see the section *Recruiting/Admissions Tables and Records* in this manual.

Order of Table Information in This Section

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

- Tables appear in the order of implementation recommended by Jenzabar. Typically, the order of implementation is determined by dependencies between tables (e.g., if a code in Table A must exist in Table B, then you must set up Table B before Table A).
- Tables that you can complete in any sequence appear at the end of the list, in alphabetical order.

Table to Set Up in Sequential Order

The following lists the sequence in which you should set up the tables. Information about defining key values for these tables follows.

- Note:** For reference information about the tables listed below, see the *Recruiting/Admissions Tables and Records* section in this manual.
1. Academic Status table (acad_stat_table)
Note: Typically, this table is the responsibility of the registrar's office, and is set up when the CX Registration/Student Records products are implemented.
 2. Decision table (dec_table)
 3. Reference table (ref_table)
 4. Enrollment Status table (enr_stat_table)
 5. Enrollment Sequence table (enr_seq_table)
 6. Country table (ctry_table)

7. Lead Country table (leadctry_table)
8. Denomination table (denom_table)
9. Lead Denomination table (leaddenom_table)
10. Ethnic table (ethnic_table)
11. Lead Ethnic table (leadethnic_table)
12. Major table (major_table)

Note: Typically, this table is the responsibility of the registrar's office, and is set up when the CX Registration/Student Records products are implemented.

13. Lead Major table (leadmaj_table)
14. Tickler table (tick_table)
15. Program table (prog_table)
16. Session table (sess_table)
17. Academic Calendar record (acad_cal_rec)

Note: Typically, this table is the responsibility of the registrar's office, and is set up when the CX Registration/Student Records products are implemented.

18. Communication table (comm_table)
19. ADR Runcode table (adr_table)
20. Alternate Address table (aa_table)
21. Relationship table (rel_table)
22. Addressing record (adr_rec)
23. Contact table (ctc_table)
24. County table (cty_table)
25. State table (st_table)
26. Zip Code table (zip_table)
27. Hold table (hold_table)
28. Hold Action table (hold_act_table)
29. Entry Selection/Sort table (entsel_table)
30. Entry Selection/Sort Criteria table (entselcrit_table)
31. Office table (ofc_table)
32. Office Permissions table (ofcperm_table)
33. Division table (div_table)
34. Department table (dept_table)
35. Privacy Act table (priv_table)
36. Privacy Act Field table (privfld_table)

Tables to Set Up in Any Order

You can implement the following tables in any order after you set up the tables listed above.

1. Accomplishment table (accomp_table)

2. Citizen table (citz_table)
3. Degree table (deg_table)
4. Exam table (exam_table)
5. Handicap table (hand_table)
6. Interest table (int_table)
7. Involvement table (invl_table)
8. Marital table (mrtl_table)
9. Suffix table (suffix_table)
10. Title table (title_table)
11. Visa table (visa_table)

Building the Recruiting/Admissions Tables

Introduction

The processes in the Recruiting/Admissions product use the tables described here to control and validate data entry. You must build these tables before performing any processing and before building any other Recruiting/Admissions tables.

Access

Access instructions assume you are beginning at the Recruiting/Admissions: Main Menu, Table Maintenance option.

Academic Status Table

The registrar's office typically controls the Academic Status table (`acad_stat_table`), which defines valid types of statuses a student might have (e.g., accepted conditionally, admitted and enrolled, graduated, or probation). You must, however, use codes established in the Academic Status table to complete the Decision table. You can access this table from the Systems Management: Table Maintenance menu or from the Registration: Table Maintenance: Registrar menu.

The important field in the table for Recruiting/Admissions is:

Admissions Update

A Y/N flag indicating whether the admissions office can update the records of a student with the specified status. This field reflects policy decisions about the relationship between admissions and registration at an institution.

Decision Table

The Decision table (`dec_table`) defines valid types of admittance decisions that the institution associates with applicants. Admissions decisions are not simply acceptances or rejections; CX can process a variety of other decision types.

Example: You might have a Decision code that denies an applicant immediate acceptance, but encourages the applicant to reapply after completing remedial work at a community college. You might also have a code that denies an applicant's acceptance with no recourse.

You can access this table from the Systems Management: Table Maintenance menu or from the Recruiting/Admissions: Table Maintenance menu. To complete the Decision table, use codes established in the Academic Status table.

Note:

- The Decision table should contain only one entry for missing documents, not an entry for every possible combination.
- By using WP_IF statements within your acceptance, denial, and waitlist letters, you can customize letters to print different words, sentences, or paragraphs depending on the applicant's Decision code. Consider letter writing requirements when establishing entries in the Decision table.

Example:

```
WP_IF(WP_DECISION,FULL){You have been fully accepted...}  
or  
WP_IF(WP_DECISION,PROB){You have been accepted on a  
probationary basis...}
```

Important fields in the table are:

Academic Status Code

The status the applicant will have with the registrar's office, should he/she become an enrolled student (e.g., for FULL acceptance decision, the academic status might be ACPT; for PROB acceptance decision, the academic status might be ACPR). This code becomes the value for the prog_enr_rec.acst field, which in turn regulates student registration processing. This field must be completed only for those Decision codes indicating the applicant has been accepted, since only accepted applicants will have a Program Enrollment record created for them.

Code

The decision made concerning acceptance to the institution. Standard values are:

- CODC (Conditional pending documents)
- COND (Conditional acceptance)
- DENY (Denied admission)
- FULL (Full acceptance)
- PROB (Probationary acceptance)
- SPEC (Special student admission)

Description

The description of the decision made concerning acceptance to this institution (e.g., Full Acceptance, for the code FULL).

Reference Table

The Reference table defines valid types of referrals for a prospect. The *admentry* program uses the table to retain the initial source of referral for the prospect or applicant. The Reference code can be tied to a reference ID (adm_rec.ref_id) to link the reference to a specific entity, and also enables you to create variable letters in the Letter Processing System (*lps*).

Example: The Reference code and reference ID are linked to the WP_REF_SOURCE and WP_REF_NAME macros respectively; you can use them with IF-THEN-ELSE statements to insert variable text in letters to prospects.

You can access this table from the Systems Management: Table Maintenance menu or from the Recruiting/Admissions: Table Maintenance menu.

Important fields in the table are:

Code

A code identifying how a prospective applicant first learned about your institution or became known to the institution (e.g., CNSL for counselor at a high school, ACTS for ACT student search data file, or UN for unknown referral).

Description

A description of how a person first learned about the institution (e.g., counselor - high school or ACT search data file).

Type

The type of referral. This field groups codes into more general categories for reporting purposes (e.g., PR for personal referrals, AD for advertisements, RT for recruitment trips, or SS for student search data files). You can use the same type for multiple codes.

Enrollment Status Table

The Enrollment Status table defines the valid statuses an applicant can have, from Inquired to Enrolled. You can access this table from the Systems Management: Table Maintenance menu or from the Recruiting/Admissions: Table Maintenance menu.

You should define a Contact code in the Contact table that matches each of the statuses you define in the Enrollment Status table.

The *admstats* program uses the Enrollment Status table in conjunction with the Enrollment Sequence table to define valid status names for the admission process. Minimize the number of valid Status codes to ease daily maintenance and reporting requirements. Jenzabar recommends that you avoid using the student's enrollment status to track information that can be (or already is) tracked by other fields in the database.

You may also want to establish Status codes that ease the creation of Program Enrollment records at the beginning of each academic session. For example, you might use CONFIRM as the status for students who have confirmed and paid deposits, and CONFNOFE for students who have verbally confirmed but not submitted a deposit. Using wildcards, you can then create Program Enrollment records for all students with a status of CONF*. For more information about creating Program Enrollment records, see *Completing Session Processing* in this manual.

Important fields in the table are:

Code

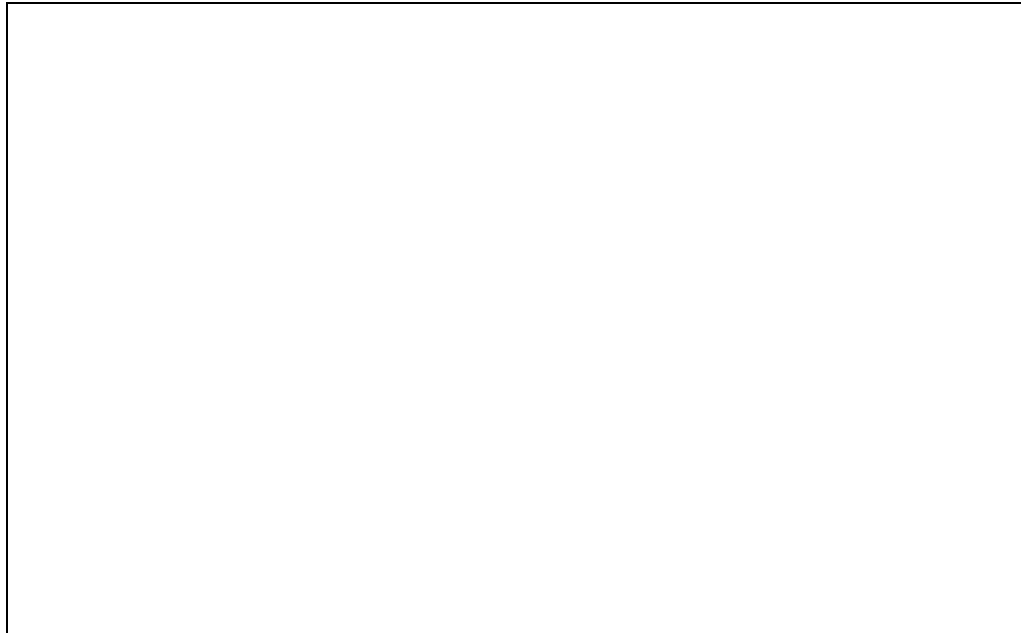
The enrollment status or stage of enrollment (e.g., ACCEPTED, for accepted for admission). There must be one blank entry.

Description

A description of the enrollment status or stage of enrollment (e.g., Accepted for admission).

Priority

The priority for ranking this enrollment status (e.g., 50). The highest priority is zero (0). Assign your blank status table entry a priority of 0, then set other statuses based on the logical progression through the admissions flow (e.g., APPLIED should have a higher priority (lower number) than ACCEPTED, since students must apply before they can be accepted). The following chart can help you establish priorities and other enrollment status information.



Based on the Status codes in this example, you could establish the following priorities:

Code	Status	Priority
	Blank	1
INQUIRED	Self-inquired recruit	5
APPLIED	Applied - fee paid	20

Code	Status	Priority
APPCOMP	Application complete	25
WITHDREW	Withdrew application	30
WAITLIST	Placed on wait list	35
ACCEPTED	Accepted for admission	40
DENIED	Denied admission	45
DEFERRED	Deferred acceptance	50
CONFIRM	Confirmed - deposit paid	60
DECLINED	Declined acceptance	65
NOSHOW	Did not show for classes	70
ENROLLED	Enrolled in classes	75

Temporary Workers Update

A Y/N flag indicating whether temporary workers may update information while an applicant has this status. Enter Y for yes or N for no.

Note: You define temporary workers within your office by adding them to the *admtemp* permissions group. To add a temporary worker to this group, use the following command at the UNIX prompt:

addtogrp admtemp <login name>

To remove a temporary worker from this group, use the following command at the UNIX prompt:

rmfromgrp admtemp <login name>

Enrollment Sequence Table

The Enrollment Sequence table (*enr_seq_table*) defines the order of all the admissions steps from inquired to enrolled. The *admstats* program evaluates the Contact table and validates the change it has been directed to make against the entries in the Enrollment Sequence table. The Enrollment Sequence table must contain all possible combinations for status changes from the Enrollment Status table. You can access this table from the Systems Management: Table Maintenance menu or from the Recruiting/Admissions: Table Maintenance menu.

Important fields in the table are:

First/Current Status Code

The first or current enrollment status from which one may progress to another enrollment status (e.g., APPLIED).

Note: A blank entry is required to begin all status sequences (e.g., a blank status to INQUIRED).

Next Status Code

A valid enrollment status to which one may progress (e.g., APPLIED) (use with Update Status of Y), or an invalid enrollment status to which one may *not* progress (use with Update Status of N).

Update Status

A Y/N flag indicating whether this enrollment status sequence is a valid and allowable status change. Set the value to N to disallow the status change from First/Current Status code to Next Status code.

Example: Changing a student's status from INQUIRED to APPLIED is a valid status change; the table entry for this sequence should have an Update Status of Y. In contrast, changing a student's status from INQUIRED to ACCEPTED is not a valid status change since students must first apply before they can be accepted.

When populating the Enrollment Sequence table, you must compare every status listed in the Enrollment Status table with every other status in the Enrollment Status table. Therefore, if the admissions office uses ten possible statuses, and ten entries have been made in the Enrollment Status table, then the Enrollment Sequence table will require 100 entries. Before you begin to populate the Enrollment Sequence table, create a worksheet table similar to the following example to determine the statuses to include.

	blank	INQUIRED	APPLIED	APPCOMP	WITHDREW	ACCEPTED	DENIED	WAITLIST
blank	N	Y	Y	N	N	N	N	N
INQUIRED	N	N	Y	N	N	N	N	N
APPLIED	N	N	N	Y	Y	N	N	N
APPCOMP	N	N	N	N	Y	Y	Y	Y
WITHDREW	N	N	N	N	N	N	N	N
ACCEPTED	N	N	N	N	N	N	N	N
DENIED	N	N	N	N	N	N	N	N
WAITLIST	N	N	N	N	Y	Y	Y	N
CONFIRM	N	N	N	N	N	N	N	N
DECLINED	N	N	N	N	N	N	N	N
DEFERRED	N	N	N	N	N	N	N	N
ENROLLED	N	N	N	N	N	N	N	N
NOSHOW	N	N	N	N	N	N	N	N
REAPPLY	N	N	N	N	N	N	N	N

Note: Once the entries made in this table are used by *admstats*, they must never be deleted. If the decision is made to discontinue a status, populate the Inactive Date field with the current date for the status in question. You should also set the Inactive Date fields in the Enrollment Status table and the Contact table.

Country Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Lead Country Table

The Lead Country table (*leadctry_table*) relates Country codes used at your institution to the countries on ETS data files. ETS annual code revisions can affect the entries in this table. By using this table, you do not need to use the codes used by ACT or ETS; the *tpconvert* script will convert the country names on ACT or ETS data files to the codes you use, based on your table entries. You can access this table from the Systems Management: Table Maintenance menu or from the Recruiting/Admissions.: Table Maintenance menu.

Important fields in the table are:

ETS Country Code

The ETS country name or abbreviation as provided by ETS on the student search and SAT (Scholastic Aptitude Test) exam data files (e.g., AUSTRIA).

Note: If the country name is 29 characters or less, ETS prints the full name of the country in uppercase letters on the ETS data files. Before you load a new ETS data files, review the Country field on the data files (101-129) for each person on the data file. Review the countries listed and make sure they are also defined in the Lead Country table before you actually load the data file.

School's Country Code

The institution's Country code that corresponds to ETS's country name (e.g., AUST for Austria).

Denomination Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Lead Denomination Table

The Lead Denomination table (leaddenom_table) relates religious Denomination codes used at your institution to the codes on ACT and ETS data files. Entries in this table can be affected by ACT/ETS annual code revisions. By using this table, you do not need to use the codes used by ACT or ETS; the *tpconvert* script will convert the codes on ACT or ETS data files to the codes you want to use, based on your table entries. You can access this table from the Systems Management: Table Maintenance menu or from the Recruiting/Admissions: Table Maintenance menu.

Important fields in the table are:

ACT Denomination Code

The code for the denomination that ACT provides on the student search or enhanced record data file (e.g., B for Baptist).

ETS Denomination Code

The code for the denomination that ETS provides on the student search or enhanced record data file (e.g., 9 for Baptist).

School's Denomination Code

The institution's Denomination code that corresponds to the ACT or ETS Denomination code (e.g., BAPT for Baptist).

Ethnic Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Lead Ethnic Table

The Lead Ethnic table (leadethnic_table) relates Ethnic codes used at your institution to the codes on ACT and ETS data files. Entries in this table can be affected by ACT/ETS annual code revisions. By using this table, you do not need to use the codes used by ACT or ETS; the *tpconvert* script will convert the codes on ACT or ETS data files to the codes you want to use, based on your table entries. You can access this table from the Systems Management: Table Maintenance menu or from the Recruiting/Admissions: Table Maintenance menu.

Important fields in the table are:

ACT Ethnic Code

The code for the ethnic classification that ACT provides on the student search or exam data file (e.g., 2 for American Indian/Alaskan Native).

ETS Ethnic Code

The code for the ethnic classification that ETS provides on the student search or exam data file (e.g., A for American/Alaskan Native).

School's Ethnic Code

The Ethnic code of this institution that corresponds to the ACT or ETS Ethnic code (e.g., AA for American Indian/Alaskan Native).

Major Table

The Major table (major_table) defines the valid majors for the institution. For Recruiting/Admissions, its main purpose is to validate anticipated majors for new students, and to help in the conversion of student search data file and exam data file data.

Lead Major Table

The Lead Major table (leadmaj_table) relates Major codes used at your institution to the codes on ACT and ETS data files. Entries in this table can be affected by ACT/ETS annual code revisions. Typically, the registrar's office controls entries in this table. By using this table, you do not need to use the codes used by ACT or ETS; the *tpconvert* script will convert the codes on ACT or ETS data files to the codes you want to use, based on your table entries. You can access this table from the Systems Management: Table Maintenance menu or from the Recruiting/Admissions: Table Maintenance menu.

Important fields in the table are:

ACT/ETS Major Description

A description of the Major code supplied on the ACT and ETS student search or enhanced record data files (e.g., English for an English major).

ACT Major Code

The code for the major that ACT provides on the student search or enhanced record data file (e.g., 804 for an English major).

ETS Major Code

The code for the major that ETS provides on the student search or enhanced record data file (e.g., 628 for an English major).

School's Major Code

The code for the major that ETS provides on the student search or enhanced record data file (e.g., ENGL for an English major).

Tickler Table

For information about setting up this table, see *CX System Reference Technical Manual* and *Communications Management Technical Manual*.

Program Table

The Program table defines the valid programs (e.g., Undergraduate or Graduate) at your institution. A table primarily used in the CX Registration product, the Program table is available from the Systems Management: Table Maintenance menu or from the Student: Registration: Table Maintenance menu.

Important fields in the table that relate to Recruiting/Admissions are:

Admissions Tickler

The code associated with the admissions Tickler code for the program (prog_table.adm_tick). Typically, for the Undergraduate program, this code is ADM; for the Graduate program, it is ADMG.

Display on Web

A Y/N flag indicating whether the program is available for use or table lookup on the World Wide Web application.

Session Table

The Session table defines the valid academic sessions and their order in the academic and calendar years. You can access this table from the Systems Management: Table Maintenance menu or from the Registration: Table Maintenance menu.

For Recruiting/Admissions purposes, relevant fields in the table are:

Session

The code associated with the session (e.g., FA or SP).

Display on Web

A Y/N flag indicating whether the session is available for use or table lookup on the World Wide Web application.

Academic Calendar Record

The Academic Calendar record (*acad_cal_rec*) defines important variables used in the registration process, including academic programs, sessions, years, and date ranges. Controlled and managed by the registrar's office, this record also provides information to the *admstats* program. You can access this table from the Systems Management: Table Maintenance menu or from the Registration: Table Maintenance menu.

For Recruiting/Admissions purposes, relevant fields in the table are:

Academic Program

The academic program for the entry (e.g., UNDG for Undergraduate). Valid programs are defined in the Program table; this field must contain a value from that table. In addition, the Program code in the Admissions record (*adm_rec*) must match a code in this table. If your institution processes admissions statuses for more than one program (e.g., UNDG and GRAD), an Academic Calendar record must exist for both programs.

Official Begin Date

The first day of classes for the specified session. The *admstats* program uses this date to set statuses.

Session

The session for the entry (e.g., FA for fall). Valid Session codes are defined in the Session table; this field must contain a value from that table. The *admstats* program matches this field with the Planned Session of Enrollment field in the Admissions record.

Year

The four-digit calendar year for the entry (e.g., 1998). The *admstats* program matches this field with the Planned Year of Enrollment field in the Admissions record. The *admstats* program will also evaluate the Beginning Date in the Academic Calendar record and schedule the beginning and ending dates of the Admissions Statistics records based upon this date.

Communication Table

For information about setting up this table, see *CX System Reference Technical Manual*.

ADR Runcode Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Alternate Address Table

Within Recruiting/Admissions, for the purposes of creating School records for foreign schools, you must maintain a code of ABBR in this table. The data conversion process uses the ABBR code when creating ID records and Alternate Address records for foreign schools. For more information about setting up this table, see *CX System Reference Technical Manual*.

Relationship Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Addressing Record

For information about setting up this table, see *Communications Management Technical Manual*.

Contact Table

The Contact table validates the types of communications between your institution and its applicants and can designate the type of output created for outgoing correspondence (it serves the same purpose for constituents, donors, parents, and others with whom your institution corresponds). Within Recruiting/Admissions, the table can also update a student's enrollment status.

Example: The following values in the Contact table update any student with the APPLIED contact to an enrollment status of APPLIED. Only the relevant fields appear in this example.

Code
APPLIED
Description
Applied - fee paid
Tickler
ADM
Comm Code
MEMO
Routing
I
ACE report
ltradmit
Enrollment Status
APPLIED

Example: The following values in the Contact table create an acceptance letter (and mailing label) and update the letter recipient to an enrollment status of ACCEPTED. Only the relevant fields appear in this example.

Code
ACPTLTR
Description
Acceptance letter
Tickler
ADM
Comm Code
LTLB
Routing
O
ACE report
ltradmit
Run Code

SINGLE
Enrollment Status
 ACCEPTED

The following table indicates how to create a Contact record, depending on the purpose of each entry:

Contact purpose	Contact name	Contact status	Expected date	Complete date
Update status	INQUIRED	C	Not relevant	Current date
Create letter/label	CATALOG	E (updated to (C)ompleted after the letter/label is complete)	Current date	blank (updated to current date after the letter/label is complete)
Create letter/label and update status	ACPTLTR	E (updated to (C)ompleted after the letter/label is complete)	Current date	blank (updated to current date after the letter/label is complete)
Track miscellaneous documents	e.g., ESSAY	E (if not received; C if received)	Current date	blank, or date document was received

When you create Contact records in the Contact Detail window, Jenzabar suggests the following:

- If the routing column value is (O)utgoing, leave the contact status as (E)xpected.
- If the routing column value is (I)ncoming, change the contact status to (C)ompleted and complete the Contact's Completion Date field.

Notes:

- This is valid only if you properly defined the contacts in the contact table (ctc_table.rte).
- This does not include the contacts you use to track documents. Although documents are incoming contacts, you do not update the contact's status to (C)ompleted until you receive the document in the office.

County Table

For information about setting up this table, see *CX System Reference Technical Manual*.

State Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Zip Code Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Hold Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Hold Action Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Entry Selection/Sort Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Entry Selection/Sort Criteria Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Office Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Office Permissions Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Division Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Department Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Privacy Act Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Privacy Act Field Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Accomplishment Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Citizen Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Degree Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Exam Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Handicap Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Interest Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Involvement Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Marital Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Suffix Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Title Table

For information about setting up this table, see *CX System Reference Technical Manual*.

Visa Table

The Visa table (visa_table) defines valid visa types recognized by your institution. You can access this table from the Systems Management: Table Maintenance menu or from the Recruiting/Admissions: Table Maintenance: Common menu.

Important fields in the table are:

Code

The four-character code for the type of visa (e.g., blank or F-1).

Description

Three text lines, each 32 characters in length, to describe the individuals with the visa type (e.g., Blank code or Student visa).

Setting Up Multiple Admissions Programs

Introduction

Jenzabar delivers the standard Recruiting/Admissions module set up to process two types of students: undergraduate and graduate. If your admissions office also processes inquiries and applications for additional programs (e.g., students in professional or continuing education programs), you must modify and add files to the system. This section outlines the steps you must follow before you can use *admentry* to process students for an academic program other than undergraduate or graduate.

Example Process

To illustrate the process, assume your institution wants to set up a new program to track medical student inquiries and applicants in the Recruiting/Admissions module. The process for setting up the program follows:

1. Create new Tickler and Program codes
2. Create a new program macro
3. Modify the PROG_VALID macro
4. Create a new Tickler macro and modify the TICK_VALID macro
5. Modify the ADM_TICK_VALID macro
6. Create a new menu option file
7. Create a new form file
8. Create a new inquiry form for the new program
9. Create a new application form for the new program
10. Add a new menu option to the menu
11. Enable letter creation with a new contact Tickler code
12. Create a new WPVI FileCabinet
13. Create ACE reports for medical admissions letter creation
14. Reinstall menuopt files to recognize the new Tickler and Program codes

Step 1 - Creating New Tickler and Program Codes

You must first select a new Tickler code and Program code, and add the codes to the Tickler table (*tick_table*) and Program table (*prog_table*). The new Tickler code must be added to the Tickler table first, as this code will also be used in the Program table to link to new Program code with its associated Tickler code. In this example, assume the Tickler code will be ADMM, and the Program code will be MED.

The tables require the following entries:

TICKLER TABLE		
Code.....		[ADMM]
Description.....		[Medical Admissions]
Minimum Contact Span....	[1]	
Maximum Contact Span....	[1]	
Maximum Review Span....	[1]	
Mail Recipient.....		[jdoe]
PROGRAM TABLE		
Program Code.....	[MED]	
Description.....		[Medical Program]
Sorting Order.....	[]	
Unit Factor.....		[1.000]
Program Grade Scheme..	[]	
Official Transcript Form...	[WIDE]	
Unofficial Transcript Form	[WIDE]	
Admissions Tickler.....	[ADMM]	

Step 2 - Creating a New Program Macro (Optional)

Although it is not required, you may choose to create a new macro to store the new Program code. Since macros exist for the Undergraduate and Graduate programs, you can maintain consistency by creating a macro for any new program you set up. Macros also can minimize the amount of hard coding you must do when you create ACE reports or SQL statements.

To create a new program macro, do the following:

1. Access and open the student macro file by entering the following commands at the UNIX prompt:

```
cd $CARSPATH/macros/custom
make co F=student
vi student
```
2. Search for the PROG_UNDG macro in the file, then use *vi* commands to insert a line that resembles the following immediately below the PROG_UNDG macro:

Example: `m4_define(`PROG_MED', `MED')`
3. Save and exit the file.
4. Temporarily install the file by entering the following command at the UNIX prompt:

```
make tinstall F=student
```

Step 3 - Modifying the PROG_VALID Macro

You must modify the table macro file to include the code you have selected for the new program, to ensure the system will recognize your code as valid.

To modify the PROG_VALID macro, do the following:

1. Enter the following commands at the UNIX prompt:

```
cd $CARSPATH/macros/custom
make co F=table
vi table
```
2. Search for the PROG_VALID macro, and modify it to include the new Program code, as in the following example:

Example: `m4_define(`PROG_VALID', `UNDG,GRAD,MED')`
3. Save and exit the file.
4. Temporarily install the file by entering the following command at the UNIX prompt:

```
make tinstall F=table
```


Step 4 - Creating a New Tickler Macro and Modifying the TICK_VALID Macro

Just as you have set up the new Program code in macro files, you must also set up a new macro to define a new tickler for the medical school tickler.

To set up the tickler macro, do the following:

1. Enter the following commands at the UNIX prompt:
cd \$CARSPATH/macros/custom
make co F=common
vi common
2. Search for the TICK_ADMG macro in the file, then use *vi* commands to insert a line that resembles the following immediately below the TICK_ADMG macro:
Example: `m4_define(`TICK_ADMM',`ADMM')`
3. Search for the TICK_VALID macro, and add the new TICK_ADMM code to the list of other existing codes.
4. Save and exit the file.
5. Temporarily install the file by entering the following command at the UNIX prompt:
make tinstall F=common

Step 5 - Modifying the ADM_TICK_VALID Macro

You must also add the new Tickler code to the admissions macro file as follows:

1. Enter the following commands at the UNIX prompt:
cd \$CARSPATH/macros/custom
make co F=admissions
vi admissions
2. Search for the ADM_TICK_VALID macro, and add the new ADMM Tickler code to the list of valid Tickler codes used by the admissions office.
Example: `m4_define(`ADM_TICK_VALID',`ADM,ADMG,ADMM')`
3. Save and exit the file.
4. Temporarily install the file by entering the following command at the UNIX prompt:
make tinstall F=admissions

Step 6 - Creating a New Menu Option File

You must create a new menuopt file to pass the new Tickler and Program code parameters to the *admentry* program. Other parameters, such as the Office code and the menu form, may need to be changed. The easiest way to create a new menuopt file is to copy the existing menuopt file used for undergraduate admissions entry, and then modify the copy for medical student admissions entry.

To create the new menuopt file, do the following:

1. Enter the following commands at the UNIX prompt:
cd \$CARSPATH/menuopt/admit/programs
cp adme madme
chmod 660 madme
make add F=madme
vi madme
2. Change the menu option screen title to read "MEDICAL ADMISSIONS ENTRY".
Example: `m4_center_clipped(MEDICAL ADMISSIONS ENTRY,40)`

3. Change the short description to read "Medical Admissions Entry".

Example: SD: optional,
default = "Medical Admissions Entry";
4. Change the program parameter to use the new medical Program code.

Note: You can hard-code the program parameter (i.e., MED), or use the PROG_MED macro if you created a PROG_MED macro in Step 2.

Example: PA3: optional,
default = "MED";
5. Change the tickler parameter to use the new medical admissions Contact Tickler code.

Note: You can hard-code the tickler parameter (i.e., ADMM), or use the new TICK_ADMM macro created in Step 4.

Example: PA5: optional,
default = "ADMM";
6. If a separate office handles the admissions process for students applying to medical school, a different Office code should be passed for the office parameter. This Office code must also be added to the Office table (ofc_table). In this example, assume the Office code for the medical admissions office is "MADM", and that you have already added it to the ofc_table.

Example: PA7: optional,
default = "MADM";
7. If medical school applicants use a separate application form, then a different form menu should be passed for the menu form name parameter. In this example, assume the medical admissions form menu name is abbreviated as "madmmenu".

Note: A file by the same name must be added to the \$CARSPATH/modules/admit/progscr/admentry directory which is explained in *Step 7 - Creating a New Form Menu*.

Example: PA9: optional,
default = "madmmenu";
8. Save and exit the file.
9. Temporarily install the file by entering the following command at the UNIX prompt:
make tinstall F=madme

Step 7 - Creating a New Form Menu

You may use a variety of unique forms and reply cards to address the needs of the medical school admissions office. If so, you can copy the forms used for undergraduate or graduate admissions, and then change them as needed.

Note: Assume for the purposes of this illustration that the standard forms (dec_1, id_1, sch_1, church_1, bus_1) loaded by the *madmmenu* file do not require modification. If they do need to be changed, follow steps similar to those outlined in the procedure below.

To create a new form menu, do the following:

1. Enter the following commands at the UNIX prompt:
cd \$CARSPATH/modules/admit/progscr/admentry
cp admmenu madmmenu

```
chmod 660 madmmenu
make add F=madmmenu
vi madmmenu
```

2. Change the GUI title to read "Medical Admissions Form Menu".

Example: SCREEN_INFO: optional,
gui_title = "Medical Admissions Form Menu",

3. If the questions on the Inquiry card for medical admissions are different from the Inquiry card for undergraduate admissions, change the name of the inquiry form (e.g., to "minq").

scr1: optional,
default = "minq",

4. If the questions on the application for medical admissions are different from the application for undergraduate admissions, change the name of the application form (e.g., to "mlongapp").

scr2: optional,
default = "mlongapp",

5. If the decision form for the medical school admissions office is different from the form used for undergraduate admissions, change the name of the decision form (e.g., to "mdec").

scr3: optional,
default = "mdec",

6. Save and exit the file.

7. Temporarily install the file by entering the following command at the UNIX prompt:

```
make tinstall F=madmmenu
```

Step 8 - Creating a New Inquiry Form for a New Program

You may also have to create a new form for collecting medical student inquiry information. If so, you can copy the form used for standard admissions, and then change it as needed.

To create a new inquiry form, do the following:

1. If you completed Item 3 of Step 7, enter the following commands at the UNIX prompt:

```
cd $CARSPATH/modules/admit/progscr/admentry
cp inq_1 minq_1
chmod 660 minq_1
make add F=minq_1
vi minq_1
```

2. Modify the minq file to match the questions asked on the inquiry card used by the medical admissions office, including a change to the GUI title to read "Medical Inquiry Form".

Example: SCREEN_INFO: optional,
center,
gui_title = "Medical Inquiry Form";

3. Save and exit the file.

4. Temporarily install the file by entering the following command at the UNIX prompt:

```
make tinstall F=minq_1
```

Step 9 - Creating a New Application Form for a New Program

To customize the application form for the medical school program, do the following:

1. If you completed Item 4 of Step 7, enter the following commands at the UNIX prompt:

```
cd $CARSPATH/modules/admit/progscr/admentry
cp longapp_1 mlongapp_1
chmod 660 mlongapp_1
make add F=mlongapp_1
vi mlongapp_1
```

2. Modify the mlongapp file to match the questions asked on the Application form used by the medical admissions office. Be sure to change the GUI title to read "Medical Application Form".

Example: SCREEN_INFO: optional,
center,
gui_title = "Medical Application Form";

3. Save and exit the file.
4. Temporarily install the file by entering the following command at the UNIX prompt:
make tinstall F=mlongapp_1
5. If more than one page/screen is needed to accommodate all questions on the application form, repeat the above procedure by copying longapp_2 to mlongapp_2, and copying longapp_3 to mlongapp_3.

Note: Repeat the above procedure if you completed Item 5 of Step 7 and need a separate decision form (dec_1 to mdec_1) for the medical admissions office.

Step 10 - Adding a New Menu Option to the Menu

After you create the forms and menuopts to accommodate the additional academic program, you must add the menu option to the menu.

To add a menu option, do the following:

1. Enter the following commands at the UNIX prompt:
cd menusrc/admit/screens
make co F=menudesc
vi menudesc
2. Search for the line that adds the menu option for undergraduate admissions entry (*adme*).
3. Use *vi* commands to copy the line to the line below it (or to wherever you want the menu option to appear on the menu), then modify the new menu option line look like the example shown below. This line causes the menudesc file to call the newly created menu option for medical admissions entry.

Example: MNU_OPT(admit/programs/madme)

4. Save and exit the file.
5. Temporarily install the file by entering the following command at the UNIX prompt:
make tinstall F=menudesc

Step 11 - Enabling Letter Creation with the New ADMM Contact Tickler Code

To be able to create letters using the new ADMM Tickler code, you must modify the *ltbrun* script.

To enable letter creation, do the following:

1. Enter the following commands at the UNIX prompt:
cd modules/common/scripts
make co F=ltbrun
vi ltbrun

2. Search for the section of the *ltbrun* script that looks like the example shown below and add the new TICK_ADMM macro.

```
set tickcodes = ( TICK_ADM  TICK_ADMG  TICK_ADMM  TICK_LEAD  TICK_DEV  TICK_PLAC  " "  )
set tickmod   = ( admit      admit      admit      admit      develop  placement common )
set usrmod    = ( admissions admissions admissions admissions development placement `groups` )
```

Note:

- The tickmod line represents the path in which the ACE report used for letter creation (e.g., *ltradmit*) may be found (e.g., `$CARSPATH/modules/admit/reports`). For the Recruiting/Admissions module, this value is always “admit”, and must have an entry in the script.
- The set usrmod line represents the word processing FileCabinet where the form letter resides if you are using *nroff* to create your letters (e.g., `$WPPATH/admissions/FileCabinet/letters`).
- If the medical admissions office wants to keep its form letters separate from the form letters used by the undergraduate admissions office, you must continue with steps 12 and 13.

Step 12 - Creating a New WPVI FileCabinet

If the medical admissions office is using *nroff* to create its letters, and wants to keep its form letters in a letters drawer separate from other admissions offices on campus, you must create a new WPVI FileCabinet in which to store the letters.

To create a new WPVI FileCabinet (in this example, named `med_admit`), do the following:

1. Enter the following commands at the UNIX prompt:
cd \$WPPATH
mkdir med_admit
2. Set the UNIX permissions for the individuals in the medical admissions office so the new `med_admit` File Cabinet will be visible from the WPVI Word Processing menu. Access the permissions file by entering the following commands at the UNIX prompt:
cd /etc
SU csh
<password>
vi group
3. Add the `med_admit` group to the `/etc/group` file, ensuring the group ID number is unique. Be sure to add the login names of all the end users working in the medical admissions office, and anyone else who may need permissions to the `med_admit` FileCabinet.

Example: `med_admit::666:jdoh,dcross,egreene,nbenson,qhibbert`

After the new group has been created and an end user has logged onto the system after being added to the new permissions group, that user can create WPVI drawers within the `med_admit` file cabinet. The standard WPVI drawers that should be created are: Merge, letters, reports, wpreports, wastebasket.

To create the drawers, do the following from the CX menu:

1. Select Word Processing.
2. Enter all the following commands:
o med_admit
n Merge
n letters
n reports
n wpreports

n wastebasket

The final process in this procedure is to change the name of the FileCabinet in the *ltbrun* script file.

To change this name, do the following:

1. Enter the following commands at the UNIX prompt:
cd \$CARSPATH/modules/common/scripts
vi ltbrun
2. For the TICK_ADMM column, change the name of the file cabinet (on the set usrmod line) from *admissions* to *med_admit*, as in the following example:

```
set tickcodes = ( TICK_ADM   TICK_ADMG  TICK_ADMM TICK_LEAD  TICK_DEV   TICK_PLAC ""           )
set tickmod =   ( admit      admit    admit    admit     develop   placement common )
set usrmod =    ( admissions admissions med_admit admissions development placement `groups` )
```

3. Save and exit the file.
4. Temporarily install the file by entering the following command at the UNIX prompt:
make tinstall F=ltbrun

Step 13 - Creating ACE Reports for Medical Admissions Letter Creation

ACE reports such as *ltradmit* will look for the form letter in the letters drawer of the admissions FileCabinet. If you want an ACE report to look for a form letter in the new FileCabinet (*med_admit*), you must copy the *ltradmit* ACE report to a new file name, and modify it to look for form letters in the letters drawer of the *med_admit* FileCabinet.

To make this modification, do the following:

1. Enter the following commands at the UNIX prompt:
cd \$CARSPATH/modules/admit/reports
cp ltradmit ltrmed
chmod 660 ltrmed
make add F=ltrmed
vi ltrmed
2. Remove all revision control information (RCS) from the Header section of the ACE report.
3. Search for the word pattern "*admissions*" at the beginning of the format section of the ACE report. Change the word "*admissions*" to "*med_admit*" as shown in the example below.

Example: LTB_FORMAT(,"med_admit",id_no,name,resrc)

4. Save and exit the file.
5. Temporarily install the file by entering the following command at the UNIX prompt:
make tinstall F=ltrmed

Note: You may need to repeat this procedure for other ACE reports that are used to create letters/labels. To determine if this is necessary, review the list of ACE reports used for letter creation in the ACE reports section of this manual.

Step 14 - Reinstalling Menuopt Files to Recognize the New Tickler and Program Codes

The final step to setting up a new program is to install all the modifications you have made. To perform the reinstall, enter the following commands at the UNIX prompt:

```
cd menuopt/admit  
make reinstall F=ALL
```

Creating School Records for High Schools from a Data File

Introduction

A required step in the implementation process is the creation of ID records and School records for all schools. A School record (`sch_rec`) for a school is similar to a Profile record for an individual, defining the unique characteristics of schools (e.g., public or private, high school or college, total enrollment, CEEB number, etc.). School records define the valid schools you can enter on students' Education records (`ed_rec.sch_id`), the records that track each student's history of schools attended.

You can create individual ID and School records as needed, using *admentry*; however, such an approach could eventually consume hundreds of hours in data entry time. As an alternative, your institution can purchase a data file of all high schools for the entire country from either ACT (American College Testing) or ETS (Educational Testing Service). During implementation, you can easily load the information from these data files into the CX database. In fact, Jenzabar recommends that you purchase the data file and load the data into the database at the beginning of the implementation process (even before Jenzabar implementation specialists become involved) so users at your institution can begin to practice creating Education records for applicants.

ETS and SAT Data File Formats

If you purchase a high school data file, you have the choice of buying the information on electronic file or diskette formats (ACT and ETS do not distribute high school data on DAT data file format). CX is already set to read and convert information from a data file loaded into the default tape drive. If your default drive is for DAT data files and not 9-track tapes, and you instead purchase high school information on diskette format, you must complete the following steps:

1. Load the diskette into the disk drive (e.g., a: drive) of your personal computer.
2. Download the information from the high school diskette to a file on your hard drive (e.g., c: drive).
3. FTP the file from your c: drive over to your CX home directory on the CX system.
4. Copy the file from your home directory to the /tmp directory on the CX system, using the following commands at the UNIX prompt:
cd \$HOME
cp <filename> /tmp
5. Change the script that creates School records to look for the file in the /tmp directory instead of the default tape drive device. The example shown below illustrates the files that need to be modified and the commands to enter at the UNIX prompt:
cd modules/admit/others
make co F=hstape
vi hstape
6. Look for the line in the *hstape* file that looks like the example below:
Input: "dd if=/dev/tape HS_FMT"
7. Modify the line to resemble the example below:
Input: "cat /tmp/<filename>"
8. Save the changes and tinstall the file, entering the following command at the UNIX prompt:
make tinstall F=hstape

Note: To eliminate creating a local revision of the hstape file you can use the command at the UNIX prompt:

```
make unco F=hstape
```

Using this command after you have finished creating School records from the purchased diskette will revert the files back to their original form.

If you plan to always update your School records from a diskette instead of a 9-track tape, you may want to save and check in the changes made to this file. Another possibility is to make the changes mentioned above in the Install path of the CX system, using the following commands at the UNIX prompt:

```
cd $OTHPATH/admit  
vi hstape.oth
```

You can then modify the Input line as described in step 7.

Verifying Records Before Conversion

After you complete the above procedure, use the following command to be sure no ID, School, or Alternate Address records exist on your system with an ID number of zero (0). The conversion script *tpconvert* cannot correctly interpret an ID number of 0.

```
senter2 id_rec sch_rec aa_rec
```

Note: This command will display a screen for each record on which you can perform a query for an ID of 0. If you detect a record with an ID of 0, remove it.

Setting Macros in Preparation of Conversion

Before running the High School Convert/Add menu option, verify that the following macros are set to your institution's preferences.

\$CARSPATH/macros/custom/matric

```
m4_define(`SCH_TYPE_HS', `HS')  
m4_define(`SCH_TYPE_4YR_COL', `COL')  
m4_define(`SCH_TYPE_2YR_COL', `CC')
```

\$CARSPATH/macros/custom/admissions

```
m4_define(`ADM_OFFICE_DEF', `ADMS')
```

\$CARSPATH/macros/user/common

```
m4_define(`ID_AACODE_DEF', `PERM')
```

Menu Options to Create High School Records

After you modify the script, verify that your database does not have an ID of 0, and review your macro settings, you are ready to run the High School Convert/Add conversion script. The menu options to create high school records from a purchased electronic file/diskette exist on the Admissions/Data Conversion menu.

Process Flow Description of the High School Data Conversion Script

The High School data conversion process extracts data from either an ACT or ETS high school electronic file/diskette to create ID records (*id_rec*), School records (*sch_rec*), and possibly Alternate Address records (*aa_rec*). All processing is based on the assumption that no ID records exist where the ID field is equal to zero (0). The script creates and/or updates records based on the following criteria:

If data file records do not have a matching CEEB number on your Jenzabar CX database

- The conversion process adds an *id_rec* and *sch_rec*.

- If the address information on the school exceeds the length of the address fields in the id_rec, or if the address is foreign, the conversion process also adds an aa_rec for the school.
- The script adds or updates an Alternate Address record only if the Alternate Address code in the ID record associated with the school is set to “ABBR” (Abbreviated Address).

If data file records have a matching CEEB number on your Jenzabar CX database

- If an electronic file/diskette record matches on the sch_rec.ceebs field in the database, then the script will update the associated sch_rec if any of the data file information is different from that in the database record.
- Likewise, if the School record has an associated ID record and the electronic file data differs from the database, the script will update the ID record.
- If there is no ID record associated with a matched School record, the script will create an ID record.
- If the script detects a sch_rec, it updates the corresponding id_rec.
- If the script does not detect a sch_rec, it adds an id_rec for the school.

Report Output from High School Data conversion

The report output lists ID and School record information from both the database and the data file data for any updated records. This listing will indicate those fields whose values on the data file differ from the database. The report provides this information for only those schools whose records exist both on the data file and the database, and that match by CEEB number. Records added to the database are not listed on the report output. However, totals at the bottom of the report indicate all records added, matched and updated.

Creating School Records for Colleges from a Data File

Comparison of Processes for High School and College Data Files

The college data conversion process converts data from either an ACT College data file or an ETS College data file. The process of creating ID and school records for colleges from an electronic file/diskette is essentially the same as creating the same type of records for high schools. If you purchased an electronic file/diskette from ACT, you use the ACT College Convert/Add menu option located on the Recruiting/Admissions: Data Conversion menu, which runs the \$CARSPATH/modules/admit/others/actcol script. If you purchased an electronic file/diskette from ETS, you use the ETS College Convert/Add menu option located on the Recruiting/Admissions: Data Conversion menu, which runs the \$CARSPATH/modules/admit/others/etscol script. The report output of the conversion process will look the same as the high school data file report output.

Because of the similarities between High School and College data file processes, refer to *Creating School Records for High Schools from a Data File* in this section for more information.

Process Description

The data conversion process performs the following:

- Adds a sch_rec and an id_rec for those data file records that do not have a matching CEEB number in your CX database (sch_rec.ceebs).
- If a data file record's CEEB number matches that of a currently existing sch_rec, then the process updates the associated sch_rec if any of the data file information is different from the database record. Also, if the sch_rec has an associated ID record and the data file data differs from the database, the process will update the ID record. If your database does not contain an ID record associated with a matched sch_rec, the process will add an ID record.

ID Records with an ID of Zero

As with the High School data conversion process, accurate record processing depends on the absence of any ID record with an id_rec.id of zero (0).

Report Output

The report output lists ID and school information from both the database and electronic file/diskette for any updated records. This listing will indicate those fields whose value on the electronic file/diskette differs from the database. The report only provides this information on colleges whose records exist both on the data file and database and match by CEEB number. Records added to the database are not listed on the report output. However, there are totals given at the bottom of the report for all records added, matched, and updated.

Setting Up the Cron Process

Introduction

The *cron* process uses the UNIX operating system processor CRON. This processor enables users at your institution to schedule processes (e.g., scripts, reports, programs, or SQL statements) to run at regularly scheduled intervals of time. Within Recruiting/Admissions, scheduled processes include, for example:

- The Admissions Statistics program (to update the status of prospects and applicants)
- The Tickler Review program (to review the communications strategies associated with prospects, applicants, students, and other individuals or organizations)
- The *infage* script (to update individuals' ages in their Profile records)
- The process to update the admissions waitlist (to update the order of an admissions waitlist)
- The process to create all expected letters and labels and spool them to *lps* (Letter Production System).
- The output of the Letters/Labels Due report to a file in your home directory and/or to your e-mail.

To set up these types of scheduled processes, you must:

1. Review and update the *admprocess* script.
2. Review the *admprocess* crontab file.
3. Set directory and file read/write permissions.

Reviewing the *admprocess* Script

The *cron* process uses the *admprocess* script to identify the processes to run. As part of implementation (and periodically as a maintenance procedure), institutions must review the *admprocess* script to ensure it includes all the desired processes.

The *admprocess* script resides in the `$CARSPATH/modules/admit/scripts` directory.

Note that the pound sign (`#`) at the beginning of a line designates that line as a comment within the file. Therefore, any processes that appear on lines beginning with `#` are ignored by the script.

Review the *admprocess* file to determine if any changes need to be made. You may want to turn off certain processes by adding a `#` before the path variable, or you may want to turn on certain processes by removing the `#` from the path variable. You also need to determine if the parameters passed to certain processes are set to the values you want. Macros are usually used to pass parameters to the various processes.

If you want to change the *admprocess* script file, check out the file, make your edits, and check in the file, as follows:

1. At the UNIX prompt, enter the following: **make co F=admprocess**
2. At the UNIX prompt, enter the following: **vi admprocess**
3. Change the file as desired, then exit the file, saving changes.
4. At the UNIX prompt, enter the following: **make cii F=admprocess**

Reviewing the *admprocess* Crontab file

After the *admprocess* script has been set to your institution's needs, you can set up the *admprocess* file in the crontabs directory (`$CARSPATH/system/crontabs`).

Before you can set up this file, you must first determine how the environment variables are set on your Jenzabar CX system. The *admprocess* crontab file provides an example of one crontab file which runs two separate cron jobs; the first cron job runs the *admprocess* script and the second cron job runs the Letters/Labels Due report and e-mails the report's output to a specified person.

Environment variables

At the UNIX prompt, enter each of the following commands, and write down the system's response (echo) to each:

- **echo \$HOME**
- **echo \$USER**
- **echo \$LOGNAME**
- **echo \$CARSDB**
- **echo \$CARSV**
- **echo \$CARSPATH**
- **echo \$INFORMIXDIR**
- **echo \$INFORMIXSERVER**
- **echo \$TBCONFIG**

Informix database variable settings for `admprocess` crontab

- The HOME, USER, and LOGNAME environment variables should be set for the person you want to receive the email and output files from the cron jobs that are executed. You can, for example, use the login name of the MIS Director, the Director of Admissions, or someone else working in the admissions office.
- If the `admprocess` file does not look like the example below, you must modify it by defining the environment variables the Jenzabar CX system echoes back to you.

Note: You do not need to check out this file.

```
#
# runs admprocess
#
#22 21 * * * HOME=/home/carsids/admit; USER=admit; LOGNAME=admit; CARSDB=cars;
ARSV=carsi; CARSPATH=/opt/carsi; INFORMIXDIR=/opt/informix; ONCONFIG=onconf.hpde
v; INFORMIXSERVER=carshpdev; export HOME USER LOGNAME CARSDB CARSV CARSPATH INF
RMIXDIR ONCONFIG INFORMIXSERVER;/bin/csh -c '/usr/carsi/install/scp/admit/admpr
cess.scp >>/tmp/admscp.log' 2>&l;

22 21 * * * HOME=/home/carsids/admit; USER=admit; LOGNAME=admit; CARSDB=cars; C
RSV=carsi; CARSPATH=/opt/carsi; INFORMIXDIR=/opt/informix; ONCONFIG=onconf.hpde
; INFORMIXSERVER=carshpdev; export HOME USER LOGNAME CARSDB CARSV CARSPATH INFO
MIXDIR ONCONFIG INFORMIXSERVER;/bin/csh -c '/usr/carsi/install/scp/admit/admpro
ess.scp >>/tmp/admscp.log' 2>&l;
```

Interpreting and Modifying the `admprocess` File

Consider the following when you edit the `admprocess` file:

- The order in which the environment variables are defined before the word “export”, and the order in which they are each listed after the word “export” are the same. This identical order is required for the *cron* process to run without errors.
- The login name of ‘admit’ is used to define the HOME, USER and LOGNAME environment variables for both of the cron jobs shown in the sample crontab file. Although ‘admit’ is a valid login name pre-defined within the Jenzabar system, it is unlikely you would want an employee to log onto the Jenzabar system using the generic and mostly anonymous ‘admit’ login name. For both cron jobs, change the HOME, USER and LOGNAME environment variables to the login name of an actual admissions employee. Also change the word ‘admit’ appearing after the words “Admissions Processes Completed” in the first cron job, and the words “Letters/Labels Due” in the second cron job to the login name of the actual admissions employee. This login name represents the person you want to receive the e-mail messages and/or output files created by the cron jobs.
- Two “>>” symbols in front of the “/tmp/admscp.log” clause (as shown) mean that mail messages will be appended to one another. One “>” symbol in front of the “/tmp/admscp.log” clause (an alternative not shown) means that mail messages will be overwritten each time *cron* runs. Remove one of the “>” symbols to enable the system to overwrite the mail messages; otherwise leave both “>>” symbols as shown.

- The *admprocess* file cannot contain any hard carriage returns. To ensure you have not inadvertently added any carriage returns, position your cursor on the first line of the file (on the “30” in the above example), and use the <Shift-J> command to join any lines separated by returns.
- In the above example, the crontab job is set to run at 11:30 p.m. on every day of every month. (30 23 * * *). The “30” represents the number of minutes after the hour you want the process to start. The “23” is military time for the 23rd hour, or 11:00 p.m. For the greatest efficiency, Jenzabar recommends the *admprocess* script run overnight when no one is using the system.
- Save your changes in the *admprocess* file using normal *vi* commands.

Example Values for crontab Entries

The following table shows example values to enter in the crontab *admprocess* file, and the results.

Will run ...	If you enter ...				
	Minute	Hour	Day of Month	Month	Weekday
8:05 a.m. every Monday	05	08	*	*	1
4:00 p.m. of the 10th and 20th days of every month	0	16	10,20	*	*
9:15 a.m. and 2:15 p.m. every day	15	9,14	*	*	*

Selecting Days for Processing

Numeric equivalents for weekdays are as follows:

- 0=Sunday
- 1=Monday
- 2=Tuesday
- 3=Wednesday
- 4=Thursday
- 5=Friday
- 6=Saturday

If you want your crontab job to run only Mondays through Fridays, replace the * with 1-5 in your Weekday value.

Setting cron to Run

After you modify the *admprocess* file, you are ready to set *cron* to run. If you are setting up *cron* to run under someone else’s name (e.g., the MIS Director or the Director of Admissions), you must assume that person’s username by using the *superuser* command. You can use the *whoami* command to make sure the system echoes back the login name of the person you tried to become. If you are trying to set up an *admprocess* crontab file for yourself, you do not need to use the *superuser* command.

To perform these tasks, enter the following at the UNIX prompt:

```
SU su <login name of person>
<password>
whoami
crontab -l
```

Purpose of the crontab -l Command

The crontab -l command (the last command above) lets you know if the person has any crontab processes already set up under his/her login name. By entering this command, you make sure you do not overwrite any pre-existing crontab files. If the system echoes back to the screen a crontab file such as the example shown above, you must do the following:

1. Enter **exit** or **<Ctrl-d>** to revert to your own login name.
2. Use the superuser command again to assume someone else's login name.
3. Set up the crontab file for them.
4. Change the \$HOME, \$USER, and \$LOGNAME environment variables to the new login name in the *admprocess* crontab file.
5. Change the login name to which the mail messages should be sent.

If the system echoes back to the screen "crontab: can't open your crontab file.", the person does not already have a crontab file set up, and you can continue to set up the new crontab file. Enter the following command at the UNIX prompt:

```
crontab admprocess
```

If the command is successful the system will echo the following message back to the screen: "warning: commands will be executed using /usr/bin/sh"

Setting Directory and File Read/Write Permissions

When *cron* runs, it first sends any output files and mail messages to the /tmp directory, and when it finishes, it sends the files and messages to the mailbox of the person listed in the *admprocess* crontab file. Therefore you must be sure the permissions to the /tmp directory and the end user's mailbox and home directory are set so *cron* can write to the /tmp directory and to the end user's mail file and home directory.

Use the following procedure to check the permissions:

1. At the UNIX prompt, enter the following: **cd /**
2. At the UNIX prompt, enter the following: **ll | more**
3. Are the permissions for the /tmp directory set to drwxrwxrwx?
 - If yes, go to step 4.
 - If no, enter the following:
SU csh
<password>
chmod 777 tmp
4. At the UNIX prompt, enter the following:
cd /var/mail
ll <end user's login name>
5. Are the permissions for the end user's mail directory set to -rw-rw----?
 - If yes, go to step 6.
 - If no, enter the following:
chmod 660 <end user's login name>
6. Enter **exit** or **<Ctrl-d>** to revert to your own login prompt/identity.

Creating a crontab Job for a Menu User

You can add a crontab file to an end user's home directory, either by copying the *admprocess* crontab file in the system/crontabs directory to the home directory of the end user, or by creating a completely new crontab file.

Example

Assume the following:

- You have Tickler set up to schedule a contact called PHONCALL for students who have recently inquired.
- You have written an ACE report that will list students who have expected PHONCALL contacts that admission counselors need to call. You named the ACE report *cnslrcalls* and installed it in the \$CARSPATH/modules/admit/reports directory.
- You want *cron* to run this ACE report and sent the output to an admission counselor's mailbox.

Based on these assumptions, the following crontab file (assume the filename is *daily-jobs*) would execute the desired process at 8:15 a.m. each morning (Mondays through Fridays) and use *mailx* to route the results to the admission counselor with the login name *jdoe*.

```
15 08 * * 1-5 HOME=/opt/carsids/jdoe; USER=jdoe; LOGNAME=jdoe; CARSDb=cars; CARSV=carsi;
CARSPATH=/opt/carsi; TBCONFIG=tbconf.cars; INFORMIXDIR=/opt/informix; export HOME USER LOGNAME
CARSDb CARSV CARSPATH TBCONFIG INFORMIXDIR; /bin/csh -c 'sacego -q
/opt/carsi/install/arc/admit/cnslrcalls.arc | mailx -s "Students to Call Today" jdoe'
```

Note: After you set up and save the *daily-jobs* crontab file, change the permissions to the file so the end user cannot accidentally change the file using the Visual Editor menu option in the Utilities/File Options menu.

Use the following commands to change the file to read permission only, and to make sure the file is owned by the end user. The group permissions can be set to *common* or to a group to which the end user belongs (e.g., admissions).

```
cd ~<end user's login name>
chmod 440 <file name>
chown <end user login name> <file name>
chgrp admissions <file name>
```

You must then assume the login name of the end user using the *superuser* command, and set the crontab to run.

```
SU su <end user login name> (to assume the end user's name)
crontab -l (to ensure the end user does not already have a crontab file set up.)
crontab <daily-jobs> (to activate the file that cron will run)
```

Preventing a cron Process from Running

If you want to stop a crontab file from running for an end user, use the following commands as superuser:

```
SU su <end user login name>
crontab -r
crontab -l
```

The crontab -l command echoes the message: "crontab: can't open your crontab file.", indicating the crontab job was removed.

SECTION 14 - SETTING UP RECRUITING/ADMISSIONS LETTERS

Overview

Introduction

Recruiting/Admissions processes require extensive interaction with prospects, typically achieved through letters. CX's Communications Management product enables users in the admissions office to produce letters according to their unique needs and time requirements.

This manual contains information about the following Communications Management issues that relate specifically to Recruiting/Admissions:

- Using sourcing to create departmental letters
- Creating interview confirmation letters

Information Resources

Because CX's Communications Management product crosses product lines and is available for use with the Alumni/Development, Registration, and Financial products, documentation for creating letters is maintained in the following stand-alone manuals:

- Communications Management User Guide
- Communications Management Technical Manual

Creating Departmental Letters Using Sourcing

Introduction

Admissions offices may want to send a departmental letter written and signed by the appropriate Department Chair to applicants or to those who send inquiries. Admissions offices may also want to have the Tickler program schedule the DEPTLTR contact, which can then be used to generate the DEPTLTR letter.

However, the Tickler cannot determine a student's intended major, and therefore cannot schedule a specific departmental letter under which a student's intended major falls. The Tickler can only schedule the generic DEPTLTR contact.

To resolve this issue, you can use the *sourcing* feature to pull a specific Department Chair's letter into the main DEPTLTR letter.

Setting Up the Sourcing Feature

To use the sourcing feature for a departmental letter, complete the following setup to the Department table (dept_table) and the Major table (major_table):

1. Make required entries to the Department table.
2. Make required entries to the Major table.
3. Make required entry to the Contact table (e.g., DEPTLTR).
4. Create the general departmental letter (e.g., DEPTLTR).
5. Create individual departmental letters.

Creating Entries for the Department Table

You must add codes and descriptions for all the departments within your institution to the Department table. You can access the Department table within the Common submenu of the Admissions: Table Maintenance menu.

Creating Entries for the Major Table

After all departments have been added to the Department table, you must associate each major with its department in the Major table. The Major table is controlled by the registrar's office, and therefore the admissions office must work with the registrar's office for this step of the setup.

Sample values for an entry in the Major table follow:

Major Code

[BIO]

Major

[Biology]

CIP Number

[26.01]

IPEDS Select

[N]

Occupational

[N]

Entrance Restr

[]

Level

[AA]

Department

[LAS] Liberal Arts & Sciences

Note: Make sure all majors in the Major table are associated with a department. For Major codes such as UNDE (Undecided), you must add a non-existent Department code to the Department table and associate it with the UNDE major. With this approach you will be able to send a generic departmental letter to students who have not yet decided on a major.

Creating an Entry for the Contact Table

After both the Department and Major tables contain the correct entries, you can add the DEPTLTR contact to the Contact table, as in the following example:

Code

[DEPTLTR]

Description

[Departmental letter]

Tickler

[ADM]

Comm Code

[LTLB]

Routing

[O]

Span Waived

[N]

Reissued

[N]

Ace Report

[ltradmit]

Run Code

[SINGLEI]

Bulk Mail

[N]

Default type

[stdlps]

Document Tracking Type

[N]

Enrollment Status

[]

Creating the General Departmental Letter

After you have added the DEPTLTR entry to the Contact table, you are ready to compose the main DEPTLTR letter and each of the departmental letters. If you are using the CX word processor, you must add the DEPTLTR letter file to the *admissions* file cabinet/letters drawer

within WPVI. The name of the letter *must* match the name of the contact added to the Contact table.

The DEPTLTR letter file uses a series of WP_IF and/or WP_IFOR conditional statements, and the .so *nroff* command, to determine both a student's intended major and which departmental letter to source into the main DEPTLTR form letter. You must therefore make sure all majors in the Major table appear in the DEPTLTR letter file. Each major can be listed individually with a WP_IF statement, and would usually be selected when a department only offers one major. All majors within a department can be grouped together by using a WP_IFOR statement. Each .so *nroff* command must be flush left against the left margin.

Sample DEPTLTR Letter

The following example shows the use of the WP_IF and WP_IFOR macros in the general departmental letter, and demonstrates the correct syntax for the .so *nroff* command.

```
WP_HEAD(10,WP_SALUT)

I have been informed by our admissions office that you have been
accepted by CARS Solution College and that your intended major is WP_MAJOR1.

WP_IFOR(WP_MAJOR_CODE,ART,WP_MAJOR_CODE,INTR)\{
.so WP_DEPT_LTR\}
WP_IFOR(WP_MAJOR_CODE,BIO,WP_MAJOR_CODE,IMMU)\{
.so WP_DEPT_LTR\}
WP_IF(WP_MAJOR_CODE,CHEM)\{
.so WP_DEPT_LTR\}
WP_IF(WP_MAJOR_CODE,ENGL)\{
.so WP_DEPT_LTR\}
WP_IFOR(WP_MAJOR_CODE,PIAN,WP_MAJOR_CODE,VOCL)\{
.so WP_DEPT_LTR\}
WP_END
```

Creating Individual Departmental Letters

Once the DEPTLTR letter file is created, you can create the individual departmental letters. Make sure there is a departmental letter for each Department code added to the Department table.

Each departmental letter must have the same name as the Department code entered in the Department table. For example, the letter must be named LAS to match the Liberal Arts and Sciences Department code entered in the Department table. Macros such as WP_MAJOR1, WP_PLAN_ENR_YEAR, or WP_PLAN_ENR_SESS cannot be used within the departmental letters. If you want to include such macros, they must be in an opening or closing paragraph of the main DEPTLTR form letter file.

Sample LAS Departmental Letter

The following example demonstrates how you can customize your letters by department.

```
Along with the rest of the faculty within the Department of Liberal Arts and
Sciences, I would like to congratulate you on your acceptance into CARS Solution College.
We look forward to seeing you on campus.

.nf
Sincerely,

Professor Mendel, Chair
Department of Liberal Arts and Sciences
```

Processing Departmental Letters

When the DEPTLTR letters are created, the *ltradmit* ACE report will determine each student's major, and then determine (from the Major table) the department to which that major relates. It will then source the correct departmental letter into the main DEPTLTR letter at the point where the .so *nroff* command is located.

Creating Departmental Letters Using MSWord or WordPerfect

If your institution uses MSWord or WordPerfect to create letters, you must make a modification to the WP_DEPT_LTR macro within the *ltradmit* ACE report, as follows:

1. At the UNIX prompt, enter the following commands to access the *ltradmit* file:
cd \$CARSPATH/modules/admit/reports
make co F=ltradmit
vi ltradmit
2. Search for the definition of the WP_DEPT_LTR macro. In the standard CX product, this macro is defined as shown below:
**WP_MAC(WP_DEPT_LTR,`WP_MAKEPATH("admissions","letters",dept
clipped)',NP)**
3. While editing the *ltradmit* file, modify the WP_DEPT_LTR macro as in the example shown below:
WP_MAC(WP_DEPT_LTR,dept)
4. Save your changes and check in the file, using the following commands:
:wq
make cii F=ltradmit

You can now use the mail merge functionality in MSWord or WordPerfect to create conditional If/Then/Else statements to determine which departmental letter to pull into the DEPTLTR typed into your PC's word processor.

Creating Interview Confirmation Letters

Introduction

You can use CX to create a letter telling applicants the date and time of their scheduled interview for admission. You can also mention in the letter the name of their interviewer and the interviewer's office address.

Unique Aspects of Interview Confirmation Letters

The interview confirmation letter is created using contacts just like other letters, but with some minor differences, including the following:

- Variations in Contact table setup
- Creation of Interview record
- Creation of unique macros

Contact Table Setup

You must first set up your contact in the Contact table as shown in the example below. Notice that the ACE report used to create the INT1 letters is *ltrintconf*. This ACE report not only creates the INT1 letter, but will also create an Interview record (intvwrecom_rec) in the Interview/Recommendation detail window. It is therefore important that the name of the contact is either INT1, INT2, or INT3 to match the codes used within the Interview/Recommendation detail window.

Code

[INT1]

Description

[1st Interview letter]

Tickler

[ADM]

Comm Code

[LTLB]

Routing

[O]

Span Waived

[N]

Reissued

[N]

ACE Report

[ltrintconf]

Run Code

[SINGLE]

Bulk Mail

[N]

Default type

[stdlps]

Document Tracking Type

[N]

Enrollment Status

[]

Once you define the INT1 contact in the Contact table, it can be used to create a Contact record for an applicant in the Contacts detail window. Enter the date of the interview in the Complete Date column, and the time of the interview in the Time column using military time (1600 = 4:00 p.m.). Then enter the ID number of the interviewer in the Corresponding ID column. The Contact Status and Expected Date can default to (E)xpected and the current date respectively.

During letter creation, the process updates the INT1 Contact record. Values in the updated fields of the Contact record are:

- Status - (C)ompleted
- Complete Date - the date the letter was created

This Contact record maintains proof of when the INT1 letter was sent.

Interview Record Creation

Also during letter creation, the process creates an Interview record that appears on the Interview/Recommendation screen. Values in the created Interview record are:

- Code - contact name (in this case INT1)
- Intrvwr ID - the ID number entered in the Corresponding ID column for the Contact record
- Due Date - the future date of the interview entered in the Complete Date column of the INT1 Contact record
- Status - (E)xpected
- Add Date (does not appear on the screen) - the date the INT1 letter was created

When the interviewer returns the interview evaluation form, the admissions office only needs to update the INT1 Interview record to (C)ompleted, the Received date to the current date (t for today's date), and the evaluation score the interviewer assigned to the applicant.

Macro Creation

The *lrintconf* creates some unique macros that no other ACE report uses, and associates these macros with the information entered for the INT1 Contact record. The unique macros are:

WP_INTV_DATE

Associated with the date of the interview as entered in the Complete date column of the INT1 Contact record.

WP_INTV_TIME

Associated with the time of the interview as entered in the time column of the INT1 Contact record.

WP_INTV_NAME

Associated with the ID number/name of the interviewer as entered in the Corresponding ID column of the INT1 Contact record.

WP_INTV_ROOM

Associated with the office address of the interviewer if the interviewer is a faculty member.

Sample Letters

Examples of the form letter (if using WPVI), and a finished letter follow:

Sample Form Letter

WP_HEAD(10,WP_SALUT)

Listed below are the details for your interview. The faculty member who will be interviewing you will already have a copy of your application so you do not need to bring anything else with you to the interview. Good luck.

WP_NOFILL

DATE: WP_INTV_DATE

TIME: WP_INTV_TIME

INTERVIEWER: WP_FAC_TITLE WP_INTV_NAME

LOCATION: WP_INTV_ROOM

WP_FILL

If you have any questions about your interview call me at (555) 234-5678.

WP_CLOSE

WP_COUNSELOR_NAME

Admissions Counselor

WP_END

Sample Finished Letter

April 12, 1998

Mr. Curt Simpson
514 Park Ave
Loveland, OH 45140-2245

Dear Mr. Simpson,

Listed below are the details for your interview. The faculty member who will be interviewing you will already have a copy of your application so you do not need to bring anything else with you to the interview. Good luck.

DATE: May 12, 1998

TIME: 11:00 a.m.

INTERVIEWER: Dr. James Washington

LOCATION: 747 Curtiss Hall

If you have any questions about your interview call me at (555) 234-5678.

Sincerely,

Dave Hall
Admissions Counselor

SECTION 15 - ADMISSIONS IMPORT PROGRAM/ LOADING ACT AND SAT EXAM DATA FILES

Overview

Introduction

When prospective students take the ACT or SAT college entrance examination, they may request that their scores be sent to your institution. After scoring, ACT (American College Testing) or ETS (Educational Testing Services) send these examination results (in addition to other information about each student) to your institution, in either electronic file or diskette format. CX offers the capability to convert student exam score information received on ACT or SAT electronic files/diskettes, and to create Exam records (exam_rec) for each student. This process eliminates the need to manually enter examination scores for each student. Depending on the types of information on the ACT or ETS data files, you may also obtain data for the following records:

- Admission records (adm_rec)
- Education records (ed_rec)
- ID records (id_rec)
- Profile records (profile_rec)
- Site records (site_rec)

This section provides the steps for loading information from an exam electronic file or diskette into CX.

Creating Exam Records From A Data File

Introduction

CX is configured to read and convert exam score information from a data file provided by ACT (American College Testing) or ETS (Educational Testing Service). The Admissions office at your college should request that ACT and/or ETS distribute exam score information to your college via diskettes. The example data conversion process described below pertains to loading ACT examination scores. No example is given to load another type of data file, but be assured that the steps taken to load an ACT Student Search data file or any data file provided by ETS are the same.

Converting Data from Diskette Format

Once a diskette with exam score information is received from ACT or ETS, you must complete the following steps before converting and loading it.

1. Load the diskette into the disk drive (e.g., a: drive) of your personal computer.
2. Copy the information from the diskette to a file on the hard drive (e.g., c: drive) of your personal computer.
3. Rename the file, (using the Windows Explorer software). If the exam scores are from ACT, the file should be renamed to ACTEXAM. If the exam scores are from ETS, the file should be renamed SATEXAM. To verify that you have renamed the file with the correct name, you should access the menu option used to convert and load the exam score data. The menu options are located in the Admission/Data Conversion menu. For example, the menu option used to convert and load ACT exam score data is named ACT Exam Convert/Add. When this menu option is selected and the initial parameter screen is viewed, the required data file name and its required directory path location is displayed below the Output parameter. Rename your ACT exam score data file to the file name displayed on this screen.
4. Using the File Transfer Protocol software installed on your personal computer, FTP the renamed file from the c: drive of your personal computer to your home directory on CX UNIX platform. Your home directory will be named the same as your CX login name, and will be located within the "/home/carsids" directory. The exam score data file must be copied to the CX home directory of the person who will run the data conversion process.

Running the ACT Exam Data Report

The menu options to create an accurate import data file exist on the Admissions/Data Conversion menu. After the above steps have been completed, you are ready to run the ACT Exam Data Report. Jenzabar recommends you always run this report before and after you load data from an exam score data file. The ACT Exam Data report has three different sections. For each menu option appearing in the Admissions/Data Conversion menu used to convert and add a data file, there is an associated menu option appearing above/before it. This menu option will produce a report of the data file's contents without actually converting and adding the data file records into the CX database.

Section 1 of the ACT Exam Data Report: Report of Matched Records

The ACT Exam Data report checks the social security number of every record on the ACT electronic file/diskette to determine if there is already an ID record in your database with a matching social security number. If the report process detects a match by social security number, it prints identifying information from the electronic file and the matching ID record. You must review this information to determine if the matching records are for the same person or two

completely different people. An asterisk (*) displays beside the electronic file record data if it differs from the data of the person with the matching social security number.

If you determine that the person on the electronic file/diskette is the same as the person in your database, nothing needs to be done. After the data is loaded into your database, you can, if desired, manually update certain data elements flagged as differences by the asterisks on the report. The data from the electronic file is often more accurate or timely than the information in your database.

If you determine that the person on the electronic file/diskette is not the person with matching data in your database, then you must resolve the problem of the matching social security number before loading the electronic file/diskette data into your database. Determine if the social security for the person in your database is correct, or if the social security number for the person of the electronic file/diskette is correct, and make the appropriate change. If you do not make this change, the Exam records for the person on the electronic file will be assigned to the other person in your database when the data is loaded.

Example of Section 1

The following is an excerpt from Section 1 of the ACT Exam Data report:

ACT Exam Tape: Report of Matched Records										
ID NO/ TAPE NO	NAME/ADDR	SS NO/CITY	ST	ZIP	PHONE	ETH	SX			
BDATE	DENOM	CEEB	SCH ID	ENR	FA MAJ					
306499	Adelsberger, Katherine Ann	111-11-1111								
	14112 Parker St		Omaha			NE	68154	201-66-		
7874	WH F	12/07/80	LUTH	281771	299667	1998	Y	BIO		
	1 *Adelsberger, Katherine A.	111-11-1111								
	14112 Parker St		Omaha			NE	68154	201-66-		
7874	WH F	12/07/80	LUTH	281771	299667	1998	Y	BIO		

Section 2 of the ACT Exam Data Report: Report of Unmatched Records with SS Number

If the report does not find a matching social security number in the database for a record on the ACT exam score data file, then the data file record information will print in the second section of the report. Before continuing with the ACT Exam Convert/Add menu option Jenzabar highly recommends you perform the following checks using Section 2 of the report.

1. Use either the Inquiry or Application screen within the Admissions Entry program to query the CX database for each name appearing on Section 2 of the report. If the student is found in the CX database update his/her Social Security number field with the student's Social Security number printed on the report.
2. Populate the ACT/ETS Major table with any ACT numeric major codes printed in the MAJ column of the report. Numeric codes in this column for any data file record is an indication that the ACT/ETS Major Data Conversion table has not been sufficiently populated. For example, ACT may use the numeric code '123' to indicate a student intends to major in Biology, but your Admissions office uses the code BIO to indicate a Biology student. If an entry linking ACT's major code of 123 to your college's major code of BIO, does not exist in the ACT/ETS Major table, the report will print the numeric ACT code of 123 for the person in the data file. Before the menu option to convert and add a data file is used, the report should first be reviewed for numeric codes in the MAJ column. If the numeric major codes used by ACT are unfamiliar to you, use the printout provided by ACT to look up the academic major description of any numeric codes found on the report.
3. The situation that can occur with ACT numeric codes for academic majors can also occur for ethnic codes. Populate the ACT/ETS Ethnic table with any ACT numeric ethnic codes printed in the ET column of the report. If the numeric ethnic codes used by ACT are unfamiliar to you, use the printout provided by ACT to lookup the ethnic code description of any numeric codes found on the report.

4. The situation that can occur with ACT numeric codes for academic majors can also occur for denomination codes. Populate the ACT/ETS Denomination table with any ACT numeric denomination codes printed in the DENOM column of the report. If the numeric denomination codes used by ACT are unfamiliar to you, use the printout provided by ACT to lookup the denomination code description of any numeric codes found on the report.
5. Check for zeros (0) in the Sch ID column for each student listed in Section 2 of the report. A zero in the Sch ID column means there is no ID record and/or School record matching the CEEB number (which appears to the left of the Sch ID column), listed on the report. The CEEB number must be looked up on the Internet or in a College Entrance Examination Board directory of schools, (check with your library for a directory). Once the name and address of the school associated with the CEEB number is determined, all information known about the school must be entered into the CX database using the School data entry screen within the Admissions Entry program, thus creating a CX identification number for the school.
6. Run the ACT Exam report a second time to verify that any student with an updated Social Security number now appears in Section 1 of the report, and that there are no numeric codes printed in the MAJ/ET/DENOM columns, and that there are no zeros printed in the Sch ID column.

Note: You can access these tables in the Admissions/Table Maintenance/Admissions Tape (A-Z) menu. The tables include the following:

- ACT/ETS Denomination (leaddenom_table)
- ACT/ETS Ethnic (leadethnic_table)
- ACT/ETS Major (leadmaj_table)

Example of Section 2

The following is an excerpt from Section 2 of the ACT Exam Data report:

ACT Exam Tape: Report of Unmatched Records with SS Number												
TAPE NO	NAME/ADDR	SS NO/CITY	ST	ZIP	PHONE	ETH	SX					
BDATE	DENOM	CEEB	SCH ID	ENR	FA MAJ							
		2	Abbasi, Daanish A.			311-94-3418						
			5956 Tybalt Ln			Indianapolis			IN	46254		317-297-
5312	AS	M	10/31/79 UNCC		152510 56581	1998 Y	CHEM					
		3	Adams, Eric W.			384-02-5161						
			Po Box 2016			Middlebury			IN	46540		219-825-
2214		M	04/03/80			135790 -2147483648		1998	Y			

Section 3 of the ACT Exam Data Report: Report of Records without SS Number

The third section of the ACT Exam Data report lists students who did not provide their social security number when they took the ACT exam. Since the report does not have a social security number to compare for matches in the database, it will not load these tape records (it does not add records because students who do not provide social security numbers might be on your database). Section 3 of the report therefore includes information coming from the electronic file/diskette that you must manually evaluate and enter, if desired, using the Admissions Entry Inquiry screen. During data entry, you can query on the name of the person listed in Section 3 of the report. If the person does exist in the database, you only need to add or update their exam scores in the Tests/Exams detail window. If the person does not exist in the database, you need to create an ID record, Profile record, Admission record, Education record, and Exam records using the information on the report.

Example of Section 3

The following is an excerpt from Section 3 of the ACT Exam Data report:

ACT Exam Tape: Report of Records without SS Number										
TAPE NO	NAME/ADDR/EXAM			CITY		ST	ZIP	PHONE		ETH SX
BDATE	DENOM	CEEB	SCH ID	ENR	FA	MAJ				
	16	Bates, Angie G.								
		21739 Lonkey Ln								
2827	WH	F	01/08/80	PENT	130285	55066	1998	Caldwell	ID 83605	208-459-
							Y	ENG		
			ACT	10/01/1997	25	30	30	24	27	
			ACT1	10/01/1997	14	12	17	14	15	
			ACT2	10/01/1997	15	16	0	0	0	

Summary Report Information

The end of the report lists the summary subtotals of each section of the report. These three subtotals, when added together, equal the total number of records on the ACT exam data file/diskette.

Following is an excerpt of the summary information at the end of the ACT Exam Data report:

Number of tape records matched on ss_no:		1
Number of tape records with ss_no but no match:	228	
Number of tape records without ss_no:		6

Rerunning the ACT Exam Data report

After you have resolved any issues with different students having the same social security numbers, missing codes in the data conversion tables, or missing school records, you should run the report a second time to ensure all problems have been resolved. If you sent the output of the report to a file in your home directory, rename the file before performing the Convert/Add step to avoid overwriting the output file at that time. For example, after you run the report the first time you might rename the `tpconv.out` file to `tpconv.rep1`. The second time you run the report you might rename the `tpconv.out` file to `tpconv.rep2`. After you convert and add the data file you might rename the `tpconv.out` file to `tpconv.load1`. If you have shell access, use the commands shown below to rename the file. Menu users can use the Change Name of a File menu option in the Utilities: File Options menu to rename the output file.

```
cd $HOME
mv tpconv.out tpconv.rep1
```

Loading Conversion Data

After you are sure you have resolved all the error conditions or irregularities between your database and the electronic file, you can load the data into your database using the ACT Exam Convert/Add menu option on the Admissions/Data Conversion menu. If you specify that the output be sent to a file, the process routes a `tpconv.out` file to your home directory after it adds the records to the database. The report is formatted to print on wide paper (i.e., 132 characters wide).

Jenzabar recommends you print this file and review its contents for accuracy before continuing. You should also run the Admissions Statistics program after the data file has been loaded into the database, (if a contact record was added during the data conversion process).

Example of Output from tpconvert Process

The following is an excerpt from the `tpconv.out` file created by the data conversion process.

Note: The last lines of the report contains summary information about the number of records from the data file that were added to your database.

ACT Exam Tape: Report of Records Added/Updated				
The leftmost column of the report indicates what type of action was taken. An asterisk (*) in the leftmost column means that no education record was added for this person, since the ceeb number had no match in the school record.				
'*'	added id, profile, adm and exam records for new tape record			
' '	added id, profile, adm, education and exam records for tape record			
'p'	added profile record for matched ID			
'a'	added adm record for matched ID			
'e'	added education record for matched ID			
'x'	added exam record for matched ID			
'e*'	updated exam record for matched ID			
TAPE NO	ID NO	NAME		SS NO
	1	5032366	Abbasi, Daanish A.	311-94-3418
e	2	5031607	Henderson, Carole L.	281-80-3518
a	2	5031607	Henderson, Carole L.	281-80-3518
x	2	5031607	Henderson, Carole L.	281-80-3518
x	2	5031607	Henderson, Carole L.	281-80-3518
x	2	5031607	Henderson, Carole L.	281-80-3518
	3	5032367	Adams, Eric W.	384-02-5161
	4	5032368	Addison, Richard E.	419-29-2659
	5	5032369	Ailes, Erin B.	276-90-8549
	6	5032370	Alexander, Stacey R.	378-94-4404
	7	5032371	Alwine, Sarah R.	315-98-2459
Number of tape records added:				22
Number of exam records updated:				0
Number of profile/adm/ed/exam records added to matched Ids:				5

Creating Contact Records for Students Added Using Data Conversion

If the data conversion program adds a new person to your CX database (as opposed to updating information for a student who already exists in your database), an initial contact record and tickler record can also be created at the same time. A contact record will be created only if the `ENABLE_FEAT_ADD_DEF_INQ_CTC` macro has been turned on. The initial contact that will be created is determined by the value of the `INQUIRY_DEFAULT_CTC` macro. The tickler code used for all tickler records is determined by the value of the `ADM_PROG_TICK` macro.

A tickler record will be created only if the `ENABLE_FEAT_ADM_TICKLER` macro has been turned on. The tickler code used for all tickler records is determined by the value of the `ADM_PROG_TICK` macro. The tickler track used for all tickler records is determined by the value of the `ADM_TICKTRACK_DEF` macro. The tickler completion date used for all tickler records is determined by the value of the `ADM_TICK_CMPL_DATE` macro.

The menu options ACT Exam Convert/Add and ETS SAT Exam Convert/Add create ID, Profile, Admission, Education, Site, Exam, Contact, and Tickler records from a data file/diskette. If you want to create additional contact records after these records are created from the data file/diskette, you can do so quickly and easily by using the Contact Batch Entry menu option on the Communications Management menu. Follow the steps listed below.

Note: When running the Exam Convert/Add menu option to create new records from the data file/diskette, be sure to direct the output of the report to a file in your home directory.

1. From the menu, use the Copy File to Another File menu option on the Utilities/File Options menu to copy the `tpconv.out` file to another file in your home directory using a different name for the file (e.g., `tpconv.load1`). From the UNIX prompt, you can use the following commands.

```
cd $HOME
cp tpconv.out tpconv.load1
```

2. Use the Visual Editor menu option on the Utilities/File Options menu to eliminate all data in the `tpconv.load1` file except each student's ID number, name, and social security number. Use the following `vi` editor commands to complete this step:

- Move your cursor to the line where each column heading is underlined, and use the `vi` command `d1G`. This command deletes the line where your cursor resides, as well as all lines above it.

- Move your cursor below the line of the last student listed in the file, and use the *vi* command **dG**. This command deletes the line where your cursor resides and all lines below it.
- Move your cursor to the first line in the file using the **1G** command, and delete all spaces/characters/numbers before each student's ID number, (under the TAPE NO column). For example, the *vi* command **7x** will delete seven spaces/characters to the right of the cursor.
- Press **j** to advance the cursor to the next line, and then press **.** (period) to repeat the last command (e.g., 7x). Repeat the **j** and **.** command sequences until you have reached the last line in the file.

Note: Even if students appear on the report more than once, the Contact Batch Entry program will only create one Contact record for each person.

3. Save the file using the **<Shift-ZZ>** or **:wq** command.
4. Access the Contact Batch Entry menu option located on your Communications Management menu.
5. Enter the parameters to pass to the Contact Batch Entry program that will add your initial status contact.

Example:

```

CONTACT BATCH ENTRY

Tickler..... [ADM ]
Resource..... [INQUIRED]
Status..... [C]
Due Date..... [06/29/1997]
File..... [tpconv.load1]

```

6. After the Contact Batch Entry program adds the contacts for all the new students added to the system from the data file/diskette, run the Admissions Statistics program to update their admission statuses.

Data Conversion Errors

If the data conversion process does not work a tpconv.err file will be created in your home directory. Review the tpconv.err file for a clue as to why the process did not run to completion. The two most common reasons a tpconv.err file is created include:

1. The data file is not named correctly, and/or the data file is located in the wrong directory path.
2. There are records in the CX database where the ID number is equal to zero (0). The following database tables should be queried to determine if there is a row in the table where the ID number field is equal to zero. Any rows/records where the ID number is zero must be deleted from the table.
 - adm_rec
 - ctc_rec
 - ed_rec
 - exam_rec
 - id_rec
 - profile_rec
 - site_rec
 - tick_rec

Records Created from ACT Exam Data File

The records created from an ACT exam score data file, and the fields which are populated are listed below.

ID record (id_rec)

CX ID Number	Created by CX System
Fullname	Loaded from data file
Address Line 1	Loaded from data file
City	Loaded from data file
State	Loaded from data file
Zip Code	Loaded from data file
Country	Uses value of CTRY_DEF macro if a U.S. citizen
Address Code	Always set to PERM
Title	Uses MR for males and MS for females
Social Security Number	Loaded from data file
Phone Number	Loaded from data file
Correct Address	Always set to Y
Deceased	Always set to N
Add Date	Uses current date
Update Date	Uses current date
Office Added By	Uses ADM or ADMS, depending on the Office table

Profile record (profile_rec)

CX ID Number	Created by CX system
Ethnic Code	Derived from data file and conversion table
Gender	Derived from data file; either M or F
Birth Date	Loaded from data file
Denomination Code	Derived from data file and conversion table
Update Date	Uses current date
Handicap Code	Derived from data file and Handicap table
Veteran	Loaded from data file
State of Residence	Loaded from data file (used for id_rec) if a state resident

Site record (site_rec)

State of Residence	Loaded from data file (used for id_rec) if a state resident
CX ID Number	Created by CX system
Site Code	Uses value of SITE_DEF macro
Begin Date	Uses current date
Home Campus	Always set to Y

Education record (ed_rec)

CX ID Number	Created by CX system
CEEB Number	Loaded from data file
CX School ID Number	Derived from data file and sch_rec
Graduation Year	Loaded from data file
Academic Transcript Status	Always set to N
Site Code	Uses value of SITE_DEF macro

Admissions record (adm_rec)

CX ID Number	Created by CX system
CEEB Number	Loaded from data file
CX School ID Number	Derived from data file and sch_rec
Graduation Year	Loaded from data file
Academic Transcript Status	Always set to N
Site Code	Uses value of SITE_DEF macro

Admissions record (adm_rec)

CX ID Number	Created by CX system
Planned Enrollment Session	Uses value of ADM_ENR_SESS macro
Planned Enrollment Year	Set to either the current year or a future year
Financial Aid	Loaded from data file; either Y or N
Academic Program Code	Uses value of PROG_UNDG macro
Add Date	Uses current date
Academic Major Code	Derived from data file and conversion table
Counselor ID Number	Updated if ASSIGN_CNLSR_BY_SCHL macro is enabled
ACT School Choice Priority	Loaded from data file
Transfer Student	Loaded from data file; either Y or N
Full Time vs. Part Time	Set to 12 for full time. Set to 6 for part time

Exam record (exam_rec)

CX ID Number	Created by CX system
Exam Code	Uses ACT, ACT1 and ACT2, (or ACT and ACTE if prior to 1989)
Site Code	Uses value of SITE_DEF macro
Exam Completion Date	Loaded from data file
Exam Year	Loaded from data file
Score 1	Loaded from data file
Score 2	Loaded from data file
Score 3	Loaded from data file
Score 4	Loaded from data file
Score 5	Loaded from data file

Tickler record (tick_rec)

Created if ENABLE_FEAT_ADM_TICKLER macro is on

CX ID Number	Created by CX system
Tickler Code	Uses value of ADM_TICK_PROG macro
Track Code	Uses value of ADM_TICK_TRACK_DEF macro
Level Code	Uses value of ADM_TICKLEVEL_DEF macro
Completion Date	Uses value of ADM_TICK_CMPL_DATE macro
Manual On/Off Code	Always set to N
Next Review Date	Uses current date

Contact record (ctc_rec)

Created if ENABLE_FEAT_ADD_DEF_INQ_CTC macro is on

CX ID Number	Created by CX system
Tickler Code	Uses value of ADM_TICK_PROG macro
Contact Code	Uses value of INQUIRY_DEFAULT_CTC macro
Status	Always set to C
Completion Date	Uses current date
Add Date	Uses current date

Overview

Introduction

When prospective students take the ACT or SAT college entrance examination, they may request that their scores be sent to your institution. After scoring, ACT (American College Testing) or ETS (Educational Testing Services) send these examination results (in addition to other information about each student) to your institution, in either electronic file or diskette format. CX offers the capability to convert student exam score information received on ACT or SAT electronic files/diskettes, and to create Exam records (exam_rec) for each student. This process eliminates the need to manually enter examination scores for each student. Depending on the types of information on the ACT or ETS data files, you may also obtain data for the following records:

- Admission records (adm_rec)
- Education records (ed_rec)
- ID records (id_rec)
- Profile records (profile_rec)
- Site records (site_rec)

This section provides the steps for loading information from an exam electronic file or diskette into CX.

Creating Exam Records From A Data File

Introduction

The CX is configured to read and convert exam score information from a data file provided by ACT (American College Testing) or ETS (Educational Testing Service). The Admissions office at your college should request that ACT and/or ETS distribute exam score information to your college via diskettes. The example data conversion process described below pertains to loading ACT examination scores. No example is given to load another type of data file, but be assured that the steps taken to load an ACT Student Search data file or any data file provided by ETS are the same.

Converting Data from Diskette Format

Once a diskette with exam score information is received from ACT or ETS, you must complete the following steps before converting and loading it.

5. Load the diskette into the disk drive (e.g., a: drive) of your personal computer.
6. Copy the information from the diskette to a file on the hard drive (e.g., c: drive) of your personal computer.
7. Rename the file, (using the Windows Explorer software). If the exam scores are from ACT, the file should be renamed to ACTEXAM. If the exam scores are from ETS, the file should be renamed SATEXAM. To verify that you have renamed the file with the correct name, you should access the menu option used to convert and load the exam score data. The menu options are located in the Admission/Data Conversion menu. For example, the menu option used to convert and load ACT exam score data is named ACT Exam Convert/Add. When this menu option is selected and the initial parameter screen is viewed, the required data file name and its required directory path location is displayed below the Output parameter. Rename your ACT exam score data file to the file name displayed on this screen.
8. Using the File Transfer Protocol software installed on your personal computer, FTP the renamed file from the c: drive of your personal computer to your home directory on CX UNIX platform. Your home directory will be named the same as your CX login name, and will be located within the "/home/carsids" directory. The exam score data file must be copied to the CX home directory of the person who will run the data conversion process.

Running the ACT Exam Data Report

The menu options to create Exam records from an ACT exam score data file exist on the Admissions/Data Conversion menu. After the above steps have been completed, you are ready to run the ACT Exam Data Report. Jenzabar recommends you always run this report before and after you load data from an exam score data file. The ACT Exam Data report has three different sections. For each menu option appearing in the Admissions/Data Conversion menu used to convert and add a data file, there is an associated menu option appearing above/before it. This menu option will produce a report of the data file's contents without actually converting and adding the data file records into the CX database.

Section 1 of the ACT Exam Data Report: Report of Matched Records

The ACT Exam Data report checks the social security number of every record on the ACT electronic file/diskette to determine if there is already an ID record in your database with a matching social security number. If the report process detects a match by social security number, it prints identifying information from the electronic file and the matching ID record. You must review this information to determine if the matching records are for the same person or two

completely different people. An asterisk (*) displays beside the electronic file record data if it differs from the data of the person with the matching social security number.

If you determine that the person on the electronic file/diskette is the same as the person in your database, nothing needs to be done. After the data is loaded into your database, you can, if desired, manually update certain data elements flagged as differences by the asterisks on the report. The data from the electronic file is often more accurate or timely than the information in your database.

If you determine that the person on the electronic file/diskette is not the person with matching data in your database, then you must resolve the problem of the matching social security number before loading the electronic file/diskette data into your database. Determine if the social security for the person in your database is correct, or if the social security number for the person of the electronic file/diskette is correct, and make the appropriate change. If you do not make this change, the Exam records for the person on the electronic file will be assigned to the other person in your database when the data is loaded.

Example of Section 1

The following is an excerpt from Section 1 of the ACT Exam Data report:

ACT Exam Tape: Report of Matched Records										
ID NO/ TAPE NO	NAME/ADDR	SS NO/CITY	ST	ZIP	PHONE	ETH	SX			
BDATE	DENOM	CEEB	SCH ID	ENR	FA MAJ					
306499	Adelsberger, Katherine Ann	111-11-1111								
	14112 Parker St		Omaha			NE	68154	201-66-		
7874	WH F	12/07/80	LUTH	281771	299667	1998	Y	BIO		
	1 *Adelsberger, Katherine A.	111-11-1111								
	14112 Parker St		Omaha			NE	68154	201-66-		
7874	WH F	12/07/80	LUTH	281771	299667	1998	Y	BIO		

Section 2 of the ACT Exam Data Report: Report of Unmatched Records with SS Number

If the report does not find a matching social security number in the database for a record on the ACT exam score data file, then the data file record information will print in the second section of the report. Before continuing with the ACT Exam Convert/Add menu option Jenzabar highly recommends you perform the following checks using Section 2 of the report.

- Use either the Inquiry or Application screen within the Admissions Entry program to query the CX database for each name appearing on Section 2 of the report. If the student is found in the CX database update his/her Social Security number field with the student's Social Security number printed on the report.
- Populate the ACT/ETS Major table with any ACT numeric major codes printed in the MAJ column of the report. Numeric codes in this column for any data file record is an indication that the ACT/ETS Major Data Conversion table has not been sufficiently populated. For example, ACT may use the numeric code '123' to indicate a student intends to major in Biology, but your Admissions office uses the code BIO to indicate a Biology student. If an entry linking ACT's major code of 123 to your college's major code of BIO, does not exist in the ACT/ETS Major table, the report will print the numeric ACT code of 123 for the person in the data file. Before the menu option to convert and add a data file is used, the report should first be reviewed for numeric codes in the MAJ column. If the numeric major codes used by ACT are unfamiliar to you, use the printout provided by ACT to look up the academic major description of any numeric codes found on the report.
- The situation that can occur with ACT numeric codes for academic majors can also occur for ethnic codes. Populate the ACT/ETS Ethnic table with any ACT numeric ethnic codes printed in the ET column of the report. If the numeric ethnic codes used by ACT are unfamiliar to you, use the printout provided by ACT to lookup the ethnic code description of any numeric codes found on the report.

10. The situation that can occur with ACT numeric codes for academic majors can also occur for denomination codes. Populate the ACT/ETS Denomination table with any ACT numeric denomination codes printed in the DENOM column of the report. If the numeric denomination codes used by ACT are unfamiliar to you, use the printout provided by ACT to lookup the denomination code description of any numeric codes found on the report.
11. Check for zeros (0) in the Sch ID column for each student listed in Section 2 of the report. A zero in the Sch ID column means there is no ID record and/or School record matching the CEEB number (which appears to the left of the Sch ID column), listed on the report. The CEEB number must be looked up on the Internet or in a College Entrance Examination Board directory of schools, (check with your library for a directory). Once the name and address of the school associated with the CEEB number is determined, all information known about the school must be entered into the CX database using the School data entry screen within the Admissions Entry program, thus creating a CX identification number for the school.
12. Run the ACT Exam report a second time to verify that any student with an updated Social Security number now appears in Section 1 of the report, and that there are no numeric codes printed in the MAJ/ET/DENOM columns, and that there are no zeros printed in the Sch ID column.

Note: You can access these tables in the Admissions/Table Maintenance/Admissions Tape (A-Z) menu. The tables include the following:

- ACT/ETS Denomination (leaddenom_table)
- ACT/ETS Ethnic (leadethnic_table)
- ACT/ETS Major (leadmaj_table)

Example of Section 2

The following is an excerpt from Section 2 of the ACT Exam Data report:

ACT Exam Tape: Report of Unmatched Records with SS Number												
TAPE NO	NAME/ADDR	SS NO/CITY	ST	ZIP	PHONE	ETH	SX					
BDATE	DENOM	CEEB	SCH ID	ENR	FA MAJ							
		2	Abbasi, Daanish A.			311-94-3418						
			5956 Tybalt Ln			Indianapolis			IN	46254		317-297-
5312	AS	M	10/31/79 UNCC		152510 56581	1998 Y	CHEM					
		3	Adams, Eric W.			384-02-5161						
			Po Box 2016			Middlebury			IN	46540		219-825-
2214		M	04/03/80			135790 -2147483648		1998	Y			

Section 3 of the ACT Exam Data Report: Report of Records without SS Number

The third section of the ACT Exam Data report lists students who did not provide their social security number when they took the ACT exam. Since the report does not have a social security number to compare for matches in the database, it will not load these tape records (it does not add records because students who do not provide social security numbers might be on your database). Section 3 of the report therefore includes information coming from the electronic file/diskette that you must manually evaluate and enter, if desired, using the Admissions Entry Inquiry screen. During data entry, you can query on the name of the person listed in Section 3 of the report. If the person does exist in the database, you only need to add or update their exam scores in the Tests/Exams detail window. If the person does not exist in the database, you need to create an ID record, Profile record, Admission record, Education record, and Exam records using the information on the report.

Example of Section 3

The following is an excerpt from Section 3 of the ACT Exam Data report:

ACT Exam Tape: Report of Records without SS Number										
TAPE NO	NAME/ADDR/EXAM			CITY		ST	ZIP	PHONE		ETH SX
BDATE	DENOM	CEEB	SCH ID	ENR	FA	MAJ				
	16	Bates, Angie G.								
		21739 Lonkey Ln								
2827	WH	F	01/08/80	PENT	130285	55066	1998	Caldwell	ID 83605	208-459-
			ACT	10/01/1997	25	30	30	24	27	
			ACT1	10/01/1997	14	12	17	14	15	
			ACT2	10/01/1997	15	16	0	0	0	

Summary Report Information

The end of the report lists the summary subtotals of each section of the report. These three subtotals, when added together, equal the total number of records on the ACT exam data file/diskette.

Following is an excerpt of the summary information at the end of the ACT Exam Data report:

Number of tape records matched on ss_no:		1
Number of tape records with ss_no but no match:	228	
Number of tape records without ss_no:		6

Rerunning the ACT Exam Data report

After you have resolved any issues with different students having the same social security numbers, missing codes in the data conversion tables, or missing school records, you should run the report a second time to ensure all problems have been resolved. If you sent the output of the report to a file in your home directory, rename the file before performing the Convert/Add step to avoid overwriting the output file at that time. For example, after you run the report the first time you might rename the `tpconv.out` file to `tpconv.rep1`. The second time you run the report you might rename the `tpconv.out` file to `tpconv.rep2`. After you convert and add the data file you might rename the `tpconv.out` file to `tpconv.load1`. If you have shell access, use the commands shown below to rename the file. Menu users can use the Change Name of a File menu option in the Utilities: File Options menu to rename the output file.

```
cd $HOME
mv tpconv.out tpconv.rep1
```

Loading Conversion Data

After you are sure you have resolved all the error conditions or irregularities between your database and the electronic file, you can load the data into your database using the ACT Exam Convert/Add menu option on the Admissions/Data Conversion menu. If you specify that the output be sent to a file, the process routes a `tpconv.out` file to your home directory after it adds the records to the database. The report is formatted to print on wide paper (i.e., 132 characters wide).

Jenzabar recommends you print this file and review its contents for accuracy before continuing. You should also run the Admissions Statistics program after the data file has been loaded into the database, (if a contact record was added during the data conversion process).

Example of Output from tpconvert Process

The following is an excerpt from the `tpconv.out` file created by the data conversion process.

Note: The last lines of the report contains summary information about the number of records from the data file that were added to your database.

ACT Exam Tape: Report of Records Added/Updated				
The leftmost column of the report indicates what type of action was taken. An asterisk (*) in the leftmost column means that no education record was added for this person, since the ceeb number had no match in the school record.				
'*'	added id, profile, adm and exam records for new tape record			
' '	added id, profile, adm, education and exam records for tape record			
'p'	added profile record for matched ID			
'a'	added adm record for matched ID			
'e'	added education record for matched ID			
'x'	added exam record for matched ID			
'e*'	updated exam record for matched ID			
	<u>TAPE NO</u>	<u>ID NO</u>	<u>NAME</u>	<u>SS NO</u>
		1	5032366 Abbasi, Daanish A.	311-94-3418
e		2	5031607 Henderson, Carole L.	281-80-3518
a		2	5031607 Henderson, Carole L.	281-80-3518
x		2	5031607 Henderson, Carole L.	281-80-3518
x		2	5031607 Henderson, Carole L.	281-80-3518
x		2	5031607 Henderson, Carole L.	281-80-3518
		3	5032367 Adams, Eric W.	384-02-5161
		4	5032368 Addison, Richard E.	419-29-2659
		5	5032369 Ailes, Erin B.	276-90-8549
		6	5032370 Alexander, Stacey R.	378-94-4404
		7	5032371 Alwine, Sarah R.	315-98-2459
	Number of tape records added:			22
	Number of exam records updated:			0
	Number of profile/adm/ed/exam records added to matched Ids:			5

Creating Contact Records for Students Added Using Data Conversion

If the data conversion program adds a new person to your CX database (as opposed to updating information for a student who already exists in your database), an initial contact record and tickler record can also be created at the same time. A contact record will be created only if the `ENABLE_FEAT_ADD_DEF_INQ_CTC` macro has been turned on. The initial contact that will be created is determined by the value of the `INQUIRY_DEFAULT_CTC` macro. The tickler code used for all tickler records is determined by the value of the `ADM_PROG_TICK` macro.

A tickler record will be created only if the `ENABLE_FEAT_ADM_TICKLER` macro has been turned on. The tickler code used for all tickler records is determined by the value of the `ADM_PROG_TICK` macro. The tickler track used for all tickler records is determined by the value of the `ADM_TICKTRACK_DEF` macro. The tickler completion date used for all tickler records is determined by the value of the `ADM_TICK_CMPL_DATE` macro.

The menu options ACT Exam Convert/Add and ETS SAT Exam Convert/Add create ID, Profile, Admission, Education, Site, Exam, Contact, and Tickler records from a data file/diskette. If you want to create additional contact records after these records are created from the data file/diskette, you can do so quickly and easily by using the Contact Batch Entry menu option on the Communications Management menu. Follow the steps listed below.

Note: When running the Exam Convert/Add menu option to create new records from the data file/diskette, be sure to direct the output of the report to a file in your home directory.

- From the menu, use the Copy File to Another File menu option on the Utilities/File Options menu to copy the `tpconv.out` file to another file in your home directory using a different name for the file (e.g., `tpconv.load1`). From the UNIX prompt, you can use the following commands.

```
cd $HOME
cp tpconv.out tpconv.load1
```

- Use the Visual Editor menu option on the Utilities/File Options menu to eliminate all data in the `tpconv.load1` file except each student's ID number, name, and social security number. Use the following `vi` editor commands to complete this step:
 - Move your cursor to the line where each column heading is underlined, and use the `vi` command `d1G`. This command deletes the line where your cursor resides, as well as all lines above it.

- Move your cursor below the line of the last student listed in the file, and use the *vi* command **dG**. This command deletes the line where your cursor resides and all lines below it.
- Move your cursor to the first line in the file using the **1G** command, and delete all spaces/characters/numbers before each student's ID number, (under the TAPE NO column). For example, the *vi* command **7x** will delete seven spaces/characters to the right of the cursor.
- Press **j** to advance the cursor to the next line, and then press **.** (period) to repeat the last command (e.g., 7x). Repeat the **j** and **.** command sequences until you have reached the last line in the file.

Note: Even if students appear on the report more than once, the Contact Batch Entry program will only create one Contact record for each person.

8. Save the file using the **<Shift-ZZ>** or **:wq** command.
9. Access the Contact Batch Entry menu option located on your Communications Management menu.
10. Enter the parameters to pass to the Contact Batch Entry program that will add your initial status contact.

Example:

```

CONTACT BATCH ENTRY

Tickler..... [ADM ]
Resource..... [INQUIRED]
Status..... [C]
Due Date..... [06/29/1997]
File..... [tpconv.load1]

```

7. After the Contact Batch Entry program adds the contacts for all the new students added to the system from the data file/diskette, run the Admissions Statistics program to update their admission statuses.

Data Conversion Errors

If the data conversion process does not work a tpconv.err file will be created in your home directory. Review the tpconv.err file for a clue as to why the process did not run to completion. The two most common reasons a tpconv.err file is created include:

3. The data file is not named correctly, and/or the data file is located in the wrong directory path.
4. There are records in the CX database where the ID number is equal to zero (0). The following database tables should be queried to determine if there is a row in the table where the ID number field is equal to zero. Any rows/records where the ID number is zero must be deleted from the table.
 - adm_rec
 - ctc_rec
 - ed_rec
 - exam_rec
 - id_rec
 - profile_rec
 - site_rec
 - tick_rec

Records Created from ACT Exam Data File

The records created from an ACT exam score data file, and the fields which are populated are listed below.

ID record (id_rec)

CX ID Number	Created by CX System
Fullname	Loaded from data file
Address Line 1	Loaded from data file
City	Loaded from data file
State	Loaded from data file
Zip Code	Loaded from data file
Country	Uses value of CTRY_DEF macro if a U.S. citizen
Address Code	Always set to PERM
Title	Uses MR for males and MS for females
Social Security Number	Loaded from data file
Phone Number	Loaded from data file
Correct Address	Always set to Y
Deceased	Always set to N
Add Date	Uses current date
Update Date	Uses current date
Office Added By	Uses ADM or ADMS, depending on the Office table

Profile record (profile_rec)

CX ID Number	Created by CX system
Ethnic Code	Derived from data file and conversion table
Gender	Derived from data file; either M or F
Birth Date	Loaded from data file
Denomination Code	Derived from data file and conversion table
Update Date	Uses current date
Handicap Code	Derived from data file and Handicap table
Veteran	Loaded from data file
State of Residence	Loaded from data file (used for id_rec) if a state resident

Site record (site_rec)

State of Residence	Loaded from data file (used for id_rec) if a state resident
CX ID Number	Created by CX system
Site Code	Uses value of SITE_DEF macro
Begin Date	Uses current date
Home Campus	Always set to Y

Education record (ed_rec)

CX ID Number	Created by CX system
CEEB Number	Loaded from data file
CX School ID Number	Derived from data file and sch_rec
Graduation Year	Loaded from data file
Academic Transcript Status	Always set to N
Site Code	Uses value of SITE_DEF macro

Admissions record (adm_rec)

CX ID Number	Created by CX system
CEEB Number	Loaded from data file
CX School ID Number	Derived from data file and sch_rec
Graduation Year	Loaded from data file
Academic Transcript Status	Always set to N
Site Code	Uses value of SITE_DEF macro

Admissions record (adm_rec)

CX ID Number	Created by CX system
Planned Enrollment Session	Uses value of ADM_ENR_SESS macro
Planned Enrollment Year	Set to either the current year or a future year
Financial Aid	Loaded from data file; either Y or N
Academic Program Code	Uses value of PROG_UNDG macro
Add Date	Uses current date
Academic Major Code	Derived from data file and conversion table
Counselor ID Number	Updated if ASSIGN_CNLSR_BY_SCHL macro is enabled
ACT School Choice Priority	Loaded from data file
Transfer Student	Loaded from data file; either Y or N
Full Time vs. Part Time	Set to 12 for full time. Set to 6 for part time

Exam record (exam_rec)

CX ID Number	Created by CX system
Exam Code	Uses ACT, ACT1 and ACT2, (or ACT and ACTE if prior to 1989)
Site Code	Uses value of SITE_DEF macro
Exam Completion Date	Loaded from data file
Exam Year	Loaded from data file
Score 1	Loaded from data file
Score 2	Loaded from data file
Score 3	Loaded from data file
Score 4	Loaded from data file
Score 5	Loaded from data file

Tickler record (tick_rec)

Created if ENABLE_FEAT_ADM_TICKLER macro is on

CX ID Number	Created by CX system
Tickler Code	Uses value of ADM_TICK_PROG macro
Track Code	Uses value of ADM_TICK_TRACK_DEF macro
Level Code	Uses value of ADM_TICKLEVEL_DEF macro
Completion Date	Uses value of ADM_TICK_CMPL_DATE macro
Manual On/Off Code	Always set to N
Next Review Date	Uses current date

Contact record (ctc_rec)

Created if ENABLE_FEAT_ADD_DEF_INQ_CTC macro is on

CX ID Number	Created by CX system
Tickler Code	Uses value of ADM_TICK_PROG macro
Contact Code	Uses value of INQUIRY_DEFAULT_CTC macro
Status	Always set to C
Completion Date	Uses current date
Add Date	Uses current date

SECTION 16 - SETTING UP WEB APPLICATIONS

Overview

Introduction

Electronic Applications processing via the World Wide Web is an optional feature in the CX product suite. This section describes the steps for implementing the Web Application process.

Macros to Modify

Before you install the Web Application software, you must enable the feature by modifying the ENABLE_FEAT_ELECAPP macro. Set the macro value to Y. This macro appears in the following file: \$CARSPATH/macros/custom/admissions.

Affected Tables

Because the Web application displays the following tables, you must verify that they contain only valid data:

- \$CARSPATH/schema/common/taccomp
- \$CARSPATH/schema/common/tcitz
- \$CARSPATH/schema/common/tctry
- \$CARSPATH/schema/common/tcty
- \$CARSPATH/schema/common/tethnic
- \$CARSPATH/schema/common/thand
- \$CARSPATH/schema/common/tint
- \$CARSPATH/schema/common/tinvolve
- \$CARSPATH/schema/common/tmrtl
- \$CARSPATH/schema/common/tsuffix
- \$CARSPATH/schema/common/tst
- \$CARSPATH/schema/common/ttitle
- \$CARSPATH/schema/common/tvisa
- \$CARSPATH/schema/student/tcl
- \$CARSPATH/schema/student/tdeg
- \$CARSPATH/schema/student/tmajor
- \$CARSPATH/schema/student/tprog
- \$CARSPATH/schema/student/tsess

Web Display Flag Setup

The CX Web application offers the student prospect table lookup functionality where appropriate. However, for any given table (e.g., the Title table), not all table entries are valid for selection (e.g., MRMS, as a husband and wife would not submit a joint admissions application). You must therefore review the entries in the following tables that allow for Web application table lookup, and ensure that only the desired entries contain a Web Display flag of Y:

- accomp_table
- int_table
- involve_table
- prog_table
- sess_table
- title_table

Web Server Setup

The script for the Web application resides in the following file:
\$CARSPATH/modules/admit/cgi/apply. You must ensure that your institution's Web server accesses the correct path.

Verifying the Implementation

After you complete your setup of the Web Application process, perform the following verification steps:

1. Verify that you can access the Web application using your Web browser.
2. Determine that the command buttons for Reset Application, Submit Partial Application and Submit Complete Application appear correctly in a row format on the Web screen.
3. Execute a test application, using the Web software. During this process, verify that the Web Display field in the following tables works correctly:
 - \$CARSPATH/schema/common/tacomp
 - \$CARSPATH/schema/common/tint
 - \$CARSPATH/schema/common/tinvolve
 - \$CARSPATH/schema/common/tprog
 - \$CARSPATH/schema/common/tsess
 - \$CARSPATH/schema/common/ttitle
4. Verify that the test application drop down boxes show only active table values (i.e., the current date falls between the Active Date and the Inactive Date specified on the table entry).
5. Submit the test application.
6. Use ISQL tools to determine that all the data from the test application appears correctly in the following records in the CX database:
 - accomtmp_rec
 - apptmp_rec
 - edtmp_rec
 - inttmp_rec
 - invltmp_rec
7. Determine that you can access the Electronic Application program (*elecapp*) from the CX menu, ensuring that the menu option only allows access to the Tickler codes directly related to the Program code you are passing as a parameter.
8. Using the Electronic Application program, process the test application you submitted via the Web. Accept the application, then verify that all the data from the temporary records listed in step 3 appears correctly in the following CX database records:
 - aa_rec
 - accomp_rec
 - ctc_rec
 - ed_rec
 - id_rec
 - int_rec
 - involve_rec
 - profile_rec
 - site_rec
9. Determine that the Contact record created through the Electronic Application program contains the ctc_rec.tick passed via the menu option.
10. Access *admentry* and verify the data submitted using the *elecapp* program displays

correctly.

SECTION 17 - RECRUITING/ADMISSIONS MAINTENANCE PROCEDURES

Overview

Introduction

This section provides procedures you need to maintain the Recruiting/Admissions product.

The maintenance procedures in this section are organized in these groups:

- Daily
- Session-based
- Annually
- Other

Daily Maintenance

To use Recruiting/Admissions most effectively, perform the following activities daily:

1. Resolve any error messages sent by the *admstats* program.
2. Run the Letters/Labels Due report.
3. Process any electronic applications received via the Web application.
4. Run reports as needed to find data entry errors.
5. Resolve any duplicate ID record problems identified using *dupid*.
6. If using Tickler, read messages generated by *tickler* and respond accordingly.
7. Run the Eligible for Decision report in the Contact Reports menu, and update statuses of students who appear on report (if required by your office's procedures).

Session-Based Maintenance

When a new session begins, ensure the following procedures have been performed:

1. Update the following session macros in the \$CARSPATH/macros/custom/periodic file:
 - ADM_ENR_SESS
 - LEAD_ENR_SESS
2. Create Program Enrollment records for incoming students, using the Create Program Enrollment menu option in the Session Processing menu.
3. Update incoming students to final enrollment status, using the Update to Final Status menu option in the Session Processing menu.

Annual Maintenance

Once a year, perform the following maintenance operations:

1. Update the following yearly macros in the \$CARSPATH/macros/custom/periodic file:
 - ADM_ENR_YR
 - LEAD_ENR_YR
 - ADM_HS_GRAD_DATE
 - ADM_TICK_CMPL_DATE
2. Create Academic Calendar records. Academic Calendar records should be created for every term at least two years into the future. To complete this task, enlist help from the registrar since only the registrar's Table Maintenance menu offers access to the Academic

Calendar.

3. If desired, delete old Lead records and Lead Contact records, using the Remove Specific Leads menu option in the Lead Processing menu.

Other Maintenance

Some procedures you perform to keep your data current are performed on an as-needed basis. For example, you can create Lead records from ETS or SAT tapes whenever the tapes are received.

Perform the following maintenance operations as needed:

1. Create Lead records.
2. Move leads to Inquired status.

Completing Session Processing

Introduction

After students have paid tuition deposits and you have updated their admissions statuses to CONFIRMED, you must complete several steps to make student information available to the registrar's office. Users complete these steps from the Session Processing menu, and should coordinate the process with the registrar's office.

1. Create Program Enrollment records.
2. Add document contacts.
3. Update final statuses.
4. Track no-show students.

Tips for Creating Program Enrollment Records

The Create Program Enrollment menu option transfers data collected in an Admissions record to a Program Enrollment record. Students must have Program Enrollment records to register for classes. Consider the following hints to create the Program Enrollment records correctly:

- Users should complete this process before registration for each term. For example, if incoming students register for classes during orientation, the process should be complete before that time.
- You can use wildcards to specify the statuses for which you want to create Program Enrollment records. For example, to create Program Enrollment records for all students with a confirmed status (e.g., CONFIRM or CONFNOFE), enter CONF* as the status parameter.
- If your institution has both undergraduate and graduate students, you must run this process twice, once for each group.
- When you create Program Enrollment records, you must pass a parameter (prog_enr_rec.plan_grad_sess) specifying the expected graduation session. This information populates a field in the Program Enrollment record and causes the system to create a Petition to Graduate form later in the students' academic careers. The appropriate year (prog_enr_rec.plan_grad_yr) to create the form is computed based on the number of years it takes to complete the students' intended degree (e.g., a B.S. degree is typically a 4-year degree, while a M.S. might be a 2-year degree). You specify the intended degree in the Admissions record (adm_rec.deg). The typical time it takes to complete each degree is recorded in the deg_table.yrs_to_compl field.

Tips for Adding Document Contacts

The Add Document Contact menu option adds required document contacts for a particular Tickler. It is a batch process. This option is useful when applications are loaded onto the system from a tape (e.g., the AMCAS tape used at medical schools). Consider these tips to add the contacts correctly:

- Before you use this menu option, add document contacts to the Contact table. If you want to add the contacts automatically, set the value in the Document Tracking Type field (ctc_table.doc_req) to R (required).
- The process locates applicants in the database who have an Admissions record for the specified program, session, and year. It then selects contacts in the Contact table with a Document Tracking Type of R and the specified Tickler code. Finally, it creates Expected document Contact records.
- If the applicant already has the required Contact records, the process does not add them again unless the existing Contact records have a status of V (void).

Tips for Updating Final Statuses

The Update Contact Records menu option adds status contacts in batch to update students to their final status with the admissions office. It performs the following steps.

Note: The statuses in these tips are determined by how the following macros are set:

- ADM_STAT_CONF
- ADM_STAT_ENR
- ADM_STAT_NOSHOW

When you run this menu option, it will execute an SQL statement to locate students with an admissions status (`adm_rec.enrstat`) of CONFIRM who have registered for classes (obtaining a `stu_acad_rec.reg_stat` of C). It adds the ENROLLED contact.

The menu option also locates students with CONFIRM status who have not registered for classes (obtaining a `stu_acad_rec.reg_stat` of N). For these students, it adds the NOSHOW contact.

Finally, the option locates students with ENROLLED status who have withdrawn from classes (obtaining a `stu_acad_rec.reg_stat` of W). It adds the NOSHOW contact for these students.

After you run this menu option, you can run *admstats* to update the students' final admissions statuses. Run this option soon after the deadline to register for classes.

Tips for Locating Students Not Registered

The Students Not Registered menu option produces a report of students who do not have a status of confirmed or registered for the specified session and year. This information comes from the status field in the Student Academic record (`stu_acad_rec`).

Jenzabar recommends you run this report after registration has closed, then distribute it to the admissions counselors who have tracked the students, to determine whether or not the students are planning to enroll.

Locking and Unlocking the Admissions Record

Background

The Decision table links the Decision codes used by the admissions office (e.g., FULL) with Academic Status codes used by the registrar's office (e.g., ACPT). When a user selects the Create Program Enrollment menu option to create Program Enrollment records (prog_enr_rec) for incoming students, the process checks the Decision table to determine what academic status to assign each student in his/her Program Enrollment record.

After a Program Enrollment record is created for a student, institutions may choose to lock a student's Admissions record (adm_rec) so no other changes can be made. Whether or not the student's Admissions record becomes locked depends on the student's academic status according to his/her Program Enrollment record (prog_enr_rec.acst). The Admissions Update field in the Academic Status table (acad_stat_table.adm_upd) controls whether a particular academic status allows a student's Admissions record to be updated. A Y in the field permits updates, while an N prohibits updates.

Unlocking Admissions Records

Sometimes a student will leave your institution and then later reapply for admission. If your table and record values have prohibited the updating of the student's Admissions record, the record is considered *locked*. In these situations you must have the registrar change or blank out the student's academic status in their Program Enrollment record. This will unlock the student's Admissions record so the admissions office can begin to process the student's new application for admission.

Setup and Maintenance Procedures for Lead Processing and Data Conversion

Introduction

The following pages outline the differences between Leads and Inquiries, explain how to obtain lead information that is usable within CX, and describe how to incorporate leads into the CX database.

Distinguishing Between Leads and Inquiries

CX applies the term *Lead* to a student who has *not* initiated contact with your institution. The admissions office usually acquires the names and addresses of Leads by purchasing these names from various services such as ACT (American College Testing) or ETS (Educational Testing Services). Leads may also come from alumni, high school guidance counselors, and pastors who ask the admissions office to send information to a student, who in their opinion, would be a good prospect for the institution.

In CX, Leads are maintained in Lead records. Users can manually enter lead names and addresses from alumni or high school counselors into the system by using the Lead Entry menu option on the Lead Processing menu, or you can use a loading process to load the lead information from a purchased electronic file into your lead database.

In contrast, CX applies the term *Inquiry* to a student who has initiated contact with your institution, or who has responded to your institution's general information packet by returning a reply card. This contact may be mail, e-mail, or telephone contact from the student to the admissions office requesting information. In CX, data for students who initiated the inquiry are maintained in ID records and Admission records.

When a Lead initiates contact, therefore becoming an Inquiry, it is good practice to flag all Leads as having responded each day when reply cards are received in the mail. After all respondents have been flagged for the day, users can use the Move Leads to Inquired menu option to copy a respondent's Lead record (lead_rec) information over to an Admissions record (adm_rec), and other permanent records, in batch. You may also choose to use the Move command within the Lead Entry program to move lead data one student at a time.

Maintaining Leads in the Lead Record

CX maintains Leads in the Lead record. Leads are separate because relatively few leads actually become prospective or actual students at your institution, and data about them is not needed for your daily processing. On the other hand, when you enter an inquiry or applicant into the system using the Admissions Entry program, CX creates a permanent ID number for the student.

Example: If you were to purchase 30,000 names from ACT or ETS and immediately create ID and Admissions records for each, you would be creating 30,000 new and permanent ID numbers. However, perhaps only 5 percent of those 30,000 students will actually have an interest in attending your institution; therefore, you would have created 28,500 permanent ID numbers for students who will never inquire or apply to your institution. When you create a Lead record for a student (either manually or by loading an electronic file of purchased names), the system only creates a temporary ID number for the student. Leads who never respond to your mass mailings can eventually be removed from the system. Therefore, by creating Lead records first, you eliminate the entry of unnecessary data into your database and wasting ID numbers.

Creating Lead Records Using Data Conversion

If you purchase student search names and addresses from ETS, you have the choice of buying the information on electronic file or diskette formats (ACT and ETS do not currently distribute student search data on DAT tape format). CX is already set to read and convert information from an electronic file loaded into the default drive.

Creating Lead Records Using ACT or ETS Diskettes

Introduction

If your default tape drive is for DAT tapes and not 9-track tapes, your best alternative may be to purchase student search information on diskette format. To use diskette information to create Lead records, you must complete the following process. Typically, you will complete this process whenever you purchase ACT and ETS student search information.

1. Copy information from diskettes.
2. Alter the script that creates the records.
3. Run the ETS SS Tape report.
4. Resolve data inconsistencies.
5. Use the ETS SS Convert/Add menu option.
6. Rerun the ETS SS Tape report.
7. Change data codes and values to conform to CX conventions.

Copying Lead Records Using ACT or ETS Diskettes

The menu options to create Lead records from an ETS purchased electronic file exist on the Admissions/Data Conversion menu. If your default tape drive is for DAT tapes and not 9-track tapes, your best alternative may be to purchase student search information on diskette format.

To create Lead records from diskette, do the following:

1. Load the diskette into the disk drive (e.g., a: drive) of your personal computer.
2. Access the files on the student search diskette and copy them to a file on the hard drive (e.g., c: drive) of your personal computer.
3. Use your institution's file transfer protocol (e.g., FTP) to transmit the file from the c: drive of your personal computer over to your home directory on the CX system's UNIX platform.
4. Copy the file from your home directory to the EdVanta/tmp directory by entering the following commands at the UNIX prompt:

```
cd $HOME  
cp <filename> /tmp
```

Adapting the Script

After you obtain the data from the diskette, you must change the script that creates Lead records to access the file in the /tmp directory instead of the default tape drive device. The following example procedure illustrates the files you must modify if the student search diskettes were purchased from ETS.

1. Enter the following at the UNIX prompt:
cd modules/admit/others
make co F=etsss_rep
vi etsss_rep
2. Look for the input line in the etsss_rep file that looks like the example below and modify it to resemble the following:
As originally distributed:
Input: "dd if=/dev/tape ETSSS_FMT"
Change to:
Input: "cat /tmp/<filename>"

3. Save the changes (e.g., enter :wq) and temporarily install the file by entering the following:
make tinstall F=etsss_rep

Repeat the steps for the file *etsss* as follows:

1. Enter the following at the UNIX prompt:
cd modules/admit/others
make co F=etsss
vi etsss
2. Look for the input line in the *etsss_rep* file that looks like the example below and modify it to resemble the following:
As originally distributed:
Input: "dd if=/dev/tape ETSSS_FMT"
Change to:
Input: "cat /tmp/<filename>"
3. Save the changes (e.g., enter :wq) and temporarily install the file by entering the following:
make tinstall F=etsss

Eliminating Local Revisions

To eliminate creating a local revision of the *etsss_rep* and *etsss* files, you can use the following command after the ETS student search tape data has been loaded into Lead records.

make unco F="etsss_rep etsss"

Using this command after you have finished creating the Lead records from the purchased ETS diskette will cause the files to revert to their original form. If you plan to always load your Lead records from a diskette instead of a 9-track tape, you may want to save and check in the changes made to these files (assuming you will always copy all subsequent tape data files to the /tmp directory using the same filename). Alternatively, you can also make the same changes to the files in the CX Install path as follows:

1. At the UNIX prompt, enter the following:
cd \$OTHPATH/admit
vi etsss_rep.oth
2. Modify the Input line, save the changes, and exit the file.
3. At the UNIX prompt, enter the following:
vi etsss.oth
4. Modify the Input line, save the changes, and exit the file.

CAUTION: By making these changes to the CX Install path, you will not have Revision Control history, and if your institution reinstalls the corresponding files in the Source path, the changes in the Install path will be overwritten.

Running the ETS SS Data Report and Resolving Data Inconsistencies

After you modify the scripts as described in the previous section, you are ready to run the ETS SS Data report. You should *always* run this report before and after the student search electronic file or diskette data is loaded into the Lead record.

The report lists all students on the electronic file/diskette for whom you will create Lead records. If the report finds an existing Lead record with a social security number that matches a student's social security number on the search electronic file/diskette, the report will list the matching Lead record and the record from the electronic file/diskette. The record on the electronic file/diskette will not be added to the Lead record if a Lead record already exists for a matching social security number.

In these cases, you must resolve any matches that appear on the report and enter the data manually using the Lead Data Entry screen (under the Lead Data Entry menu option). Matches usually occur when you purchase the name and address of the same student twice, or if a Lead record was manually added to the database first and then the same name/address was purchased on an electronic file/diskette.

If a Lead record and a tape record have matching social security numbers, but do not have matching names/addresses, an asterisk (*) will appear in front of the electronic file/diskette information. In this situation you must determine which record (the Lead record or the electronic file/diskette record) has the incorrect social security number and manually resolve it in the Lead Data Entry screen.

Sample ETS Student Search Data Report

The following is an example of the report you can use to resolve differences between ETS and your existing lead information.

ETS SS Tape: Report of Matched Records										

The lead_rec from the database is followed by the tape record considered to be a possible duplicate, due to a match in the social security number. If the tape field differs from the lead field, the tape field is preceded with an asterisk.										
LEAD ID/ TAPE NO ETH	NAME/ADDRESS	SS NO/CITY	ST	ZIP	CTY	SEX	CEEB	SCH ID	BDATE	MAJOR 1 MAJOR 2

5295	Woodson, William E. 9946 Tall Oak St	523-45-6789 De Funiak Spgs	FL	32433	USA	1	100390	53652	01/28/75	053
BL										
5	Woodson, William E. 9946 Tall Oak St	523-45-6789 De Funiak Spgs	FL	32433	USA	1	100390	53652	*1/28/15	053
*UN										
5296	Vickery, Angie N. 921 New Luck Ave	623-45-6789 Lynn Haven	FL	32444	USA	2	101357	54091	12/25/75	053
UN										
6	Vickery, Angie N. 921 New Luck Ave	623-45-6789 Lynn Haven	FL	32444	USA	2	101357	54091	*2/25/15	053
UN										
ETS SS tape: Report of Unmatched Records with SS Number										

Includes tape records which have a social security but do not match a record in the lead file.										
TAPE NO ETH	NAME/ADDRESS	SS NO/CITY	ST	ZIP	CTY	SEX	CEEB	SCH ID	BDATE	MAJOR 1 MAJOR 2

102	Lyle, Natalie D. 876 Seminole Dr	654-35-6789 Auburn	AL	36830	USA	2	11914	49915	07/03/75	053
UN										
103	Baker, Sara P. 987 Cube Ct	187-65-4789 Camp Hill	AL	36850	USA	2	10035	49519	03/13/75	054
UN										
ETS SS tape: Report of Unmatched Records without SS Number										

Includes tape records which do not have a social security number.										
TAPE NO ETH	NAME/ADDRESS	SS NO/CITY	ST	ZIP	CTY	SEX	CEEB	SCH ID	BDATE	MAJOR 1 MAJOR 2

100	Dyer, Sarah A. 8765 Peachtree Cir	- - Auburn	AL	36830	USA	2	10220	49552	05/23/75	053
UN										
Number of tape records matched on ss_no: 2										
Number of tape records with ss_no but no match: 2										
Number of tape records without ss_no: 1										

Using the ETS SS Convert/Add Menu Option

After you run the ETS SS Data report, you are ready to use the ETS SS Convert/Add menu option. This menu option creates the new Lead records from the electronic file/diskette and generates a report listing the following:

- Electronic file/diskette number
- New Lead record number
- Student name
- Student social security number

The report also displays a total of the Lead records created and the Lead records *not* created (because of a match). If you did not resolve matches on the previously run ETS SS Data report, the total number of matched records that were not added should equal the number of matches listed on that previous report.

Jenzabar Recommendations for Verifying Lead Records

Jenzabar strongly recommends you route the output from the ETS SS report to a file in your home directory. You can then print the file as needed. Jenzabar also strongly recommends you rerun the ETS SS Data report after the Lead records have been created so you can compare it to the original report.

After you complete the whole process, you should have three reports in three different files in your home directory, as follows:

- A report of the data *before* it was loaded
- A report of the data that was actually loaded *during* the Convert/Add process
- A report *after* the data was loaded

Note: Consider the following renaming conventions for these reports:

- If you select *file* as your Output parameter when running the ETS SS Data report for the first time, the process will create an output file in your home directory named *tpconv.out*. You must rename this file with a new filename (e.g., *tpconv.before*) to avoid overwriting it with subsequent runs of the report. If you have menu access only (i.e., no shell access), you can rename the *tpconv.out* file using the Change Name of a File menu option located in the Utilities: File Options menu. If you have shell access, you can rename the *tpconv.out* file by entering the following command at the UNIX prompt:
cd \$HOME
mv tpconv.out tpconv.before
- If you select *file* as your Output parameter when running the ETS SS Convert/Add process, the process will create another output file in your home directory with the name *tpconv.out*. You must rename this output file to a new filename (e.g., *tpconv.during*) before you rerun the ETS SS Data report the second time.
- If you select *file* as your Output parameter when rerunning the ETS SS Data Report the second time, you should rename the *tpconv.out* file in your home directory to a new filename (e.g., *tpconv.after*).

Updating New Lead Records Using SQL

If you acquire lead information in diskette format, the students' genders are coded as a 1 or 2 instead of M or F, respectively. After you create Lead records from an ETS diskette, run the following SQL statement to update the lead_rec.sex field to M or F, the codes used in CX.

```
update lead_rec set sex = "M"
where sex = "1";

update lead_rec set sex = "F"
where sex = "2";
```

Since the ETS electronic file/diskette information does not include a student's title, run the following SQL statement to update the lead_rec.title field.

```
update lead_rec set title = "MR"
where sex = "M";

update lead_rec set title = "MS"
where sex = "F";
```

Using the Lead Records to Create Lead Contact Records

Once you have created the Lead records, use the Select by User Parameters menu option on the Lead Processing menu to create Lead Contact records in batch for all the newly added leads. You can then create a personalized letter using the *Itrlead* ACE report.

If your admissions office uses a mailing house service to send out mass mailers to Leads, you are not required to create the Lead Contact records. However, Jenzabar recommends you create Lead records before you provide the electronic file or diskette to the mailing house service. By creating the Lead records, you save data entry time when the Leads mail in their reply cards, as you will only need to flag the Leads as having responded and then move them over to an

Admissions record, using the Move Leads to Inquired menu option, or the Move command within the Lead Entry program.

Moving Leads to Inquired

Records Created

A Lead record is considered a *flat file*, meaning that all the information known about a Lead is contained in a single record. CX formats the Lead record as a flat file to work in conjunction with the electronic files that can be purchased from ACT or ETS. Lead record information includes the following:

- Title
- Name
- Address
- Phone number
- Social security number
- Gender
- Birthdate
- Ethnic code
- Current school of attendance
- Intended major
- Intended second major
- Planned session and year of attendance
- Referral source

The Move Leads to Inquired menu option breaks down a Lead record into the various records used by *admentry*. The process uses the Lead record on the following CX records:

ID record

- Title
- Name
- Address
- Phone number
- Social security number

Profile record

- Gender
- Birthdate
- Ethnic code

Education record

- School ID number (of Lead's current school)
- School CEEB number (of Lead's current school)

Admissions record

- Intended major
- Intended second major
- Planned session and year of attendance
- Referral source

Flagging Leads to Move to Inquiry

As you receive reply cards from Leads who are responding to your mass mailing, you must flag these Leads as having responded. Using the Lead Form screen, you can query for the name or social security number on each reply card. Once the correct Lead is found, use the Update command to change the Response field to Y, or use the Move command.

Results of the Move Leads Process

After the Move Leads process has completed, the prospect will have the following records

- Admissions record
- Contact record
- Education record
- ID record
- Profile record
- Site record
- Tickler record (if your office uses Tickler)

You can then run *admstats* to update the new inquiry's status (e.g., INQUIRED). You can either use the Admissions Statuses menu option on the Admissions Processing menu, or you can wait for CRON to run the *admstats* program overnight. If your admissions office uses Tickler to automatically schedule Contact records, you can also run the Tickler program using the Schedule Tickler Review menu option on the Tickler menu, or you can wait for CRON to run the Tickler program overnight.

CAUTION: If you are using the lead record to process both undergraduate and graduate leads, your institution may want to consider having two separate menu options and informers for moving undergraduate and graduate leads. The current moveleads informer uses the value of the macro `ADM_CL_DEF` to populate the `adm_rec.cl` field, and the value of the macro `ADM_TICKTRACK_DEF` to populate the `tick_rec.trk` field. Therefore, unless your institution uses the same classification and tickler track codes for both undergraduate and graduate students, one moveleads informer will not work for all students.

When the moveleads informer is run for a student, the student's lead tickler record is deactivated (by blanking out the Next Review Date), and any lead contact records with a status of "E" (Expected) are updated to a status of "V" (Voided). After the moveleads informer has run to completion, a file named "moveleads.out" will be created in your home directory. The moveleads.out file will list how many records were inserted and/or updated by the moveleads informer.

Using Lead Records to Track Both Undergraduate and Graduate Leads

If you are using the Lead record to track both undergraduate and graduate Leads, you must complete the following process to avoid creating incorrect or inconsistent records.

1. Flag the appropriate undergraduate Leads as having responded.
2. Run the Move Leads process.
3. Go back into the Lead Data Entry screen and flag all appropriate graduate Leads as having responded.
4. Run the Move Leads process again (this time passing the Program code of GRAD and the Tickler code of ADMG).

This method eliminates the possible error of creating an undergraduate Admissions record for a graduate Lead, and vice-versa.

SECTION 18 - PROGRAM ERRORS AND CRASH RECOVERY

Overview

Introduction

This section provides the following:

- A list of serious and fatal errors
- Crash recovery procedures

Note: Refer to *Recruiting and Admissions User Guide* for a list of the more common status, field error, and warning messages that can occur when menu users execute the programs in Recruiting/Admissions.

Fatal Errors

Fatal errors can occur in the following programs in Recruiting/Admissions:

- *admentry*
- *admstats*
- *callentry*

Errors from admstats

The *admstats* program abnormally terminates if any of the following occur. These errors display as standard output.

Note: In actual error messages, the system substitutes actual ID numbers or record numbers for the NNNNNs that appear in the following examples.

ADM_ID#NNNNN - Status from DBUPDATE adm_rec: NNNNN

Error in updating the Admissions record. The record has been deleted or the operator does not have permission to update the record.

ADM_ID#NNNNN, ADM_PLAN_ENR_YR 'NNNNN'-Status from GETCAL acad_cal_rec: NNNNN

Error in reading the Academic Calendar record. The record does not exist or the operator does not have permission to read the record.

ADM_ID#NNNN, ENRSTAT1-'XXXXX' 2-'XXXXX' - Status from PUTSTAT adm_stat_rec: NNNN

Error in updating the Admissions Statistics record. The record has been deleted or the operator does not have permission to update the record.

CTC_ID#NNNNN - Status from DBFIND ctc_rec NNNN

Could not find the Contact record (ctc_rec). This may occur if the record does not exist, or the operator does not have permission to the record. This is a fatal error condition.

CTC#NNNNN, CTC_ID#NNNNN - Invalid change of status - XXXXXXXX. to XXXXXXXX.

No Enrollment Sequence table (enr_seq_table) exists for this sequence of enrollment status change.

CTC#NNNNN, CTC_ID#NNNNN - Status from DBFIND adm_rec: NNNNN

Could not find the Admissions record (adm_rec) indicated. The adm_rec in question either does not exist or the operator does not have permission to read the file.

CTC#NNNNN, RESRC 'XXXXXXX' - Status from GETRES resrc_rec: NNNNN

Error when trying to read Contact table (ctc_table). This is probably because it does not exist.

CTC#NNNNN - Status from DBUPDATE ctc_rec NNNN

Could not update the Contact record (ctc_rec). This can occur if the record does not exist, or if the user does not have appropriate permission.

CTC#NNNNN - Contact not voided; Status from DBFIND ctc_rec: NNNN"

Could not update the Contact record (ctc_rec). This can occur if the record does not exist, or if the user does not have appropriate permission.

ENRSTAT_ID#NNNNN, ENRSTAT 'XXXXXXX' - Status from DBADD enr_stat_rec: NNNNN

Error in adding the Enrollment Status record (enr_stat_rec). The operator does not have permission to update the record.

No entries in enr_seq_table

No entries exist in the Enrollment Sequence table (enr_seq_table). Entries are needed to establish valid enrollment sequences.

Status from DBFIND enr_seq_table: NNNN

Could not access the Enrollment Sequence table (enr_seq_table). The user does not have permission to view or select entries in the Enrollment Sequence table.

Status from LOCK adm_stat_rec: NNNNN

Could not lock the Admissions Statistics record (adm_stat record). It is already locked or the operator does not have permissions.

Status from LOCK enr_stat_rec: NNNNN

Could not lock the Enrollment Status record (enr_stat_rec). It is already locked or the operator does not have permissions.

Status from OPEN enr_seq_table: NNNN

Could not access the Enrollment Sequence table (enr_seq_table). The user does not have permission to view the record.

Status from SELFIELD enr_seq_table: NNNN

This and similar errors indicate that the Enrollment Sequence table (enr_seq_table) could not be accessed properly. This is a fatal error condition.

Status from STRUCVIEW enr_seq_table: NNNN

This and similar errors indicate the Enrollment Sequence table (enr_seq_table) could not be properly accessed, typically because of a schema change. To resolve this error, reinstall *admentry*.

Structinit: Invalid field type in enr_stat_rec

Could not perform a *structinit* function call. This can occur if the record is locked, or if the user does not have the appropriate permissions.

WARNING: CTC#NNNNN, CTC_ID#NNNNN: Invalid change: XXXXXXXX to XXXXXXXX

During the voiding process, the Enrollment Sequence from one status to another could not be found in the Contact table (ctc_table). The record does not exist; only valid enrollment status sequences are entered into the Contact table (ctc_rec).

WARNING: VOIDED CTC#NNNNN - DBDELETE enr_stat_rec: NNNN

During the voiding process, the Enrollment Status record (enr_stat_rec) could not be deleted. The record does not exist or the operator does not have permissions.

WARNING: VOIDED CTC#NNNNN - DBFIND enr_stat_rec: NNNN

During the voiding process, the Enrollment Status record (enr_stat_rec) could not be found. The record does not exist or the operator does not have permissions.

WARNING: VOIDED CTC#NNNNN, RESRC 'XXXXXXX' - resrc_rec: NNNN

During the voiding process, the Contact Resource (ctc_resrc) could not be found in the Contact table. The record does not exist or the operator does not have permissions.

Errors in admentry

The following errors can occur in running *admentry*.

Aborting due to unexpected return from scr_getset of STATUS

This can only occur when additional return statuses are implemented in the screen handling routines. This indicates the program needs to be updated to handle the new return status.

Add field 'FIELDNAME' not found in FILENAME file.

Requires program modification. The fieldname specified in the addfld array did not appear in the view of the given filename.

Common field 'FIELDNAME' not found in FILENAME file.

Requires program modification. The fieldname specified in the commonfld array did not appear in the view of the given filename.

Field 'FIELDNAME' specified in the fldchk array was not found in the FILENAME file.

Requires program modification. An inconsistency exists in the fldchk array; the field specified does not exist in the current view of the file specified for the field.

Field 'FIELDNAME' was not found in FILENAME file, as expected by fldchk array.

Requires program modification. An inconsistency exists in the fldchk array; the specified field does not exist in the current view of the file specified for the field.

File updating has been aborted.

This message prints if an error occurs in some of the first files that are updated, causing the updates of subsequent files to be aborted.

Filename 'FILENAME' from updateorder array was unrecognized.

Requires program modification. A filename exists in the updateorder array that does not exist in the filename array.

Getkey field 'FIELDNAME' not found in FILENAME file.

Requires program modification. Either the key or a component of the key does not appear in the view of the given filename.

Help routine has not yet been implemented.

The Help feature has been invoked for a Help routine that is not yet implemented.

Invalid command

Indicates an invalid key has been entered.

Invalid key 'KEYNAME' was specified for TABLENAME file.

Requires program modification. The getkey specified for the given table does not appear in the view for that table.

Invalid record status 'CHARACTER' on file FILENAME.

The program cannot process the record. This can occur when the definition of the record statuses has been changed but not yet fully implemented, or when the buffer has unexpectedly been overwritten.

Libentry: Parameters not found: auto_mode

Program modification required. The entry library requires the variable to be defined in the prog_params array.

Libentry: Parameters not found: critical_files

Program modification required. The entry library requires the variable to be defined in the prog_params array.

Libentry: Parameters not found: debug_level

Program modification required. The entry library requires the variable to be defined in the prog_params array.

Libentry: Parameters not found: menuname

Program modification required. The entry library requires the variable to be defined in the prog_params array.

Libentry: Parameters not found: scr_path

Program modification required. The entry library requires the variable to be defined in the prog_params array.

No files were bound to the '%s' form

The specified form had no fields from the files known by the program; therefore, the processing of the form was aborted.

Out of memory

Memory resources have been exhausted. Try loading fewer tables and/or files.

Permission is not granted to add records

Either the user does not have the appropriate database permission or the program does not allow additions of this record type.

Permission is not granted to delete records

Either the user does not have the appropriate database permissions or the program does not allow deletions of this record type.

Status occurred on scr_get of STATUS entmain.c

Indicates the menu form does not have a screen field with the name of 'a'.

Table "TABLENAME" specified in the fldchk array was not found in tablename array.

Requires program modification. A tablename is used in the fldchk array that is not defined in the tablename array.

The FILENAME file's putkey, 'KEYNAME', is not unique. STATUS

The putkey specified in the filename array for the specified filename is not a unique key. All putkeys must be unique keys.

The FILENAME record appears to have been deleted. Please check.

Can occur on an update or delete of a record if the program retrieved a record it did not lock, and before the record could be updated or deleted, someone deleted the record from the database.

The FILENAME record was modified before your changes. Please check.

Can occur on an update or delete of a record if the program retrieved a record it did not lock, and before the record could be updated or deleted, someone else modified the record. This can also occur if one of the programmer-supplied routines modified the record's previous buffer.

The form 'FORMNAME' was not found.

The form specified was expected by the program but could not be located. Verify that the form exists and has been installed.

The value of FIELDNAME code, 'FIELDVALUE', was not found in 'TABLENAME'.

The user entered a value that was not found in the table.

Update field 'FIELDNAME' not found in FILENAME file.

Requires program modification. The fieldname specified in the updatefld array did not appear in the view of the given filename.

dmm_add: error on form FORMNAME and screen SCREENNAME.

Refer to the "Out of memory" error.

dmm_add: error on form FORMNAME and screen SCREENNAME, and file FILENAME

Refer to the “Out of memory” error.

dmm_add: error on form FORMNAME

Refer to the “Out of memory” error.

ent_bindfile: Error binding ‘%s’. %s.

Requires program modification. The given fieldname is a component of a getkey that does not appear in the view of the given filename.

ent_initdmlt. ‘FIELDNAME’ not found in common fields of FILENAME file.

Requires program modification. The given fieldname is a component of a getkey that does not appear in the view of the given filename.

ent_writerec.c: Positioning in FILENAME failed with a STATUS stats.

Refer to the Informix manual for the meaning of the status. The error indicates that, in re-finding the previous record in the database, a dbfind error occurred other than “Record not found.”

SQL Errors and Messages from *elecapp*

The following errors (shown in order of occurrence) can occur in running *elecapp*.

Declare error <SQLCODE> at line <line number> in file <filename>

Declare cursor error. See your Informix manual for more information.

Open error <SQLCODE> at line <line number> in file <filename>

Open cursor error. See your Informix manual for more information.

Fetch error <SQLCODE> at line <line number> in file <filename>

Fetch cursor error. See your Informix manual for more information.

ERROR: dbhandle <\$CARSDb> dbclose <SQLCODE>: Consult the INFORMIX manual to define this error.

Database closing error. See your Informix manual for more information.

Update error <SQLCODE> at line <line number> in file <filename>

Error occurred while updating the database. See your Informix manual for more information.

Insert error <SQLCODE> at line <line number> in file <filename>

Error occurred while inserting a record into the database. See your Informix manual for more information.

SCR_SCROLL error <SQLCODE> at line <line number> in file <filename>

Unable to define the scrolling region. See your Informix manual for more information.

Note: All modifications are made to the database by way of transaction logging. The program is therefore subject to begin, rollback, and commit errors.

Jenzabar CX Screen Package Errors in *elecapp*

The following errors from the CX screen library can occur within *elecapp*:

ERROR: <screen name> text: <scr_errm()>

Could not display the text fields on the screen.

ERROR: <screen name> display: <scr_errm()>

Could not display all the data fields on the screen.

ERROR: init <scr_errm()>

Could not initialize the screen package.

ERROR: progexit <scr_errm()>

Could not exit the screen package.

ERROR: <menuname> getmenu: <scr_errm>

Could not process the menu structure definition, and could not build the menu options across the top of the screen.

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/admit/admentry/progscr`

2. Reinstall each program screen file.

Example: `make reinstall F=<screenfilename1 screenfilename2 ...>`

Note: You can also 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/admit/admentry`

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 named 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 and reinstall the source for the entry program.

Example: `cd $CARSPATH/src/Lib/libentry`

`make reinstall`

```
cd $CARSPATH/admit/admentry
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 Support Services.

Recovery in admentry

If *admentry* exits unexpectedly, the database may need to be updated. The only time a database update occurs is after a user has selected the ID of a recruit and has executed the Finish command. All records in that form for the recruit are then added or updated. The *admentry* program then goes into Query mode to allow selection of the next recruit. The critical time is after the user executes the Finish command but before return to Query mode. The records for that recruit should all be checked for accuracy. If some were not added, they should be added again.

Recovery in admstats

If *admstats* is running when the system exits unexpectedly, the program needs only to be restarted, since *admstats* will reread all files needed for updating.

INDEX

A

- Academic Calendar record, 206
- Academic Status table, 199
- accessing
 - Admissions Entry screen, form files, 61
 - Admissions Statistic Record screen, 75
 - Caller Time Entry screen, form files, 95
 - Electronic Applications screen, form files, 87
 - evntentry program, 103
 - lead entry form screen, 109
 - schemas, 17
- Accomplishment table, 209
- ACE reports
 - actprof, 171
 - actprofrnk, 169
 - admcurstat, 171
 - adv, 170
 - allctc, 171
 - appcomp_pa, 177
 - appcomplt, 177
 - callctcs, 170
 - callinfo, 170
 - callmktctc, 170
 - callstat, 178
 - callstat2, 178
 - callstatd, 171
 - calltime, 170
 - chcurrent, 171
 - chprof, 170
 - chprosp, 171
 - chrecgrads, 171
 - cmptvsch, 175
 - cnslrctc, 171
 - cnslrsch, 169
 - collsrc, 176
 - committee, 170
 - curstat, 173
 - elecapp, 169
 - elecdeptmp, 169
 - elecscn, 172
 - elecstat, 172
 - elig_dec, 177
 - enrhistpct, 172
 - enrhistyr, 169
 - enrmajor, 174
 - enrrefsrsc, 177
 - enrstat, 175
 - examrpt, 178
 - files, 118
 - geography, 172
 - hscurrent, 173
 - hsfaculty, 173
 - hslead, 173
 - hsprof, 172
 - hsprosp, 172
 - hsrecgrads, 173
 - in others directory, 178
 - in reports directory, 168
 - interview, 173
 - leadceeb, 174, 176
 - leadctcdue, 173
 - leadmaj, 173, 176
 - leadresrc, 173, 176
 - leadrpt, 178
 - leadsex, 174, 176
 - leadsrc, 174, 176
 - leadst, 174, 176
 - leadzip, 174, 176
 - letter creation, 168
 - ltradmit, 181
 - ltrenclose, 182
 - ltrincapp, 190
 - ltrintconf, 186
 - ltrlead, 184
 - ltrmistran, 187
 - ltrparent, 188
 - ltrpreapp, 189
 - ltrschlabl, 190
 - ltrschvst, 191
 - major, 174
 - majsum, 173
 - not_elig, 177
 - onectc, 171
 - profch, 175
 - profcnslr, 175
 - profhs, 175
 - profint, 175
 - profmaj, 175
 - profone, 175
 - profzip, 175
 - recstate, 175
 - refsrc, 177
 - refsrcenr, 177
 - resnreg, 178
 - resrcappl, 171
 - runcnslr, 171
 - satprofrnk, 176
 - schdcal, 170
 - schdir, 179
 - seqcount, 169
 - setmajrnk, 174
 - setwaitrnk, 175
 - sexage, 177
 - stuvst, 177
 - tadvmedium, 170
 - tcallmctc, 170
 - tcallres, 176

tdec, 172
 tenrseq, 172
 tenrstat, 172
 tleadctry, 172
 tleaddenom, 169
 tleadethnc, 169
 tleadmaj, 169
 tref, 175
 treqadmdoc, 175
 ACT Composite Scores menu option, 136
 ACT Exam Convert/Add menu option, 144
 ACT Exam Data Report, 238, 247
 ACT Profile vs. HS Rank menu option, 136
 ACT/ETS Denomination Rep. Table menu option, 154
 ACT/ETS Denomination Table menu option, 154
 ACT/ETS Denomination Table screen, 155
 ACT/ETS Ethnic Table menu option, 154
 ACT/ETS Ethnic Table Report menu option, 154
 ACT/ETS Ethnic Table screen, 155
 ACT/ETS Major Table menu option, 154
 ACT/ETS Major Table Report menu option, 154
 ACT/ETS Major Table screen, 156
 ACT_EXAM_DEF, 34
 ACT_REF_SRC, 38
 Activity Schedule menu option, 129
 Add Contacts/School Visit menu option, 123
 Add Document Contacts menu option, 138
 Add Pre-Transfer Courses menu option, 123
 adding
 menu options onto menus, 216
 temporary workers to admtemp, 202
 Addressing record, 207
 ADM_ACE_DEF, 38
 ADM_CL_DEF, 34
 ADM_COMP_CTC, 34
 ADM_CTC_APPT_DEF, 37
 ADM_CTC_DEF, 34
 ADM_DEC_DEF_CTC, 34
 ADM_DOC_1, 41
 ADM_DOC_10, 41
 ADM_DOC_11, 42
 ADM_DOC_12, 42
 ADM_DOC_2, 41
 ADM_DOC_3, 41
 ADM_DOC_4, 41
 ADM_DOC_5, 41
 ADM_DOC_6, 41
 ADM_DOC_7, 41
 ADM_DOC_8, 41
 ADM_DOC_9, 41
 ADM_DOC_DEF, 38
 ADM_DOC_EX, 38
 ADM_DOC_INCL, 38
 ADM_DOC_VALID, 38
 ADM_EDGOAL_DEF, 34
 ADM_EDGOAL_EX, 34
 ADM_EDGOAL_INCL, 34
 ADM_EDGOAL_VALID, 34
 ADM_ENR_SESS, 45
 ADM_ENR_YR, 45
 ADM_EXAM_1, 38
 ADM_EXAM_2, 38
 ADM_EXAM_3, 38
 ADM_EXAM_4, 38
 ADM_HS_GRAD_DATE, 45
 ADM_INQ_CL_DEF, 34
 ADM_INTENDHRS_DEF, 36
 ADM_INTENDHRS_EX, 37
 ADM_INTENDHRS_INCL, 37
 ADM_INTENDHRS_VALID, 37
 ADM_LTR_ACE_INCL, 39
 ADM_MAX_WAITLIST, 32
 ADM_OFFICE_DEF, 35
 ADM_PROG_MENU, 35
 ADM_PROG_OFFICE, 35
 ADM_PROG_PROGRAM, 35
 ADM_PROG_TICK, 35
 ADM_REPORT_SORT_DEF, 38
 ADM_STAT_ACPT, 35
 ADM_STAT_CONF, 36
 ADM_STAT_DECLINE, 36
 ADM_STAT_ENR, 36
 ADM_STAT_EXCLUDE, 35
 ADM_STAT_NOFE, 35
 ADM_STAT_NOSHOW, 36
 ADM_STAT_WAIT, 35
 ADM_STAT_WTHDRW, 36
 ADM_TICK_CMPL_DATE, 45
 ADM_TICK_VALID, 37, 213
 ADM_TICKLEVEL_DEF, 37
 ADM_TICKLEVEL_INCL, 37
 ADM_TICKLEVEL_VALID, 37
 ADM_TICKTRACK_DEF, 37
 ADM_WAITLIST_TRACK_BY, 32
 admdocctc script, 161
 admdocone script, 161
 Admission History record, 18
 Admission record, 18
 Admission Record Query screen, 156
 Admission Record screen, 156
 Admissions (A-Z) Tables menu, 151
 Admissions Entry
 database files, 57
 description, 10
 Admissions Entry menu option, 121
 Admissions Info screen, 155
 Admissions menu option, 139
 Admissions Processing menu, 120
 admissions programs
 setup, 211
 Admissions record, 269

- Admissions Statistic Record screen, 157
- Admissions Statistics
 - description, 10
 - program, 223
 - record, 19
- Admissions Statistics menu option, 139
- Admissions Status Contact screen, 157
- Admissions Statuses menu option, 121
- Admissions Tape Conversion (A - Z) menu option, 153
- Admissions/Service Entry menu option, 120
- admit.cgi scripts, 81
- admprocess
 - Crontab file, 223
 - file, 224
 - script, 166, 223
- admstats script, 166
- admtemp
 - adding temporary workers, 202
 - removing temporary workers, 202
- ADR Runcode table, 206
- Advertisement Record menu option, 152
- Advertisement Record screen, 157
- Advertisement Report menu option, 152
- Advertising Medium Record screen, 157
- Advertising Medium Table menu option, 152
- Advertising Medium Table Report menu option, 152
- Alternate Address table, 207
- APP_DEFAULT_CTC, 31
- Applicant Committee Review menu option, 132
- Application Documents screen, 157
- Applications Documents Table menu option, 152
- Applications Documents Table Report menu option, 152
- apply.cgi script, 81
- Appointments - 1 Counselor menu option, 128
- Appointments - 3 Counselors menu option, 129
- ASSIGN_CNCLR_BY_SCHL, 39

B

- background
 - for implementation, 194
- Batch Elec Apps menu option, 149
- Biographical - One menu option, 120

C

- C programs, 27
- Call Contact Schedule menu option, 146
- Call Entry Contact Table Report menu option, 148
- Call entry Entry menu option, 146
- Call Entry Information menu option, 146
- Call entry menu, 146
- Call entry Reports menu, 146

- Call Entry Result table, 19
- Call Entry Result Table menu option, 146
- Call Entry Result Table screen, 158
- Call Entry Results record, 20
- Call Entry Student record, 20
- Call entry Time record, 20
- Call Result Table Report menu option, 148
- CALL_CTC_COMM_CODE, 38
- CALL_INTEREST_INCL, 39
- CALL_RESP_DEF, 39
- CALL_RESP_EX, 39
- CALL_RESP_INCL, 39
- CALL_RESP_VALID, 39
- Caller Entry
 - description, 11
- Caller Time Entry menu option, 146, 147
- Caller Time Entry screen, 158
- Calls Completed menu option, 147
- chprosp script, 166
- chreport script, 166
- Church Profile menu option, 132
- Churches menu option, 140
- Citizen table, 209
- colldayctc script, 162
- college data files, 219, 222
- columns
 - table, 13
- common tables and records
 - in elecapp, 83
- Communication table, 206
- Communications Management menu, 123, 140
- Competitive Schools menu option, 128
- Contact record
 - in Admissions Statistics, 67
 - voiding in Admissions Statistics, 67
- Contact records, 242, 251
- Contact Reports menu, 126
- Contact Resources menu option, 134
- Contact table, 203, 207, 231, 234
- Contact Table
 - maintaining, 70
- Contacts by Counselors menu option, 126
- Contacts by Prospect menu option, 126
- Contacts for All Students menu option, 126
- Contacts for One Student menu option, 126
- conventions, 3
- Counselor Reports menu, 128
- Counselor School/Prospect menu option, 129
- Country table, 203
- Counts, Subtotals menu option, 132
- County table, 208
- Create All Letters/Labels menu option, 124, 125
- Create Lead Letters/Labels menu option, 141
- Create Program Enrollment menu option, 137
- Create School Letters menu option, 124
- creating

- ACE reports for letter creation, 218
- application forms. *See* form menus, 214
- inquiry forms, 215
- interview confirmation letters, 234
- Lead Contact records, 268
- Lead records using data conversion, 263
- macros, 213
- menu option files, 213
- new program macros, 212
- new tickler and program codes, 211
- new WPVI FileCabinet, 217
- School records
 - procedure, 219
- School records from purchased data files, 219, 222
- cron
 - preventing running, 227
 - process, 223
- crontab entries, 225
- Csh Scripts, 166
- Current Statuses menu option, 134

D

- Daily Stats by Caller menu option, 147
- data conversion, 264
 - lead processing, 262
- Data Conversion
 - description, 10
- Data Conversion menu, 144
- data dictionary, 17
- data flow descriptions
 - Event Entry program, 99
- dbefield report, 17
- dbefile report, 17
- dbetrack report, 17
- dec.h files, 51
- Decision table, 20, 199
- Decision Table menu option, 152
- Decision Table Report menu option, 152
- Decision Table screen, 158
- def.c files, 51–53
 - example, 52
- definition files. *See* def.c files
- definitions
 - SQL tables, 13
 - tables, records, 17
- Degree table, 209
- deleting temporary workers from admtemp, 202
- Denomination table, 204
- Department table, 209, 230
- departmental letters, 231
 - processing, 232
 - using sourcing, 230
- diagrams

- process flow, 98, 99
- process flow for Recruiting/Admissions module, 9
- differences
 - in product, 1
- diskettes
 - in data conversion, 238, 247
- Display Financial Aid menu option, 122
- Display Registrations menu option, 122
- Division table, 209
- dmls, 51
- dmlts, 51
- dmms, 51
- document contacts
 - creating, 259
- documents, related, 2

E

- Education record, 269
- Elec App Reports menu, 150
- Elec. Applications Menu, 148
- Elecapp DUP SSN menu option, 151
- Elecapp Education Records menu option, 151
- Elecapp Status menu option, 150
- Electronic Application
 - description, 10
- Electronic Application menu option, 150
- Eligible for Decision menu option, 128
- ENABLE_ADMHIST_PROMPT, 33, 34
- ENABLE_EARLY_DECISION, 32
- ENABLE_FEAT_ADD_DEF_APP_CTC, 31
- ENABLE_FEAT_ADD_DEF_DEC_CTC, 31
- ENABLE_FEAT_ADD_DEF_INQ_CTC, 30
- ENABLE_FEAT_ADD_DEF_LEAD_CTC, 43
- ENABLE_FEAT_ADM_BUSINESS, 32
- ENABLE_FEAT_ADM_GRAD, 31
- ENABLE_FEAT_ADM_STUSERV, 30
- ENABLE_FEAT_ADM_TICKLER, 30
- ENABLE_FEAT_ADM_WAITLIST, 32
- ENABLE_FEAT_AUTO_ADMSTATS, 32
- ENABLE_FEAT_AUTO_DOC_TRACK, 32
- ENABLE_FEAT_CALLENTRY, 31
- ENABLE_FEAT_CHURCH, 30
- ENABLE_FEAT_DEC_RPT, 31
- ENABLE_FEAT_DISPLAY_FA, 30
- ENABLE_FEAT_DISPLAY_REGIST, 30
- ENABLE_FEAT_ELECAPP, 33
- ENABLE_FEAT_LEAD_PROCESSING, 30, 43
- ENABLE_FEAT_LEAD_TICKLER, 43
- ENABLE_FEAT_PHONE_CHECK, 33
- ENABLE_FEAT_PREAPP, 33
- ENABLE_FEAT_SSNO_CHECK, 33
- ENABLE_IMMUNIZATIONS, 32
- ENABLE_SCHEDULE_ENTRY, 33
- ENABLE_VIEW_FIN_AID, 32

- enabling
 - letter creation for new Tickler codes, 216
- Enroll History - Percent menu option, 133
- Enroll History - Years menu option, 133
- Enrollment Sequence table, 20, 202
 - in Admissions Statistics, 67, 69
- Enrollment Sequence Table menu option, 153
- Enrollment Sequence Table Report menu option, 153
- Enrollment Sequence Table screen, 159
- Enrollment Status menu option, 139
- Enrollment Status record, 21
- Enrollment Status screen, 158
- Enrollment Status table, 21, 200, 203
- Enrollment Status Table menu option, 153
- Enrollment Status Table Report menu option, 153
- Enrollment Status Table screen, 159
- entering contacts, 70
- Entry Selection/Sort Criteria table, 209
- Entry Selection/Sort table, 209
- environment variables, 224
- Ethnic table, 204
- ETS Country Table menu option, 153
- ETS Country Table Report menu option, 154
- ETS Country Table screen, 159
- ETS data files
 - creating School records, 219, 222
- ETS Data report, 265
- ETS SAT Exam Convert/Add menu option, 144
- ETS_REF_SRC, 38
- Event Entry
 - parameters, 101
 - program, 97, 98
- Event Entry - Schedule Activities window, 103
- Event Entry - Schedule Entry Parameters screen, 103
- Event Entry - Schedule Entry screen, 103
- exam data files
 - loading, 237, 246
- Exam record, 238, 247
- Exam table, 209
- Examination Scores menu option, 130
- examples
 - crontab job, 226
 - departmental letter for sourcing, 232
- Expected Contacts menu option, 127

F

- fatal errors
 - in Admissions Entry, 273
 - in Admissions Statistics program, 271
 - in Electronic Application, 275
- fields
 - completing, 17

- files
 - dec.h, 51
 - def.c, 51–53
 - input to Admissions Entry, 57
 - input to Admissions Statistics, 67, 71
 - mac.h, 51
 - updated by Admissions Entry, 58
 - updated by Admissions Statistics, 67, 72
- financial aid, 11
- functions
 - macros, 28

G

- general departmental letters, 231
- Geographical Distribution menu option, 137
- getrec.ph script, 81
- glossary entries
 - additional information, 2
- GRAD_DOC_CTGRY_DEF, 40, 41
- Graduate Admission Entry menu option, 121

H

- Handicap table, 209
- Hold Action table, 209
- Hold table, 208
- hsprosp script, 167
- hsreport script, 167

I

- ID record, 269
- ID/Profile Relationship menu option, 139
- incapp script, 162
- includes, 27
 - application, 47
 - general information, 46
 - locating, 46
 - purpose, 46
 - Recruiting/Admissions, 27
- Inclusive Statuses menu option, 133
- individual departmental letters, 232
- infage script, 223
- input files
 - to Admissions Entry, 57
 - to Admissions Statistics, 67, 71
- INQUIRY_DEFAULT_CTC, 31
- INST_ADMISSION_PHONE, 39
- Interactive Degree Audit menu option, 122
- Interactive Lead Tickler menu option, 142
- Interactive Tickler menu option, 125
- interdependencies
 - among macros, 46
- Interest table, 209
- interview confirmation letters
 - creating, 234

Interview record, 235
Interview/Recommendation record, 21
Involvement table, 210

L

ldtable.ph script, 81
Lead Contact record, 21
Lead Country table, 22, 203
Lead Denomination table, 22, 204
Lead Entry menu option, 140
lead entry parameters screen
 purpose, 111
 screen example, 111
lead entry program
 data consistency & keystroke saving features, 113
 data tracking features, 113
 duplicate checking features, 114
Lead Entry program, 105
lead entry program features, 113
Lead Entry screen, 159
Lead Ethnic table, 22, 204
Lead Letters/Labels Due menu option, 141
Lead Letters/Labels Due Report menu option, 124
Lead Major table, 22, 205
lead processing, 262
Lead Processing menu, 140
Lead Processing Reports menu, 142
Lead record, 23
lead record fields
 additional, 110
Lead records, 262
 copying, 264
Lead Roster by Parameters menu option, 142
lead soundex program, 116
Lead Tickler menu, 141
lead tickler program, 115
Lead Tickler record, 23
Lead Tickler Review menu option, 142
Lead Tickler System Entry menu option, 141
LEAD_DEFAULT_CTC, 43
LEAD_ENR_SESS, 45
LEAD_ENR_YR, 45
LEAD_TICKTRACK_DEF, 43
LEAD_YEAR_INCL, 43
leadctc script, 124, 140, 162
leadctcdue script, 124, 141
Leads
 process description, 10
Leads by Contact Resource menu option, 143
Leads by Major menu option, 142, 143
Leads by Reference Source menu option, 143
Leads by School menu option, 143
Leads by Sex menu option, 143
Leads by State menu option, 143
Leads by Zip menu option, 144
letter creation
 ACE reports, 168
letters
 using MSWord or WordPerfect, 233
loading
 exam data files, 237, 246
local revisions, 265
locating
 students not registered, 260
locking
 Admissions records, 261
ltbrun script, 126, 141
ltradmit ACE report, 181
ltrnclose ACE report, 182
ltrincapp ACE report, 190
ltrintconf, 235
ltrintconf ACE report, 186
ltrlead ACE report, 184
ltrmistran ACE report, 187
ltrparent ACE report, 188
ltrpreapp ACE report, 189
ltrschlabl ACE report, 190
ltrschvst ACE report, 191

M

mac.h files, 51
 example, 54
macro files
 admissions, 28
 adm rpt, 28
 common, 28
 configure, 28
 leads, 28
 ltrwp, 28
 matric, 29
 periodic, 29
 table, 29
macros, 27, 194
 admissions, 30
 customizing, 28
 definition, 28
 document control, 41
 function, 28
 interdependencies, 46
 leads, 43
 location, 28
 periodic, 45
 Recruiting/Admissions, 27
 requiring yearly updates, 45
 to setup Web Application processing, 255
 using the Applocate script, 29
Macros in the Leads file, 43
maintenance

- annual, 257
- daily, 257
- other, 258
- session-based, 257
- Major table, 205, 230
- Major Table menu option, 153
- Major Table Report menu option, 153, 154
- Major Table screen, 160
- Majors - Current Statuses menu option, 136
- Majors - Inclusive Statuses menu option, 135
- manual
 - conventions, 3
 - intended audience, 1
 - purpose, 1
 - structure, 1
- Marital table, 210
- menu option files, 117. *See also* menus
- menu source files, 117. *See also* menus
- menuopts. *See* menu options
- menusr. *See* menus
- modifying
 - macros, 212
- Move Leads to Inquired menu option, 140
- moveleads script, 140, 162
- moving leads, 269
- MSWord
 - in producing letters, 233
- multiple admissions programs
 - setup, 211

N

- non-fatal errors
 - in Admissions Entry, 273
 - in Admissions Statistics program, 271
 - in Electronic Application, 275
- Not Eligible for Decision menu option, 128
- Not Received One Contact menu option, 127

O

- Office Permissions table, 209
- Office table, 209
- One Contact - All Students menu option, 127
- One Contact - Not Another menu option, 127
- One Student's Profile menu option, 131

P

- parameters
 - Admissions Entry, 59
 - Call entry, 93
 - Electronic Applications, 85
 - Event Entry, 101
 - lead entry, 108
- PERFORM (Table Maintenance) Screens, 155
- PERFORM screen files, 117

- periodic macros, 45
- pointers
 - form and sqlda, 51
- preapp script, 163
- prerequisites
 - for implementing and supporting Jenzabar CX, 7
- Print Admissions Letters menu option, 125
- Print Lead Letters/Labels menu option, 141
- Print Student Pre-Trans Crs menu option, 123
- Privacy Act Field table, 209
- Privacy Act table, 209
- process flow
 - Event Entry, 98, 99
- process flow description
 - Call entry, 91
 - lead entry, 107
- processes
 - Admissions Entry program, 56
 - Admissions Statistics, 66
 - Call entry program, 90
 - Electronic Applications
 - scripts, 81
 - Electronic Applications program, 78
 - Electronic Applications program, 82
 - for voiding contacts in Admissions Statistics, 69
 - lead entry program, 106
- processing leads, 262
- product differences, 1
- Profile record, 269
- Profile Reports menu, 130
- PROG_VALID, 212
- Program Enrollment records
 - creating, 259
- program features
 - lead entry, 113
- program relationships
 - Electronic Applications, 83
- program screens
 - Admissions Statistics, 75
 - Call entry, 95
 - Electronic Applications, 87
 - lead entry, 109
- Program table, 205
- programs
 - Admissions Entry
 - database input, 57
 - database output, 58
 - parameters, 59
 - process flow, 56
 - process flow description, 56
 - program screens, 61
 - Admissions Entry (admentry), 55
 - Admissions Statistics
 - features, 65

- process flow description, 67
- Admissions Statistics (admstats), 65
- def.c file, 52
- definition files, 51
- details, 51
- Electronic Applications, 77
 - parameters, 85
- Event Entry, 97, 98
- files, 51
- general description, 51
- in Recruiting/Admissions, 10
- mac.h files, 53
- purchased data files
 - creating School records, 219, 222
- purpose
 - lead entry parameters screen, 111
 - macros, 28
 - Recruiting/Admissions product, 7

Q

Query by Form menu, 139

R

Reassign Counselors menu option, 138

records, 13–16

- Adm History record, 18
- Admission record, 18
- Admissions Statistics record, 19
- Call Entry Results, 20
- Call Entry Student record, 20
- Call entry Time record, 20
- common, 14
- definition, 13
- Enrollment Status record, 21
- field descriptions, 17
- in Recruiting/Admissions, 18
- Interview/Recommendation record, 21
- Lead Contact record, 21
- Lead record, 23
- Lead Tickler record, 23
- required, 15
- shared, 15
- Temporary Accomplishment record, 23
- Temporary Admissions record, 24
- Temporary Alternate Address record, 24
- Temporary Application record, 24
- Temporary Education record, 24
- Temporary Identification record, 25
- Temporary Interest record, 25
- Temporary Involvement record, 25
- Temporary Profile record, 25
- Temporary Site record, 26
- Text Blob for Lead Contact record, 26

Recruiting/Admissions

- process flow diagram, 9

- Reference Sources menu option, 134
- Reference table, 200
- Reference Table menu option, 153
- Reference Table Report menu option, 153
- Reference Table screen, 160
- references. *See* documents, related
- Referral table, 23
- regctc script, 163
- registration, 11
- reinstalling
 - macros
 - during implementation, 194
 - menu option files, 218
- related applications, 11
 - Financial Aid, 11
 - Registration, 11
- related documents, 2
- Relationship table, 207
- Remove Specific Leads menu option, 140
- removing temporary workers from admtemp, 202
- repeatable enrollment statuses
 - in Admissions Statistics, 69
- reports
 - countprt, 178
 - ctcall, 179
 - ctccond, 179
 - ctcexpect, 178
 - ctcnot, 179
 - ctcrpt, 178
 - profile, 179
 - profone, 179
- Reports menu, 126
- reqadmdoc script, 164
- required tables, records, 15
- Review New Elec. Apps menu option, 148
- Review Partial Elec Apps menu option, 149
- Review Rejected Elec Apps menu option, 148
- rmlleads script, 140, 164
- rows
 - table, 13
- run time parameters. *See* parameters. *See* parameters. *See* parameters
- runcnslr script, 167

S

- SAT data files
 - creating School records, 219, 222
- SAT Profile vs. HS Rank menu option, 137
- Schedule Entry menu option, 123
- Schedule Entry program. *See* Event Entry program
- Schedule Tickler Review menu option, 125
- Scheduled Interviews menu option, 130
- schemas, 17

- naming, 17
- reports, 17
- School Directory menu option, 132
- School Profile menu option, 131
- Schools menu option, 139
- screen examples
 - lead entry parameters screen, 111
- screen files
 - aatmp, 87
 - accomptmp, 87
 - act_1, 61
 - adm, 87
 - admhist, 61, 95
 - admmenu, 61
 - aid, 61
 - app, 87
 - bus_1, 61
 - callent_1, 95
 - callent_2, 95
 - callresult, 95
 - ctc, 61, 95
 - dec_1, 61
 - dupid, 87
 - ed, 87
 - edtmp, 87
 - gadmenu, 62
 - gdec_1, 62
 - ginq_1, 62
 - glongapp_1, 62
 - glongapp_2, 62
 - glongapp_3, 62
 - immune, 62
 - inq_1, 62
 - inttmp, 87
 - intvwrecom, 62
 - invltmp, 87
 - longapp_1, 62
 - longapp_2, 63
 - longapp_3, 63
 - proftmp, 88
 - qid, 88
 - ref_1, 63
 - sat_1, 63
 - sch, 88
 - shortapp_1, 63
- screens
 - Admissions Entry, 61
 - Admissions Statistics Record, 75
 - Caller Time Entry, 95
 - Electronic Applications, 87
 - Event Entry, 103
 - Event Entry - Schedule Activities window, 103
 - Event Entry - Schedule Entry Parameters, 103
 - Event Entry - Schedule Entry screen, 103
 - lead entry, 109
- scripts
 - adapting for lead processing, 264
 - admdocctc, 161
 - admcocone, 161
 - admprocess, 166
 - admstats, 166
 - chprosp, 166
 - chreport, 166
 - colldayctc, 162
 - hsprosp, 167
 - hsreport, 167
 - incapp, 162
 - leadctc, 124, 140, 162
 - leadctcdue, 124, 141
 - ltbrun, 126, 141
 - moveleads, 140, 162
 - preapp, 163
 - regctc, 163
 - reqadmdoc, 164
 - rmleads, 140, 164
 - runcnslr, 167
 - setcnslr, 164
 - studctc, 165
 - tpconv, 167
 - updcnslr, 165
 - used in elecapp, 81
 - Web application, 81
 - Select by User Parameters menu option, 123, 140
 - Services by Student menu option, 120
 - session processing, 259
 - Session Processing menu, 137
 - Session table, 206
 - Set Counselor Territories menu option, 138
 - setcnslr script, 164
 - setup
 - Web server, 256
 - sorting
 - Contact records in Admissions Statistics, 70
 - Sources - Current Statuses menu option, 135
 - Sources - Inclusive Statuses menu option, 135
 - sourcing
 - for departmental letters, 230
 - special features
 - in Admissions Statistics, 69
 - SQL
 - statement files, 117
 - table definition, 13
 - SQL Statements, 161
 - State table, 208
 - Statistical Reports menu, 132
 - Status Sequences menu option, 134
 - statuses
 - updating, 260
 - Statuses menu option, 139
 - studctc script, 165
 - Student Services menu option, 122

- Student Visit Agenda menu option, 130
- Students - Scheduled Visits menu option, 129
- Students by Church menu option, 131
- Students by School menu option, 131
- Students by Sex and Age menu option, 137
- Students Not Registered menu option, 138
- Suffix table, 210
- Summary by Schools menu option, 136
- syntax
 - schema names, 17

T

- Table Maintenance Menu, 151
- tables
 - Call Entry Result table, 19
 - columns, 13
 - common, 14
 - Decision table, 20
 - definition, 13
 - Enrollment Sequence table, 20
 - Enrollment Status table, 21
 - field descriptions, 17
 - in Recruiting/Admissions, 18
 - Lead Country table, 22
 - Lead Denomination table, 22
 - Lead Ethnic table, 22
 - Lead Major table, 22
 - Referral table, 23
 - required, 15
 - rows, 13
 - sequence in implementation, 196
- tables and records
 - used by Electronic Applications, 83
- Temporary Accomplishment record, 23
- Temporary Admissions record, 24
- Temporary Alternate Address record, 24
- Temporary Application record, 24
- Temporary Education record, 24
- Temporary Identification record, 25
- Temporary Interest record, 25
- Temporary Involvement record, 25
- Temporary Profile record, 25
- temporary records
 - in elecapp, 83
- Temporary Site record, 26
- terminology
 - additional information, 2
- Text Blob for Lead Contact record, 26
- TICK_ADMG, 213
- TICK_LEAD, 43
- Tickler menu, 125

- Tickler Review program, 223
- Tickler Structure Report menu option, 125, 142
- Tickler System Entry menu option, 125
- Tickler table, 205
- Title table, 210
- tpconv script, 167
- tpconvert
 - output, 241, 250
- transact.ph script, 81

U

- unavailable features. See product differences
- UNDG_DOC_CTGRY_DEF, 39
- UNIX
 - table names, 17
- unlocking
 - Admissions records, 261
- unregistered students, 260
- updatable sequences
 - in Admissions Statistics, 69
- Update Age of Students menu option, 138
- Update to Final Status menu option, 138
- Update Waitlist by Major menu option, 122
- Update Waitlist menu option, 122
- updating
 - Lead records, 268
 - statuses, 260
- updcnslr script, 165
- User Defined Profile Info menu option, 130

V

- validate.ph script, 81
- verifying
 - Lead records, 267
 - Web Application setup, 256
- Visa table, 210
- voiding
 - process in Admissions Statistics, 69

W

- Web server
 - setup, 256
- Weekly Stats by Caller menu option, 147
- Weekly Stats by Contact menu option, 147
- WordPerfect
 - in producing letters, 233

Z

- Zip Code table, 208