Jenzabar CX

Student

Reference Guide

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: systudt
Distribution date: 07/27/1999

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.
STUDENT REFERENCE GUIDE

TABLE OF CONTENTS

Registrar - How to Set Up PERFORM Screens

Overview

A description of all the fields in the 'catalog' and 'schedule' PERFORM Data Display/Entry screens.

Catalog/Schedule PERFORM Screens

The following pages describe the various data entry fields for the course, schedule and meeting records, listing required fields and optional fields, as well as possible values for certain data fields.

```
         Data Entry of Courses, Sections, Meetings for Catalog/Schedule
********************************************************************************
*                                                                              *
*                               Course Schedule                                *
*Number....[ACCT157 ]   ------------ Title -------------   Ind Study Allowed.N*
*Catalog...[UG84]      [Financial Accounting            ]  Dir Study Allowed.N*
*Program...[UNDG]      [                                ]  Fac Consent Req...N*
*Dept......[ACCT]      [                                ]  Repeatable........N*
*                 Abbrev Title...[                      ]  Times Repeatable..0*
*Division.. BUS            Sessions Offered..[FALL]                           *
*CIP.......[60.00  ]       Years Offered.....[1983]        Course Unit....    *
*Guideline.[N   ]        Min Hours..[4.00]  Grading..[LT]  Tuition Code...    *
*Level..[01] Size..[0  ] Max Hours..[4.00]  Counting.[E ]  Fee Code.......    *
*============================================================================*
*Section....[B ] Restriction..[UNDG] Title.....[                           ]*
*Session..[FA  ] Tuition Code.[    ] Fac..[    36] Clark, Ken  Exam Period    *
*Year.....[1984] Fee Code.....[    ] Grading...[LT]  Max Reg...[ 55]  Reg...20*
*Subsess....[  ] Hours.......[ 3.00] Method....[CL]  Max Wait..[  0]  Wait.. 0*
*Unit.....[    ]                                                              *
*============================================================================*
*Campus.....[MAIN]  Period Code..[A1]    Type..[L]  Faculty...[    26]Clark, K*
*Building...[ZIN ]  Begin Time...[1000]              S  M  T  W  T  F  S     *
*Room.......[ 300]  End Time.....[1130]       Days...[N][N][N][Y][N][Y][N]    *
*                                                                              *
********************************************************************************
```

Description of Course Data Fields

The following fields are from the course record.

*******Required Fields**
- Number, Catalog, Program, Dept
- {Hit the execute key (F1, PF1) to execute the course record}
- {Hit a 'd' for Detail to go from the Course File to the Section file}

Allowed
- Independent Study Allowed for this course, Y or N?
- The default for this field is "N"
- REGIST will check this value and display a warning message if a student attempts to take a course for independent study and the value in the course record is 'N'

Catalog
- Course Catalog working with maximum size of field is 4 characters. Most often will be 2 characters for school and 2 digits for the year.
- This value will become a table lookup for those schools using multiple catalogs.
- Example : MC83

CIP
- Classification Instructional Program number
- Optional 7 character field
- Use numbers provided by CIP or HEGIS; do not use decimal points

Counting Flag

Does this course count towards GPA? Is it earned (E) or not earned (NE). This value is verified from a cntg_table lookup Ind Study = Is

Dept

- Department offering course 3 to 4 characters, usually the same 3 to 4 characters preceding the digits in the course number
- Maximum size of field is 4 characters
- Must exist in dept_table because of the table lookup which occurs

Dir Study Allowed

- Is Directed Study Allowed for this course, Y or N?
- The default for this field is "N"
- REGIST will check this value and display a warning message if a student attempts to take a course for directed study and the value in the course record is 'N'.

Division

- Lookup from dept_table for the division
- Not an item for data entry

Faculty Con Req

- Is Faculty Consent Required, Y or N?
- The default for this field is "N"
- REGIST will display a message at the bottom of the student on-line registration screen is a student is attempting to register for a course requiring faculty consent.

Fee Code

- Course fee code for all sections of this course.
- Code is read by BILLING in assessing student charges.

Grading Flag

- Most common method of grading for course LT=letter, AU=audit, PF=pass/fail, SU=satisfactory/unsatisfactory
- This value is verified from a grdg_table lookup

Guideline

Core Guideline requirement code

Level

- Course Level, important for degree auditing
- Use 0 for entry level, preliminary courses; 1 for 100 level, 2 for 200 level, 3 for 300 level, etc. Use GR or 5 for graduate level
- Maximum field size is 2 characters

Max Hours

- Maximum number of credit hours
- (Neither one is for units. If 1 unit translates to 4 hours, then enter 4)
- Will default to 4 for those schools on a course or unit system and 3 for those schools on a credit hour system. The program REGIST uses these values as minimum and maximum ranges when the operator wishes to change the number of hours a student might be registering for.

Min Hours

- Minimum number of credit hours
- Will default to 4 for those schools on a course or unit system and 3 for those schools on a credit hour system

Number

- Course Number of 3 to 4 characters for department and 3 to 4 digits
- Maximum size of field is 8 characters
- Example : ART100 or HIST225; can use spaces, but must remember them in data entry

Program

- Program offering course selection
- Maximum size of field is 4 characters
- Example : UNDG
- This value is a table lookup from the prog_table

Repeatable

- Can this course be taken repeatedly, Y or N?
- The default value for this field is "Y"

Sessions Offered

Session codes to be used with degree auditing especially when offering the course only in Fall or Spring.

Size

Size of course, for planning purposes only, optional. Can use largest section size or total for all sections

Times Repeatable

- Maximum Number of times this course can be repeated.
- If Repeatable = Y, O Times Repeatable is Infinite.

Title

3 lines of 32 characters each Free form

Tuition Code

- Tuition code for associated with this course that is read by the Student Billing program
- If this field is left blank, the student will be billed by student program rather than course.

Unit

- Unit code if school is using units
- The field can be noupdate, noentry. If the field is being used, it must be in unit_table

Years Offered

Odd or Even Years, Specific Years

Description of Section Data Fields

The following are section record data fields.

*******Required Fields**

- Course number, course catalog, section number, section session, section year, section hours
- *** The faculty id number should also be entered
  - {Hit the execute key to execute the section record}
  - {Hit an 'm' for Master to return to the Course File if there is not a Meeting Record to add as a detail to the Section}
  - {Hit a 'd' for Detail to move from the Section file to the Meeting file}
- Description of Meeting Data Fields

*******Required Fields**

Campus, Building, Room, Course Number, Course Catalog, Session, Year, Section Number

- {Hit the execute key to execute the meeting record}
- {Hit an 'm' to move from the Meeting File to its Master Section File}
- {Hit another 'm' to move from the Section File to its Master Course File before entering the next course, section}
- {To return directly to the course file, hit '1f' in sequence}

Begin Time

- Time meeting/section begins in 24 hour clock
- Example : 1 p.m. is 1300

Building

- Building for meeting
- Must be in facility_table
- Maximum field size is 4 characters
- This is a table lookup field

Campus

- Campus location for meeting
- Must be in facility_table
- Most often will use MAIN for the main area of campus
- Maximum field size is 4 characters
- This is a table lookup field

End Time

- Time meeting/section ends in 24 hour clock
- Example : 6 p.m. is 1800

Exam Period

- Final Exam period code
- Maximum field size is 4 characters

Faculty

- Id number of faculty member teaching section will lookup abbreviated name of faculty member and display it.
- Make sure this value is entered so that statistics can be done for this faculty member

Faculty

- Faculty Id for faculty member teaching section
- Usually the same, but may be different than the Fac Id for Section Will lookup Faculty Abbreviated Name and show it on the screen beside the Id number.
- This field should be entered even if the value was entered at the section level since the faculty name on schedules corresponds to this field

Fee Code

The course fee code that applies to particular section(s) and not all sections of the course. A lookup from the assessment table.

Grading

- Grading Flag
- Defaults to LT for letter
- See Comments above under course about Grading Flag can be the same code or different code from that of the course table lookup occurs for this field

Hours

- Number of hours that the section will be offered

- A lab might be 0.0 hours while a lecture section for the same course might be 4.0
- This value is a must for the regist program

Max Reg

- Maximum number of students who can register for the section without being placed on a wait list
- Default is 99

Max Wait

- Maximum number of students who can be placed on the wait list for the section
- Default is 0
- This is a noupdate, noentry field if the institution is not using wait lists.

Method

- Method of Instruction or how the course was taken must be one of the codes from the method_table
  - CL=classroom
  - DS=directed study
  - IS=independent study
  - WV=waiver
  - EX=exam
- Table lookups will occur for this field

Prd

- Period Code
- Maximum field size is 2 characters
- This is a table lookup field
- Some schools may request that entry of this field lookup the Begin and End times for the meeting record. In that case, the catalog screen will be used for the lookups and INFORMER update used to place the correct times and days in the meeting record that correspond to the period code. The schedule screen will be used to enter meeting records for which there are no period codes.

Reg

Lookup for number of students registered for the section

Restriction

- Group of students section is restricted to
- May be program such as UNDG for UNDG students only or SEM for Seminary Students only
- This field might also be the program code for the section to be used with student billing

Room

- Room number within building
- Maximum field size is 4 characters
- This is a table lookup field
- If room numbers were entered in the facility table as right justified,
- this field should contain the 'right' option.
- *****The Campus, Building, Room combination must be in the facility table

S M T W T F S

- Days of the week
- Place Y or N under the day
- Defaults to N
- Y means that section meets on that day

Section

- Section Number or code
- Maximum Field size is 2 characters
- Will right justify if only one number is entered and if the school is only using numeric section codes

Section Title

- Title of section if there is a subtitle from that of the course might use with Special Topics Courses with the Course title being "Special Topics" and a Section title being"History of Lit"
- This is an optional 32 character field

Session

- Session code for the Session in which the section is offered
- Maximum Field Size is 4 characters
- Example : FA for Fall, SP for Spring, WI for Winter, etc
- A table lookup occurs for this field

Subsess

- Subsession code if section is offered during a subsession of the main session
- Might use for courses such as Physical Education that are offered for a set number of weeks within the main session
- Maximum field size is 2 characters
- If this field is used, a table lookup will occur, else the field is a noupdate, noentry field
- Recommend that subsessions be used for multi/overlapping summer sessions

Tuition Code

The tuition code for the section, which may differ from the Course tuition code. A lookup from the assessment table.

Type

Type of meeting

- L=lab
- P=Problem Session
- C=Classroom lecture

Unit

Unit code if school is using units. See note under course record for Unit

Wait

Lookup for number of students on wait list for section

Year

- Academic Year Section Offered
- Maximum Field Size is 4
- Example : 1983 for the 1983-84 school year

Registrar - How to Set Up Tables

Overview
Sequential procedures for implementing the registrar's area of the CARS Solution-Unix system include table and file data entry as well as registering students in classes and maintaining student records. This document covers information regarding the building tables needed.

Introduction
Certain files and tables must be created prior to entering course catalog and schedule data.

The first grouping of tables needed for course catalog data entry will be created by Jenzabar, Inc. with the assistance of the Registrar's Office. These tables should only be changed after discussing it with Jenzabar, Inc. since several source programs (REGIST, GRADING, TRANS) utilize the table values. However, additional fields may be added to the table, such as the code used on a prior system, but these fields will not be read by any of the application programs.

Program Table   Program Reg Table
Counting, Grading, Method, Repeat Flags
Valid Grade Table

This document contains the table names, field names and explanations, data entry screens and reports generated for the tables. New to each field name there is a number in brackets, that number appears in the associated data entry field on the example screen and under its report heading on the example reports.

Program Table

Overview
The purpose of the program table (prog_table) is to define the various programs a student may enroll in at the institution. Ideally there is one program for each transcript a student will have. Eg. "UNDG" for undergraduate, "GRAD" for graduate, etc.

The number of academic programs should be kept to a minimum if at all possible since REGIST tests for both the student program and the course program for valid registrations.

Fields
_og_tbcode' [1] is a unique four character code defining the program. Eg. UNDG = undergraduate, GRAD = graduate, etc.

_og_text' [2] is the description that is an expanded explanation of the four character code.

_og_sort' [3] is a field that is used for reporting. The user can specify the order that programs will print on reports. This is used instead of sorting the programs alphabetically.

_og_gid' [4] field is the group id number, as found in the /etc/groups file, for the registrar who will be registering the students.

_og_unit_fctr' [5] should be the number of course hours equal to one course unit. Since that is the value multiplied by the units to equal the credit hours per unit. If the school is on the credit hour system, the prog_unit_fctr should equal "1". A school where one unit is equivalent to 4 course hours, should have a value of '4' in the prog_unit_fctr field. This value may vary for schools on the unit or course system where one unit is not equal to four credit hours.

_og_adm_tick' [6] is the tickler code that the Admissions Office uses for this program.

_og_sch_code' [7] is the four character school abbreviation. If a school is using the unit or course system this code is required.

_rog_trans_frm_ofcl' [8] is the default transcript official form code for this program.

_rog_trans_frm_unofcl' [9] is the default transcript unofficial form code for this program.

There is a second screen for the program table. This screen is used to display information to be used in defaulting database field values and for validity checking of entered data. None of the fields on this screen can be updated.

_og_grd_scheme' [0] is not currently used, so it can not be updated. The value for the field will be defaulted to '1'.

The other two fields, [a] and [b], on the perform screen are not fields in the program table, but are actually values that have been found by looking up the transcript form codes for official and unofficial codes in the Transcript Form Table. The values in these fields correspond to the value of the ttransfrm_ofcl field for each transcript form code entered into the prog_table.

```
Data Entry Screen:
                                Program Table
  Program Code....[1   ][2                      ]
  Sorting order...[3   ]      Group ID...........[5   ]  Transcript Form Codes:
  Unit Factor.....[4     ]    Admissions Tickler..[6   ]     Official.....[8   ]
                              School Code.........[7   ]     Unofficial...[9   ]
```

```
Second data entry screen (only accessed during queries):
                          Program Table (continued)
Program Grade Scheme......................[0]
Flag for Official Transcript Form Code....[a]
Flag for Unofficial Transcript Form Code..[b]
```

Program Registration Table

Overview

The program registration table (prog_reg_table) is used by REGIST to determine whether or not a specific registrar has the ability to register students of a specific program. If more than one registrar office exists within the institution's structure, the values for updating and displaying are very important. The program registration table also contains the tuition code for the student/course program combination. The tuition code should be one of the tuition assessment codes found in the assessment table.

Fields

_og_stu' [1] will be the registering students program from the program table.

_og_crs' [2] will be the program that has the courses the student may register for.

_og_cntg' [3] is the default value placed in the students record at time of registration.

_og_upd' [4] is on of the following codes (S)tudent registrar, (C)ourse registrar, or (B)oth. This code allows the appropriate registrar update permissions on student records. To have update permissions they also need display permissions as set by prog_dspl.

_og_dspl' [5] uses the same code as prog_upd, S, C, or B -- Do not give update without display permission. These give display permission of student records to the appropriate registrar.

_og_tuit_code' [6] this is a 4 character that can be found in the assessment table.

The typical undergraduate program would contain the following values for its entry in the table.

Prog_stu	prog_crs	prog_cntg	prog_upd	prog_dspl	prog_tuit_code
UNDG	UNDG	E	BOTH	BOTH	TUND

Additional programs such as Continuing Education or Seminary would have entries with some variation depending upon who registers such students and which course program applies to them. The counting value for a program would depend upon whether the courses count toward graduation or not.

Prog_stu	prog_crs	prog_cntg	prog_upd	prog_dspl	prog_tuit_code
SEM	UNDG	NE	COURSE	BOTH	TSEM

In the above example, a Seminary student (SEM prog) taking undergraduate courses (UNDG prog) might have to process the registration through the undergraduate (UNDG prog) registrar's office, but the Seminary registrar (SEM prog) would like display capabilities. The Seminary student may still be assessed tuition at the Seminary rate rather than undergraduate rate, but the courses will not count towards graduation.

Data Entry Screen:

```
                          Program Reg Table
Students enrolled in program...[1   ] May take courses from program..[2   ]
For this combination:
  Counting flag................[3 ]
  Which registrar may update...[4]
  Which registrar may display..[5]
Which tuition code applies towards the above program combination?
                [6   ]
```

Report

```
                             (RG) Program Reg Codes                  tprogreg
            Stu     Crs    Cntg    Update     Display  Tuition
            [1]     [2]    [3]      [4]         [5]      [6]
            ----    ----   ----    -------    -------  -------
            CEU     CEU     NE      Both       Both
            GRAD    UNDG    E       Both       Both
            GRAD    GRAD    E       Both       Both
            POST    UNDG    E       Both       Both
            UNDG    UNDG    E       Both       Both
```

Counting Table

Overview
The two counting flags are "E" for earned and "NE" for not earned.

Fields
Àtg_tbcode' [1] is a unique 2 character code used to differentiate between the different counting flags.

Àtg_text' [2] is a explanation of the 2 character code.

Àtg_att_fctr' [4] is used to determine if hours registered are counted in the attempted hours. '1' will include any courses taken with this value in attempted hours, '0' will exclude them.

Àtg_earn_fctr' [5] is used to determine if hours registered are counted in the earned hours. '1' will include any courses taken with this value in earned hours, '0' will exclude them.

Àtg_pass_fctr' [6] is used to determine if hours registered are counted in the pass hours. '1' will include any courses taken with this value in pass hours, '0' will exclude them.

Àtg_qual_fctr' [7] is used to determine if hours registered are counted when calculating gpa. '1' will include any courses taken with this value in quality hours, '0' will exclude them.

Àtg_print' [3] is either a 1 or 0. If this value is set to '1' the code will print on the transcript.

The table should contain the values as follows:

cntg_tbcode	cntg_att_fctr	cntg_earn_fctr	cntg_pass_fctr	cntg_qual_fctr
E	1	1	1	1
NE	0	0	0	0

Data Entry Screen:

```
            Code  Text                     Print  Att  Earn  Pass  Qual
========================================================================
Counting....[1 ] [2                       ] [3]   [4]   [5]   [6]   [7]
========================================================================
```

Report

```
                              (RG) Counting Codes                      tcntg
                                Attempt  Earned    Pass   Quality
Code         Text               Factor   Factor   Factor   Factor   Print
[1]          [2]                  [4]      [5]      [6]      [7]     [3]
---- ------------------------   ------   ------   ------   ------   -----
                                   0        0        0        0       0
 E   Earned                        1        1        1        1       0
 NE  Not Earned                    0        0        0        0       0
```

Method Table

Overview
The method table (meth_table) contains the codes for the methods in which a student can take a class. For instance, "CL" is the default value for most classes since they are taken in the form of classroom instruction. Other values are "WV" for waiver, "EX" for exam, "DS" for directed study, and "IS" for independent study.

Fields

_th_tbcode' [1] is a unique 2 character code used to determine the instruction method for this course, section or registered student.

_th_text' [2] is the text to describe method of instruction.

Data Entry Screen:

```
            Code  Text
========================================================================
Method......[1 ] [2                       ]
========================================================================
```

Report

```
                       (RG) Method Flag Codes                      tmeth
              Code              Text
              [1]               [2]
              ----    ------------------------
               CL     Classroom
               DS     Directed study
               EX     Exam
               IS     Independent study
               WV     Waiver
```

Other Tables

Several tables and files may be created by the CARS coordinator with the assistance of the Registrar's Office without the aid of Jenzabar, Inc., but are needed before any further registration data entry. The following tables can be created in any order.

1.	Session Table	Maintained by the Registrar's Office
2.	Subsession Table	Maintained by the Registrar's Office
3.	Academic Divisions and Department Tables Maintained by the Registrar's Office	
4.	Building Table	
5.	Facility Table	(campus, buildings, classrooms, seating capacity). Also used by Student Services, Facility Management
6.	Faculty File	(5 lines of ID data plus fac_abbr_name). Also used by the Academic Dean, Personnel, Payroll
7.	Period Table	(if periods are used). Maintained by the Registrar's Office
8.	Section Requirements Table	Maintained by the Registrar's Office

Session Table

Overview

The session table (sess_table) should be completed in full, especially the sort values so that the transcript will sort the sessions of an academic year in the order in which those sessions occur during an academic school year. Every session entered must also have an entry in the Academic Calendar file with the corresponding dates for registration, add/drops, etc..

Fields

_ss_tbcode' [1] is a unique 4 character code for each session the run by the school, eg FA,WI,SP,S1.

_ss_text' [2] is a 24 character description for the session.

_ss_yr_offset' [3] is the order in which the session falls in calendar year; eg WI=0,SP=1,S1=2

_ss_ord' [4] is the order that session falls in academic calendar year; eg FA = 0

_ss_crs_tuit' [5] does course tuition code override program tuition code? Y or N

Data Entry Screen:

```
                               Academic Sessions
Code..[1   ][2                      ]         Offset......[3   ]
Order.[4      ]                               Crs Tuition.[5]
```

Report

```
                                (RG) Session Codes                      tsess
                                            Calendar  Academic
                                              Year     Year     Crs
          Code              Text             Offset    Order    Tuit
          [1]               [2]               [3]       [4]     [5]
          ----    ------------------------  ------    -----    ----
          FA      Fall                         7        1       N
          IT      Inter-term                   8        2       N
          WI      Winter                       1        3       N
          SP      Spring                       2        4       N
          MAY     May Mini Term                3        5       N
          SM      Summer School                4        6       N
          SM1     Summer Billing Sess 1        5        7       N
          SM2     Summer billing sess 2        6        8       N
```

Subsession Table

Overview
Entries in the subsession table are necessary if subdivisions of sessions are used for registration and/or billing.

Fields
_ubsess_tbcode' [1] is a unique 2 character code.

_ubsess_text' [2] is a 24 character explanation of the 2 character code.

Data Entry Screen:

```
================================================================================
                               Subsessions
              Code...[1 ][2                      ]
================================================================================
```

Report

```
                               (RG) Subsession Codes                 tsubsess
             Code              Text                  Session
             [1]               [2]
             ----       ------------------------     -------------
              SM        This is a test file          Fall
              M1        Graduate Summer Module 1     Summer School
              M2        Graduate Summer Module 2     Summer School
              M3        Graduate Summer Module 3     Summer School
              M4        Graduate Summer Module 4     Summer School
              M5        Graduate Summer Module 5     Summer School
              S1        UG Summer Session I          Summer School
              S2        UG Summer Session II         Summer School
```

Academic Calendar Table

Overview
The academic calendar record (acad_cal_rec) contains the registration information about a session offered at the institution. New session data should be added to the academic calendar record before entering course data.

This file includes the beginning and ending dates for the Academic Session as well as dates for Add/Drops, Withdrawals, etc. The academic calendar record will be used by REGIST and BILLING to test drop dates against refunds and grades. The session (and subsession) table(s) must be created before the academic calendar record. If the academic calendar is not built, then student's can not be registered for the particular session, year, and program combination.

Fields
_sess' [1] enter the session code for this academic session.

_yr' [2] enter the academic year.

_subsess' [3] enter subsession code that corresponds to the prog code, -- if none.

_prog' [4] enter program code that corresponds to this record.

_beg_date' [5] enter official beginning date of session.

_end_date' [6] enter official ending date of session.

_last_add_date' [7] enter last date courses can be added to registrations for session.

_last_drop_date' [8] enter last date courses can be dropped without getting 'W' grade.

_last_wd_date' [9] enter last date courses can be withdrawn without 'WF' grade.

_first_reg_date' [0] enter first date students can register for this session.

_last_reg_date' [a] enter last date registrations to be accessed this session.

Data Entry Screen:

```
                      Academic Session Calendar Table
Session.....[1   ]     Year...[2   ]    Subsession.[3 ]     Program.[4   ]
Begin/End...[5       ]-[6       ]
Last Add....[7       ]      Last Drop.....[8       ]   Last Withdraw..[9       ]
1st Register[0       ]      Last Register.[a       ]
```

Report

```
                               ACADEMIC CALENDARS                 acadcal
                             Academic Program: UNDG
                                               [4]
================================================================================
Sess Year Acyr Su Beg Date End Date First Rg Last Add Last Drp Last Wth Last Reg
[1]  [2]      [3]   [5]      [6]      [0]      [7]     [8]       [9]      [a]
--------------------------------------------------------------------------------
FA   1985         08/28/85 12/30/86 08/26/85 12/15/86 12/15/86 12/25/85 12/01/86
SP   1985         01/15/85 11/01/86 11/01/84 11/01/86 11/01/86 11/01/86 11/01/86
SU   1985         07/01/85 09/15/85 07/01/85 09/15/85 08/15/85 09/15/85 09/15/85
SU   1985      S1 07/01/85 08/15/85 07/01/85 07/15/85 07/15/85 08/01/85 08/15/85
SU   1985      S2 08/16/85 09/15/85 08/16/85 08/31/85 09/05/85 09/10/85 09/15/85
FA   1986         08/28/86 12/19/86 08/26/86 04/01/87 04/01/87 04/01/87 01/01/88
SP   1986         01/01/86 01/01/89 01/01/86 01/01/89 01/01/89 01/01/89 01/01/89
FA   1987         08/28/87 12/19/89 02/20/86 09/15/89 10/01/89 11/01/89 01/01/89
SP   1987         01/01/87 06/30/87 01/01/87 05/01/87 05/01/87 05/01/87 07/30/87
SU   1987         03/01/86 11/30/87 03/01/86 11/30/87 11/30/87 11/30/87 11/30/87
SU   1987      S1 03/01/86 04/01/86 03/01/86 04/01/86 04/01/86 04/01/86 04/01/86
SU   1987      S2 03/01/86 04/01/86 03/01/86 04/01/86 04/01/86 04/01/86 04/01/86
SP   1988 8788    01/10/88 05/15/88 01/01/88 01/10/88 01/15/88 01/30/88 05/05/88
```

Building Table

Overview

Fields

dg_campus' [1] enter 4 character code for campus (eg MAIN).

dg_tbcode' [2] enter 4 character code for building (eg GOTH).

dg_text' [3] enter 24 character description for building (eg Gotham Hall).

Data Entry Screen:

```
                           Building Table
   Campus..[1   ]     Building..[2   ] [3                       ]
```

Report

```
                           Building Codes                        tbldg
               Campus        Code              Text
                [1]          [2]               [3]
                ----         ----    -------------------------
                             DORM     Main Dormitory
                MAIN         AMH      Aaron Memorial Hall
                MAIN         ASC      Academic Science Bldg
                MAIN         AH       Alexander Hall Dorm
                MAIN         AC       Art Center
                MAIN         BURK     Burke Administration
                MAIN         CML      Campbell Mem. Library
                MAIN         CSC      Campus Student Center
                MAIN         CHAL     Chalfant Auditorium
                MAIN         LUDW     Ludwig Student Center
                MAIN         DORM     Main Dormitory
                MAIN         MILL     Miller Business Center
                MAIN         REED     Reed Hall of Science
                MAIN         WISN     Wisner Nursing Center
```

Facility Table

The facility table (facil_table) should contain a value for every classroom, dorm room, etc. on the system with the fields of campus, building, and room all entered for each room. If room numbers are entered right justified, then the catalog and schedule screens under "$(CARSPATH)/regist/screens" should have the mtg_room value entered with the 'right' option so that the values will match in data entry lookups.

Fields

cil_campus' [1] campus code.

cil_bldg' [2] code for building, must be in building table.

cil_room' [3] room number or code".

cil_max_occ' [4] maximum allowed in room.

cil_desc' [5] description of room usage.

cil_phone' [6] enter phone number if one is available for this room.

cil_type' [7] 'C' for classroom, 'D' for dorm, 'O' for Office". The data entry operator should also enter type 'C' for the type of room indicating that the room is a classroom.

Data Entry Screen:

```
============================================================================
                               Facility Table
     Campus..[1   ]  Bldg..[2   ]  Room..[3   ]  Max Occupants..[4    ]
           Facility Description..[5                      ]
     Phone..[6       ]                           Type of facility..[7]
============================================================================
```

Report

```
                                 Facility Codes                       tfacil
                                               Max
Campus  Bldg  Room        Description         Occup  Telephone Type
 [1]    [2]   [3]             [5]              [4]      [6]    [7]
------  ----  ----  ------------------------  -----  --------- ----
        COMM         Commuter
 MAIN   AC           Art Center
 MAIN   AC    002    Storage
 MAIN   AC    003    Storage
 MAIN   AC    005    Storage
 MAIN   AC    007    Storage
 MAIN   AC    101    Faculty Office
 MAIN   AC    102    Storage
 MAIN   AC    103    Gallery
 MAIN   AC    104    Faculty Office                   563-4233   O
 MAIN   AC    105    Storage                                     S
 MAIN   AC    201    Storage                                     S
 MAIN   AC    202    Darkroom
 MAIN   AC    205    Faculty Office                   563-4521   O
 MAIN   AC    208    Faculty Office                   563-4528   O
 MAIN   AC    AL     Art Lab
 MAIN   AH    146    Dorm Room
 MAIN   AH    147    Dorm Room
 MAIN   ASC   009    Lab - Aux
 MAIN   ASC   010    Lab - Aux
 MAIN   ASC   011    Lab - Herbarium
 MAIN   ASC   012    Lab - Aux
 MAIN   ASC   013    Greenhouse
 MAIN   ASC   014    Lab - Advanced Botany
 MAIN   ASC   015    Lab - General Botany
 MAIN   ASC   016    Lab - Aux
 MAIN   ASC   017    Lab - Geology
 MAIN   ASC   018    Faculty Office
 MAIN   KHD   030    Dorm Room                    2
 MAIN   SK    201    Dorm Room                    2              D
 MAIN   SK    202    Dorm Room                    1              D
 MAIN   SK    203    Dorm Room                    3              D
```

Faculty Record

Overview
Each faculty record (fac_rec) entered should contain the values of fac_id, fac_abbr_name (the 10 character abbreviation of the faculty full name), fac_campus, fac_bldg, and fac_room so that correct values can be retrieved by different screens and reports. The id record is needed before any part of the faculty record can be created. Below is the screen that the Registrar's Office will have access to for queries and data entry of faculty data:

Fields
_no' [1] = "** Query On ID Fields Only **".

me' [2] = "** Query On ID Fields Only **".

c_title' [3] = "** Faculty Member's Title, Query Field Only **".

c_bldg' [4] = "** Office Building Of Faculty Office, Query Field Only **".

c_room' [5] "** Office Room Number Of Faculty Office, Query Field Only **".

c_abbr_name' [6] = "** Faculty Member's 10 character Name Abbreviation, Required **".

Data Entry Screen:

```
Id No...[1         ]                          Title.......
Name....[2                              ]  Social Sec..
Address.                                      Telephone...
City....                                      State...     Zip...
================================================================================
Title............[3                        ]         Office.......[4   ][5   ]
Abbrev Name..[6         ]
================================================================================
```

Division and Department Tables

Overview
An academic division may divided into several departments. For example, the division of Natural Science may include the departments of Physics, Biology, and Geology. In other instances, the division may be the same as the department as in the division of Mathematics and the Mathematics department. Because departments are considered below divisions in a hierarchical structure, it is recommended that the division table (div_table) be created first. Prior to creating the department table (dept_table), the business office and academic dean should be contacted since academic departments are also considered financial cost centers and all should be in agreement as to the "official" departments. Courses are offered per department. Enrollment reporting and grade distribution statistics also depend on the department breakdowns.

Fields
v_tbcode' [1] code for academic division, eg SCI for Science, LA for Language Arts.

v_text' [2] text to describe code.

pt_tbcode' [3] code for academic departments, cost centers.

pt_text' [4] textual description of code.

pt_div' [5] enter division code department part of from division table.

Data Entry Screen:

```
                            Division Entry
         Division Code...[1   ]    Description...[2                       ]
================================================================================
                           Department Entry
         Department Code...[3   ]   Description..[4                       ]
                          Division.....[5   ]
```

Report

```
              (RG) Division Codes                           tdiv
        Code              Text
        [1]               [2]
        ----    ------------------------
        ACCT    Business / Accounting
        ARTS    Fine Arts
        EDUC    Education
        LLIT    Language and Literature
        NSCI    Natural Science
        SSCI    Social Science
================================================================================
                   (RG) Department Codes                    tdept
        Code              Text              Division
        [3]               [4]                 [5]
        ----    ------------------------    --------
        ART     Art                           ARTS
        BAE     Business Admin/Economics      SSCI
        BIO     Biology                       NSCI
        BUSA    Business/Economics            ACCT
        CHE     Chemistry                     NSCI
        COA     Communication Arts            ARTS
        EDU     Education                     EDUC
        ENG     English                       LLIT
        HIS     History                       SSCI
        HOE     Home Economics                EDUC
        HPR     Health, Phys. Ed. & Rec.      EDUC
        LAN     Foreign Languages             LLIT
        MPC     Math/Physics/Comp Sci         NSCI
        MUS     Music                         ARTS
        PHI     Philosophy                    SSCI
        POS     Political Science             SSCI
        PSY     Psychology                    SSCI
        REL     Religion                      SSCI
        SOC     Sociology                     SSCI
```

Period Table

Overview

If the institution classes meet at given time, periods you can use the period table.

Fields

_d_tbcode' [1] 2 character period code to be used by meeting record.

_d_yr' [2] year period code used.

_d_sess' [3] session code for session period code used.

_d_beg_time' [4] beginning time in 24 hour clock, eg. 1400.

_d_end_time' [5] ending time in 24 hour clock, eg. 2000.

_d_days[1,1]' [6] enter Y if period meets this day.

_d_days[2,2]' [7] enter Y if period meets this day.

_d_days[3,3]' [8] enter Y if period meets this day.

_d_days[4,4]' [9] enter Y if period meets this day.

_d_days[5,5]' [0] enter Y if period meets this day.

_d_days[6,6]' [a] enter Y if period meets this day.

_d_days[7,7]' [b] enter y if period meets this day.

Data Entry Screen:

```
      Periods With Times And Days Of Week
Code          Year       Session          Time
[1 ]         [2   ]       [3   ]     [4   ] - [5   ]
           S   M   T   W   T   F   S
          [6] [7] [8] [9] [0] [a] [b]
```

Report

```
                   (RG) Period Codes                       tprd
                        Beg     End       Days
Code    Year    Sess    Time    Time    SMTWRFS
[1]     [2]     [3]     [4]     [5]    [67890ab]
----    ----    ----    ----    ----    -------
 1      1985    FA                      NNNNNNN
 3      1985    FA       800     850    NYYYYNN
 3      1987    FA       900     950    YNNNNYY
 M1     1988    FA       800     850    NYNYNYN
 M2     1988    FA       900     950    NYNYNYN
 M3     1988    FA      1000    1050    NYNYNYN
 M4     1988    FA      1100    1150    NYNYNYN
 M5     1988    FA      1230    1320    NYNYNYN
 M6     1988    FA      1430    1520    NYNYNYN
 M7     1988    FA      1530    1620    NYNYNYN
 M8     1988    FA      1630    1720    NYNYNYN
 T1     1988    FA       800     920    NNYNYNN
 T2     1988    FA       930    1050    NNYNYNN
 T3     1988    FA      1100    1220    NNYNYNN
 T4     1988    FA      1245    1340    NNYNYNN
 T5     1988    FA      1345    1505    NNYNYNN
 T6     1988    FA      1515    1635    NNYNYNN
 T7     1988    FA      1730    1850    NNYNYNN
```

Section Requirements Table

Overview
If various sections of courses are restricted for special purposes, such as a section for majors only, the section requirements may be displayed at registration time.

Fields

_ecreq_tbcode' [1] 4 character code to be used by section record.

_ecreq_text' [2] description of the section requirement.
_ecreq_warn' [3] enter Y if warning should appear in REGIST.

Data Entry Screen:

```
Course Section Requirements Table
---------------------------------
Code....................[1   ]
Text....................[2                       ]
Registration Warning....[3]
```

Report

```
     (RG) Section Requirement Codes
                                    Display
Code              Text              Warning?
[1]               [2]                 [3]
----    ------------------------    --------
        No requirement                No
MAJ     Majors only                   Yes
```

Other Codes

Several codes that are not table driven must be established next. These include Meeting types, C=Classroom, L=Lab, P=Problem Session, used in the meeting record (mtg_rec) and Final Exam Periods used in the sec_rec the screen. Otherwise, the operator will have to enter each value accordingly.

Faculty id is asked for again and should be entered. This field is included in the meeting record as well as in the section record to include instances when courses are team taught or if a different faculty members are responsible for different meeting types; i.e. lab, problem session, classroom.

Those fields not used by the institution will be marked as noupdate, noentry fields, and those usually containing the same value for each record created will contain default values, which can be overwritten when needed. (See documentation on Course Catalogs/Schedules.)

Synopsis of Course Table and File Data Entry

This diagram illustrates the order and grouping of table and file data entry with regard to course scheduling.

```
Level                           Record
-----                           ------
1.                           Program Table
1.a.                       Program Reg Table
2.                           Course Flags
2.a.            Counting    Grading    Method    Repeat
3.                         Grade Type Table
3.a.                      Valid Grade Table
4.                          Session Table
4.a.                      Subsession Table
4.b.                   Academic Calendar File
5.                          Building Table
5.a.                        Facility Table
6.                          Faculty File
6.a.                      ID= name, id_no
6.b.                   Fac= fac_id, fac_abbr_name
7.                          Period Table
8.                    Section Requirements Table
9.                         Division Table
9.a.                      Department Table
10.           Additional Codes: Meeting types, exam periods
```

Some institutions may also wish to establish a table of core requirement codes. This table should also be created prior to creating or updating the course catalog(s).

Registrar - Maintaining Student Data Records

Introduction
The Registrar's Office has access to all biographical or academic records for students. The program enrollment and student academic records must be maintained as changes occur so that billing and reporting will be accurate. The records can be classified in three categories:

- Those required for registration
- Those used in the automatic billing process
- Those used for reporting

These types of records are described below.

Records for Registration
An ID record and a valid Program Enrollment record are required before a student can be registered for classes. Only certain fields are necessary.

Each student must be assigned an ID number by CARS Solution. You should first query on name and/or social security number to make sure that the individual is not already on the system before making any additions. The student's name, address, ID and social security number are the only fields used by Online Registration.

Several data fields of the Program Enrollment record must contain values before the student registration can occur. The first value is a program code (prog), a value in both the Program and Program Registration tables. A classification code (cl) is required. The Program Academic Status field (acst) should contain a value from the Academic Status table that allows a student to be registered. The Academic Status table defines the valid statuses for your institution and indicates if a particular group of individuals can register for classes. For example, if your Academic Status table entry for GRAD (graduates) contains an "N" in the acad_stat.reg field, then any students having an academic status of GRAD will be ineligible to register.

If the Degree Granted Date (deg_grant_date) field has a date value other than "00/00/0000," the program assumes that the student is a graduate. If the student attempts to register for classes, you will receive a warning message that the student holds a degree. You may override the warning, or disable the warning message by modifying the macro ENABLE_GRADMSG_DISP in $CARSPATH/macros/custom/student.

If students have a maximum number of hours for which they can register, the Restricted Schedule field should contain "Y." REGENT restricts the number of registered hours for the student using the Maximum Hours Registered (prog_enr_rec.max_hrs_reg) field.

The following diagram illustrates the relationships between files and tables on the system for the purpose of registration.

```
ID/Student                        Courses           Tables           Files
  Load                             Load             Check            Check
-----------                      ---------        ----------       --------
 ID                                CW               Denom            Crs
 Profile                           Reg              St               Cat
 Stu_serv                                           Ctry             Sec
 Prog_enr                                           Meth             Sess
 Stu_acad                                           Cntg             Subsess
 Sbcust                                             Rep
                                                    Grdg
                                                    Cl
                                                    Prog
                                                    Prog_reg
                                                    Doc
                                                    Major
```

```
                                                            Minor
                                                            Conc

                                                            Regperm
          -----                         -----               -----             -----
            |                             |                   |                 |
            |                             |                   |                 |
            V                             V                   V                 V
            V                             V                   V                 V

                               ----------------------------------
                               |                                |
                               |  REGENT (application program)  |
                               |                                |
                               ----------------------------------
            <---------------------|                     |----------------->
          -----                                                    -----
            |                                                        |
            |                                                        |
            V                                                        V
            V                                                        V
           SDS form                                                 Sec
                                                               sec_rec.wait_num
                                                               sec_rec.reg_num
                                      |      |
                       <--------------|      V
                |    |     |
                |    |     |
                V    V     V

               CW   CW    CW           Stu_acad

                |    |     |            reg_hrs
                V    V     V            reg_au_hrs
                                        wait_hrs
               Reg  Reg   Reg           reg_stat
                |    |     |            reg_upd
                V    V     V            no_adrp
               Reg  Reg   Reg           crs_fees
                                        prog_enr
                                        stu_serv
                                        sbcust
```

Records for Student Billing

The automatic billing program uses several current student files maintained by the Registrar's Office. They include id_rec, profile_rec, stu_serv_rec, and sbcust_rec, prog_enr_rec for each program in which the student participates, the stu_acad_rec for each session, year, program combination, and the hours, fee codes, session, subsession, and year values from cw_rec and reg_rec.

The student must have also an id number to receive bills or statements. The name, social security number, and address are needed so that the bill and/or statement can be mailed and so that the student is uniquely identified.

There are several fields in the Profile record from which a student may be billed. They include the following:

- sex
- marital status
- denomination
- resident county and/or state
- resident country

If a student is to be billed for out-of-state tuition, it is very important that the resident state (contained on the prog_enr_rec and stu_acad_rec) be maintained.

The data fields commonly used in assessing student charges from the Program Enrollment record include the following:

- student program
- classification
- major
- expected graduation date

Note:

Classification must be on the stu_acad_rec or the program will exit abnormally. Other enrollment data not maintained on a session by session basis, but needed for billing purposes may be added to this record.

Use the Student Academic record to maintain data that:

- varies from session to session
- is required for billing and is not already in one of the records discussed here

Data extracted from cw_rec and reg_rec provides billing information by registered hours, audit hours, total hours, course fee and course tuition codes. Add/drop dates are also important for refunds and late registration fees. This data is maintained through REGENT, so you should be particularly aware of the effective date entered as one of the initial parameters for a student. For example, if an add/drop form is submitted one day, but you are not able to enter it until a day or so later, the registration date should be back-dated. This way, if the student was entitled to a refund based on the day the form was submitted, the refund will be calculated correctly.

The Program Enrollment Screen

The following lists the fields in the Program Enrollment screen, the main screen used in maintaining and querying student data.

Academic Status/Date
Student's academic status; one of the codes from the Academic Status table; also, the date the academic status took effect.

Add Date
Date the ID record was added; query-only field.

Address
Student's permanent street address.

Admitted Sess/Yr.
Session, and Year student admitted to the institution; these values are placed in this record from the adm_rec with a script.

Admitted Status
Student's admitted status, one of the codes from the Academic Status table. Must be a code that allows registration for the student to be able to register for classes.

Advisor
ID number of student's advisor.

Birthdate
Date of student's birth, in the format mm/dd/yyyy.

Benefits
Type of veteran's or Social Security benefits student is receiving;

- D = dependent
- V = veteran
- N = neither

Birthplace
State of student's birth.

City
The city of the student's permanent address.

Class
Student's classification code from the Classification table. The classification may be automatically updated based upon flag and minimum values in the Classification table during TRANS Update.

Conc 1
Code from Concentration table.

Conc 2
Code from Concentration table.

Country
Country in which student's permanent address is located.

Declared Date
Date the program was declared. In most cases, it is the date that the Program Enrollment record was created.

Degree
Degree code from Degree table. Either entered when student applies for degree or when degree awarded.

Enrolled Date
Date student first enrolled in the institution.

Ethnic
Code from Ethnic table.

ID
Student ID number; query-only field.

Intended Degree
The intended degree of the student.

Last Sess Attend
Session and Year student last attended institution.

Last Upd
Last date on which ID record was updated.

Matric Date
Date the student became a matriculated student, based on institution's own criteria.

Major 1
Code from Major table.

Major 2
Code from Major table.

Marital
Marital Status code.

Max
Maximum number of hours for which the student may register.

Min
Minimum number of hours for which the student may register.

Minor 1
Code from Minor table.

Minor 2
Code from Minor table.

Name
Student Name; query-only field.

Res: Country/St/Cty
The country, state and county in which student currently resides.

Program
Program code from the Program table. A student will have one program enrollment record for every program he/she is registered in. The program code is the unique key for this record. This value is originally placed in this record from the adm_rec through a script.

Restricted Schedule
A flag indicating whether the student is restricted at time of registration. Currently not used by REGENT.

Sex
M/F indicator.

Social Security
Social Security number.

State/Zip
State and Zip Code of student's permanent address.

Subprog
Code of any subprogram in which student is enrolled.

Telephone
Student's permanent telephone number.

Title
Title (Mr. Ms, etc.)

Note: Three additional columns (ipeds_high_deg, ipeds_deg_yrs, and ipeds_deg_cmpl) have been added to the Program Enrollment record that are not displayed on this screen. These columns are maintained by the Update Program Enrollment menu option in the Registrar Reporting: IPEDS Graduation Rate Survey Reports menu. The values in these columns are based on student graduation information, the Graduation: Granted field on the second Program Enrollment screen, and information from the Degree table and Major table.

• ipeds_high_deg	Identifies the highest degree awarded for the Graduation Rate Survey.
• ipeds_deg_yrs	Indicates the number of years the student used to complete the degree.
• ipeds_deg_cmpl	Indicates if the degree was completed in 150% of the normal time.

The second Program Enrollment screen is accessed from the first screen by pressing <TAB> and contains the following data fields:

Graduation: Applied
Date student applied for degree or graduation.

Graduation: Bill For
Flag indicating if student is to be billed a graduation fee.

Graduation: Granted
Date, session, and year degree awarded to student. If the field contains a valid date (not blank or null), the student is assumed to be a graduate of the institution.

Graduation: Planned
Session and year student is expected to graduate. Could be entered at time the record is created based on a full-time four-year program.

Id No, Name, SS No.
From the ID file brought over from first screen.

Move to Alumni
Has an alumni record been created for student once graduation occurs? Y or N field.

Transcript: Official
Official transcript form code from Transcript Form table.

Transcript: Unofficial
Unofficial transcript form code from Transcript Form table.

Scroll screens may be accessed from the Program Enrollment screen. The screens that relate directly to maintaining academic records are:

Scroll Screen Name	**Record**	**Description**
Gpa/session	stu_acad_rec	Information about a specific session
gpa/Totals	stu_stat_rec	Cumulative statistical information

Record Relationships
The following diagram illustrates the relationships that exist between various records used to maintain student records:

```
                     Common ID <------- ID Info
                     detail records        |
                                           | is enrolled
                                           |
                                           |
                                           v
                 stu_serv_rec ------- prog_enr_rec -------stu_stat_rec
                                           |               (has cumulative statistics)
                                           |
                                           | has session statistics
                                           |
                                           |
                                           v
                                      stu_acad_rec -------sbcust_rec
                                           |               (has except bill info)
                                           |
                                           |
                                           |
                                           | register for courses
                                           |
                                           v
                                         cw_rec
                                           |
                                           |
                                           | has course info for billing
                                           |
                                           |
                                           v
                                         reg_rec
```

In each of the above record relationships, the first type of record can have many of the second type of record related to it. For example, a single stu_acad_rec can relate to multiple cw_recs.

Parameters for Regent

Introduction

CARS Solution contains parameters and compilation values for executing the Registration Entry program (*regent*). You can specify parameters to compile Registration Entry in a specified manner at the time of execution.

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

Parameter Syntax

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

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

Usage: regent -p program -F registration finish [-m mainopts] [-e enrollopts] [-r registopts] [-w] [-P] -L sitecode [-I] [-d] [-A] [-v]

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

Parameters

The following lists the –m mainopts, -e enrollopts, and –r registopts for running Registration Entry.

-m mainopts (to be disabled)

```
QUERY_OPT, /* ‘Q’ */
REGISTER_OPT, /* ‘R’ */
HOLDS_OPT, /* ‘H’ */
SESSION_OPT, /* ‘S’ */
INIT_OPT, /* ‘I’ */
CAPACITY_OPT, /* ‘P’ */
OUTPUT_OPT, /* ‘O’ */
CONFIRM_OPT, /* ‘C’ */
UNCONFIRM_OPT, /* ‘U’ */
FEE_OPT, /* ‘F’ */
EXIT_OPT, /* ‘E’ */
```

-e enrollopts (to be disabled)

```
DROP_OPT, /* ‘D’ */
VOIDSEC_OPT, /* ‘V’ */
ENROLDTL_OPT, /* ‘E’ */
CRSDTL_OPT, /* ‘C’ */
MTGDTL_OPT, /* ‘M’ */
LOCSEC_OPT, /* ‘L’ */
ALTCAL_OPT, /* ‘A’ */
REINSTATE_OPT, /* ‘R’ */
```

-r registopts (to be disabled)

```
DROPALL_OPT, /* ‘D’ */
VOIDALL_OPT, /* ‘V’ */
EDATE_OPT, /* ‘C’ */
REGFEES_OPTS, /* ‘I’ */
BLKADO_OPT, /* ‘B’ */
STUINFO_OPT, /* ‘E’ */
FINAID_OPT, /* ‘F’ */
DISPMTGS_OPT, /* ‘M’ */
DISPREFNO_OPT, /* ‘R’ */
LOCPRMS_OPT, /* ‘S’ */
SUBSESS_OPT, /* ‘T’ */
LOCALLSECS_OPT, /* ‘A’ */
LOADALLOW_OPT, /* ‘L’ */
```

How to Disable Options

One passes the hot key letter of the option to be disabled. If there is more than one option to be disabled, it should be separated from the other with a space. Disable the unconfirm option by making the pass as shown in the following example.

Example: –m U

Disable both confirm and unconfirm options by making the pass as shown in the following example.

Example: -m U C

The other parameters, -r and –e, can be passed any of the hot key letter values given in the submenus in the program. The –r option controls the submenu.

Drop all courses
Void all courses
Change effective date
Institutional fees
Block registration
view Enrollment
view Financial Aid
view Meetings
view Reference #
Set search criteria
Load all courses
subsess Totals
list All sections

The hot key is capitalized, as it is in the program. Pass the hot key letter to the menu option –r.

Disable the Change effective date by making the pass as shown in the following example.

Example: -r C

Registrar - Setting Up IPEDS Reporting

Introduction

Several ACE reports extract information from the CARS Solution database for standard IPEDS (Integrated Post-Secondary Education Data System) reporting. This document, used in conjunction with the IPEDS instructions provided by the NCES (National Center for Education Statistics), assists you in retrieving, organizing, and reporting such data.

Definitions

The following definitions are provided to help you understand this document:

First-time first-professional student

A student enrolled in any of the following or similar degree programs for the first time:

- Chiropractic
- Dentistry
- Medicine
- Optometry
- Osteopathic Medicine
- Pharmacy
- Podiatry
- Veterinary Medicine
- Law
- Theology

First-time freshman

An entering freshman who has never attended any college. First-time freshman status includes students who have attended any Summer sessions immediately preceding the Fall session, or are currently enrolled in the Fall session for the first time. First-time freshman status is usually designated by the code "FF" in the Student Academic Classification (cl) field in the Student Academic record (stu_acad_rec).

First-time graduate-level student

A student enrolled at the graduate level for the first time. This includes students who have attended any graduate Summer sessions immediately preceding the Fall session or who are currently enrolled in the Fall session for the first time.

First-time undergraduate transfer student

A student who has transferred to your institution in any Summer session immediately preceding the Fall session, or who is currently enrolled in the Fall session for the first time. The Transfer (trnsfr) field in the Admissions record designates a transfer student.

Remedial course

Any course section that has a "Y" in the Remedial (remedial) field in the Section record.

Overview of the IPEDS Reporting Process

Before running any IPEDS scripts or reports, you must update some specific fields in the Section record (sec_rec). The Section Remedial (remedial) and Section Credit (credit) fields must contain accurate and current data.

The next step is to verify that files in the database contain accurate and up-to-date information. The "Missing Student Data" report is run to obtain a list of missing data in student records. The "Update Age Of Students" script must also be executed to assure that student ages are accurate. See the *Database Requirements* section of this document for further information about these reports.

After the appropriate fields in the Section record (sec_rec) and the database have been properly updated, run the ipedprep script to prepare the Student Academic record (stu_acad_rec) for the IPEDS ACE reports. The two fields in this record maintained by the ipedprep script are the report on IPEDS (rpt_ipeds) and the IPEDS hours (ipeds_hrs).

Note:

Accurate enrollment counts cannot be assured if any of the above steps are omitted. The Section record (sec_rec) must be updated and the ipedprep script must be run before any IPEDS ACE reports are run.

After the preliminary requirements are complete, the IPEDS reports are ready to be run. There is no particular order in which the reports need to be run. Some of the reports require multiple executions for different population groups or programs.

The IPEDS ACE Reports

The ACE reports used for IPEDS enrollment reporting are located in the $CARSPATH/modules/regist/reports directory and may be accessed using the following procedure:

1. From the CARS Solution menu, select the Student Management menu.
2. From the Student Management menu, select the Registrar menu.
3. From the Registrar menu, select the Reporting menu.
4. From the Reporting menu, select the External Agencies menu.
5. From the External Agencies menu, select the IPEDS Reports menu.
6. Select the script or report you want from the following menu selections:
   - IPEDS Preparation
   - IPEDS First-Time Freshmen
   - IPEDS Enrollment
   - IPEDS Part or Fulltime Age
   - IPEDS Degree Completion
   - IPEDS Part D
   - IPEDS Part E

Some of these reports require special database setup. Please refer to the section entitled *Database Requirements* for further information.

Note:
This menu path, and other menu paths referenced in this document, may vary at your institution, depending on your needs and local customizations.

The following is a brief explanation of each of the ACE reports.

ipedff

> Menu option [b] for completing Part C of form EF1, "Residence of First-Time Freshmen." The report prints in two columns. The first column shows the number of first-time freshmen from each state who are degree-seeking. The second column reports the number of first-time freshmen by state who have graduated from high school or received a GED within one year prior to the date the report is run. The third column is not completed by running the report. However, space is provided for manual reporting for out-of- state centers.

ipeden

Menu option [c] for completing Part A of form EF1, "IPEDS Enrollment Summary by Racial/Ethnic Status." The report prints an enrollment summary of full-time and part-time students by classification and ethnic/racial status.

ipedage1

Menu option [d] for completing Part B of form.

ipedage2

EF1,"Enrollment of Students by Age." These reports print an enrollment summary of either full-time or part-time students by age, depending on the parameters you enter. The ipedage1 report provides data about students age 34 and under, and the ipedage2 report provides data about students over 35. The ipedage script, located in $CARSPATH/modules/regist/scripts, concatenates the reports into a single entity.

ipedcmpl

Menu option [e] for completing form C1, "IPEDS Completions." The report prints those students who have graduated between the input dates of the report according to major, sex, and ethnic background.

ipedenru.y

Menu option [i] for completing Part D. This report, titled "IPEDS Undergraduate Yearly Enrollment," prints the results from the "Create Enrollment Data IPEDS Yearly Summary" process for the undergraduate program.

ipedenrg.y

Menu option [i] for completing Part D. This report, titled "IPEDS Graduate Yearly Enrollment," prints the results from the "Create Enrollment Data IPEDS Yearly Summary" process for the program specified in the parameters.

ipeddhrs

Menu option [i] for completing Part D. This report, titled "IPEDS Part D – Credit and Clock Hours," prints the total number of credit and clock hours for registered and withdrawn coursework taken in the date range specified.

A script called ipedprep is also used in the IPEDS reporting process. This CARS Solution script is located in $CARSPATH/modules/regist/informers and is run from the first [a] menu selection on the IPEDS Reports menu.

The ipedprep script updates fields in the Student Academic record that are in turn used for IPEDS reporting. This report and how it should be used are explained in more detail in the *Running The Ipedprep SQL Query Language Script* section.

Local Customizations

To satisfy a broad range of clients, the IPEDS ACE reports are very general. Your institution may need to modify these reports to extract the appropriate enrollment information from the database. It is not expected that these reports will meet every need of each institution. For this reason, it is important that the CARS coordinator review the IPEDS ACE reports carefully, and determine if any local changes need to be made to them to accommodate differences in the definitions of student classifications and statuses.

Student Inclusions and Exclusions

All students who are enrolled in at least one course that is creditable toward a diploma, certificate, degree, or other formal award are counted in the IPEDS enrollment reports. Students enrolled in courses that are part of a vocational or occupational program are also included. A course is creditable toward a diploma, certificate, degree, or other formal award if the Section Credit (credit) field in the Section record contains a "Y." Therefore, any student who is enrolled in at least one course section with a Section Credit (credit) of "Y" will be counted in the IPEDS ACE reports.

Every student enrolled exclusively in courses that are not creditable toward a formal award or the completion of a vocational program is excluded in the IPEDS ACE reports. Students taking CEU's (continuing education units) are also excluded, unless they are also enrolled in courses creditable toward a degree or other formal award.

All students exclusively auditing classes are excluded from the IPEDS ACE reports. That is, students who are not enrolled in at least one course being taken for credit will be excluded.

Preparing the Section Record

The two fields used for IPEDS reporting in the Section record (sec_rec) are as follows. Each field contains "Y" or "N."

remedial
Indicates if a course section is considered a remedial section.

credit
Indicates if a course section is creditable toward a degree, certificate, or other award recognized by an institution.

Sections in the Fall session and any Summer sessions immediately preceding the Fall session must contain the appropriate information in these two fields. Some of the IPEDS ACE reports require information from summer sessions that immediately precede the Fall session in which the IPEDS reports are being run. A student is considered a first-time freshman in the Fall session even if that student took classes for the first time in a summer session preceding the Fall session. The IPEDS report for First-Time Freshmen requires information from the previous summer sessions to determine if the student was considered a first-time freshman during a previous summer session.

To access the Section record and update the Remedial and Credit fields, you must first access the Course Catalog screen and select a course. You can then select a section for the course.

The Course Catalog screen may be reached using the following procedure:

1. From the CARS Solution menu, select the Student Management menu.
2. From the Student Management menu, select the Registrar menu.
3. From the Registrar menu, select the Course/Class Schedule menu.
4. From the Course/Class Schedule menu, select Catalog Maintenance.
5. From the Catalog Maintenance menu, select Course catalog and schedule maintenance screens.

After you select or add a course, you can access the Section Record screen for the course. The Credit and Remedial fields on the Section Record screen allow updates of the sec_rec.credit and sec_rec.remedial fields.

Database Requirements

IPEDS ACE reports extract data from a variety of records. The validity of your reports depends on the accuracy of these records. All the IPEDS reports require that the proper Session, Year and Academic Program information be entered into the following records:

Record	**Technical Name**
Profile record	(profile_rec)
Education record	(ed_rec)
Student Academic record	(stu_acad_rec)
Course Work record	(cw_rec)
Program Enrollment record	(prog_enr_rec)

Since the IPEDS reports are collected in the Fall session, proper Academic Calendar records (acad_cal_rec) for the Fall session and any Summer sessions immediately preceding the Fall session must exist. Also, Student Academic records (stu_acad_rec) from Summer sessions must be retained for students who were enrolled in Summer classes and are currently enrolled in Fall classes. Each student must also have a Student Academic Registration Status (reg_stat) of "C" (for "Confirmed") to be counted in these IPEDS reports.

The "Missing Student Data" report should be executed to obtain a report of missing student data. This report is run by the following CARS Solution menu path:

1. From the CARS Solution menu, select the Student Management menu.
2. From the Student Management menu, select the Registrar menu.
3. From the Registrar menu, select the Reports menu.
4. From the Reports menu, select Enrollment Reports.
5. From Enrollment Reports, select Missing Student Data.
6. Enter your parameters for the following report criteria:
    - Session
    - Academic Year
    - Program
    - Subprogram
    - Status

You must also specify the type of output you want (to a screen, file or printer), and the time to run the process.

You should also execute the "Update Age Of Students" SQL Query Language script to assure that the ages of students are accurate. This report can be accessed using the following procedure:

1. From the CARS Solution menu, select the Student Management menu.
2. From the Student Management menu, select the Registrar menu.
3. From the Registrar menu, select the Session Processing menu, and enter the password to access the option.
4. From the Session Processing menu, select Update Age Of Students, and enter the time you want the process to execute.

The following lists additional database requirements that must be met for the IPEDS reports:

ipedff

Each student considered a first-time freshman must have the appropriate first-time code for freshman in the Student Academic Classification (cl) field in the Student Academic record. For example, your institution may use an "FF" code to denote a first-time freshman. The High School Graduation Date (grad_date) and the School ID (sch_id) fields in the Education record (ed_rec) must contain accurate data. If a student has received a GED instead of a high school diploma, the date the GED was received must be entered into the High School Graduation Date (grad_date) field. Also, the Degree Earned (deg_earn) field must contain a code to designate that the student received a GED. For example, your institution may use a "GED " code to designate a GED type award. In addition to these requirements, the School Category (ctgry) field in the School record (sch_rec) must contain an appropriate code for a high school (e.g., "HS ").

An additional requirement concerns the State table (st_table). The fips_code (referring to the Federal Information Processing Standard codes) must be entered for each state and province in the State table (st_table). If these values are not in the table, you must enter them to comply with the requirements of the first-time freshman report. You should assign any foreign countries a standard fips_code of 64. If you have any questions about these requirements, consult the NCES forms and instructions.

ipedenr

Each student's ethnic code (ethnic_code) and sex (sex) in the Profile record (profile_rec) must contain the proper values.

ipedage1 ipedage2

Each student's age (age) and sex (sex) must contain the proper values in the Profile record (profile_rec). Running the "Update Age Of Students" report can automatically revise ages.

ipedcmpl

For each student who has graduated between the range of the input dates of the report the degree granted date (deg_grant_date) must be stored in the Program Enrollment record (enr_rec). The First Major (major1) field must also contain an appropriate value, along with the Degree (deg) field. The sex (sex) and ethnic code (ethnic_code) in the Profile record (profile_rec) must also contain valid values.

ipedcq1

The Remedial (remedial) and Credit (credit) fields in the Section record (sec_rec) must contain the appropriate Y/N values.

ipedcq2

The Campus (mtg_campus) field in the Meeting record must contain valid campus values for each course in which students are registered.

ipedcq4

The Admissions Transfer (trnsfr) field in the Admissions record (adm_rec) must be set to "Y" for any students who are first-time undergraduate transfer students.

<u>Note</u>:

If any students were first-time undergraduate transfer students in any Summer session immediately preceding the Fall session, verify that the Admissions Transfer (trnsfr) field in the Admissions record (adm_rec) is set to "Y".

ipedcq5

There are no special database requirements for this report.

Running the Ipedprep SQL Query Language Script

The ipedprep script maintains the Report on IPEDS (rpt_ipeds) and IPEDS Hours (ipeds_hours) fields in the Student Academic record (stu_acad_rec). The values in these fields are dependent upon the values in the Credit (credit) and Remedial (remedial) fields of the Section record (sec_rec). These fields and their descriptions are as follows:

rpt_ipeds
Indicates whether the student with this record should be included in enrollment counts. Uses Y or N.

ipeds_hrs

Includes the total number of hours in which a student is currently registered in reference to the session of the record. This total does not include audit hours.

The ipedprep script is run one session at a time for every student that has a Student Academic record (stu_acad_rec) in that session. The report reads the necessary information from the Course Work record (cw_rec) for each student and only retains courses with a Course Work Status (stat) of "R" (for "Registered"). This script does not count courses that are being taken as an audit.

Next, the script looks at the Section record (sec_rec) for each Course Work record (cw_rec) of each student. If a student is enrolled in at least one course that is not being taken as an audit and has a credit (credit) value of "Y", then ipedprep places a "Y" in the Report on IPEDS (rpt_ipeds) field in the Student Academic record (stu_acad_rec).

An "N" is placed in the Report on IPEDS field in the Student Academic record under any of the following circumstances:

- If a student is registered only in courses being audited
- If a student is registered only in courses that have a credit code of "N"
- If a student is registered only in courses that have a remedial code of "Y"

The IPEDS reports only count students who are registered in classes that are creditable toward a degree, award, or other certificate. So, if a student's Student Academic record (stu_acad_rec) for the reporting session has a "Y" in the Report on IPEDS (rpt_ipeds) field, then that student will be included in the enrollment counts.

The ipedprep script also updates the IPEDS Hours (ipeds_hrs) field in the Student Academic record (stu_acad_rec). The script reads all the Course Work records (cw_rec) for a student, and totals all the non-audit registered hours. This value is placed in the IPEDS Hours (_ipeds_hrs) field in the Student Academic record (stu_acad_rec). Certain IPEDS reports only count students who are registered in at least one class that is not being audited. If the IPEDS hours (ipeds_hrs) value reflects that a student is taking more than zero hours, then the student will be counted in the IPEDS reports.

If a student has a Student Academic record (stu_acad_rec) for the IPEDS reporting session, which has a "Y" in the Report on IPEDS (rpt_ipeds) field and a value greater than zero in the IPEDS Hours (ipeds_hrs) field, then that student will be included in the enrollment statistics.

Running the IPEDS ACE Reports

After all the preliminary steps have been completed to prepare the records for the IPEDS ACE reports, the reports may be run. Following are descriptions of the reports and how to operate them.

IPEDS Enrollment

To produce the report for IPEDS Enrollment (ipedenr), you must supply the following parameters:

Parameter	**Sample Value**
Session (of the report)	"FA "
Academic year (of the report)	"199X"
Program	"UNDG"
Hours Fulltime	"12.0"

You also must specify the type of output you want (either to the screen, a file, or a printer), and the time the report should be processed.

This report produces full-time and part-time statistics according to student classification, ethnic background and sex. Run this report multiple times, once for each program at your institution.

IPEDS Part or Fulltime Age
The IPEDS Part or Fulltime Age reports require the following parameters:

Parameter	**Sample Value**
Session (of the report)	"FA "
Academic Year	"199X"
Program (primary)	"UNDG"
Program (other)	blank for none
Subprogram	blank for all
Hours (full- or part-time)	"F" for full-time, "P" for part-time
Hours Fulltime (primary program)	"12.0"
Hours (Fulltime, for other program)	"9.0"
Summary (is summary report wanted?)	Y/N
Output (printer)	"lpt"

You also must specify the type of output you want (either to the screen, a file, or a printer), and the time the report should be processed.

Note:
These reports use student classification macros to determine whether students are degree-seeking or not degree-seeking. These classifications should be reviewed and modified according to the policy of your institution.

The Part or Fulltime Age reports produce enrollment information according to the programs specified. Totals of students in degree seeking and non-degree seeking programs are listed according to age.

IPEDS Degree Completion
The report for Degree Completions requires the following parameter values:

Parameter	**Sample Value**
Beginning date (for the report)	"07/01/199X"
Ending date (for the report)	"06/30/199Y"
Program	"UNDG"

You also must specify the type of output you want (either to the screen, a file, or a printer), and the time the report should be processed.

This report can be executed for each program at your institution.

The Degree Completions Report lists those students who have completed degrees between the input dates according to major, sex, and ethnic background.

IPEDS First-Time Freshmen
The report of First-time Freshmen requires the following parameters:

Parameter	**Sample Value**
Session (of the report)	"FA "
Academic Year (of the report)	"199X"
Program	"UNDG"
Code (representing High School)	"HS "
Exam (code representing GED)	"GED "
Classification (for first-time freshmen)	"FF"
Beginning date (of previous session)	"01/01/199X"
Date (for the running of report)	"10/15/199X"

You also must specify the type of output you want (either to the screen, a file, or a printer), and the time the report should be processed.

The "Beginning date" parameter is the beginning date of the most recent Academic Calendar record session prior to any summer sessions, which immediately precede the "Session (of the report)". For example, if an institution is running this report in the FALL9X session, and has previous summer sessions SJ9X, SA9X, and a previous spring session SPRG9X, then the "Previous session beginning date" would be filled with the "beg_date" value for the SPRG9X session in the Academic Calendar record (acad_cal_rec).

If your institution has first-time freshmen who have received a GED instead of a high school diploma, then the code that your institution uses for a GED should be entered into the "GED code" parameter field.

To determine whether a student is a high school graduate or received a GED, the report looks at the student's graduation date (grad_date) in the Education record (ed_rec) to determine if it is within one year of the report date. If so, then the report finds the School record (sch_rec), which corresponds to the school ID (sch_id) from the Education record (ed_rec). The report looks at the Category (ctgry) field in the School record (sch_rec). If this value is "HS " or some other value that your institution has assigned to denote a high school, then the student is counted as a high school graduate within the last year.

In the case of the GED, the report looks only at the student's graduation date (grad_date) and the Degree Earned (deg_earn) fields in the Education record (ed_rec). If the date of graduation is within one year of the report date, and the code for GED is in the Degree Earned (deg_earn) field, then the student will be counted in column two of this report.

Note:
In order for a student with a GED to be counted in the above manner, the GED code must be entered into the Degree Earned (deg_earn) field in the Education record (ed_rec).

The report of First-time Freshmen prints the number of first-time freshmen per state and how many of those freshmen have graduated from high school, including those who have received a GED, within one year prior to the report date. The report date does not have to be the date on which the report is run. A column for Out-of-state centers by state is included for your convenience, but this column is not an automated portion of the First-time Freshmen report. An out-of-state center is a location other than the main campus where courses of your institution meet. This report includes only those states from which first-time freshmen originate.

IPEDS Part D
IPEDS Undergraduate Yearly Enrollment requires the following parameters:

Parameter	**Sample Value**
Session (of the report)	"FA "
Academic Year (of the report)	"200X"
Detail	"Y"

You also must specify the type of output you want (either to the screen, a file, or a printer), and the time the report should be processed.

This report prints the results of the Create Enrollment Data IPEDS Yearly for the undergraduate program.

IPEDS Part D
IPEDS Graduate Yearly Enrollment requires the following parameters:

Parameter	**Sample Value**
Session (of the report)	"FA "
Academic year	"200X"

Program	"GRAD"
Detail	"Y"

You also must specify the type of output you want (either to the screen, a file, or a printer), and the time the report should be processed.

This report prints the results of the Create Enrollment Data IPEDS Yearly for the program specified.

IPEDS Part D
IPEDS Part D – Credit Hours requires the following parameters:

Parameter	**Sample Value**
Session (of the report)	"FA "
Academic Year	"200X"
Program	"UNDG"

You also must specify the type of output you want (either to the screen, a file, or a printer), and the time the report should be processed.

This report prints the total number of credit hours for registered and withdrawn coursework within a specified date.

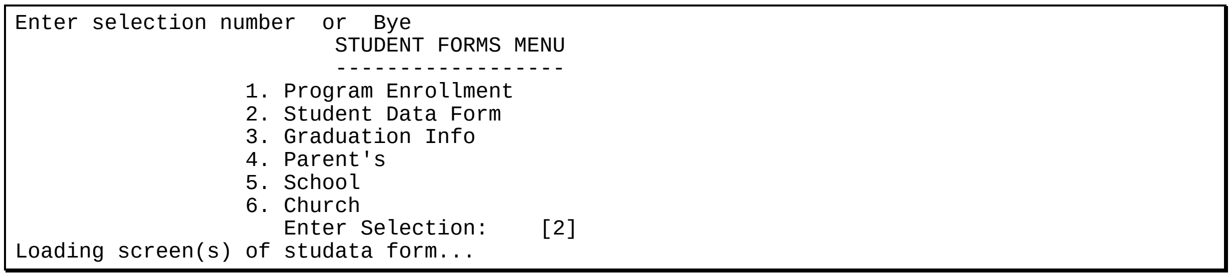
Registrar - The Student Profile Entry Program

Overview
The Student Profile Entry program allows data entry of all profile information for students, and student organizations, churches and schools utilizing user-customized screens.

Introduction
STUENTRY is the data query/entry program for the Registrar's and Student Services Office. It allows the user to perform all necessary functions of data retrieval, entry and modification of profile information that these offices deal with. It utilizes user-customized screens, which can easily be made to simulate the actual forms, including student data sheet, housing forms, etc. Included in its features are ID/Name lookup for parents, home church, advisor, etc. Entry of course information is done with the REGIST program, but may be displayed with STUENTRY.

Procedures
STUENTRY uses various screen formats for display and data entry purposes.

STUENTRY FORMS MENU Screen
After loading the tables, opening the files necessary, and loading the screens for the menu, STUENTRY displays the menu of current forms where either a form is selected or a (B)ye exits from STUENTRY. This screen is user-customized to include all available forms for a particular school. A document explaining how to build the various forms needed for a school is described in section 8 (CARS Entry Library Program) and section 9 (Creating Screens and Form Definition Files) of the *CX System Reference Technical Manual*.



```
Enter selection number  or  Bye
                         STUDENT FORMS MENU
                         ------------------
                  1. Program Enrollment
                  2. Student Data Form
                  3. Graduation Info
                  4. Parent's
                  5. School
                  6. Church
                     Enter Selection:    [2]
Loading screen(s) of studata form...
```

The STUENTRY FORMS screen allows the operator to select the form desired on which to do data entry/display. The operator selects the number associated with the form. Any number of user customized screens can be created depending on the variety of data-input forms used by the school.

Query Command
Upon selecting a screen for data entry/update, STUENTRY displays the screen selected and stops at the field where the ID number is located. A sample form of a student data sheet looks like this:

```
PF1 execute.  CTRL C abort.  <CR> verify.
                                                           ** Query Mode **
                          STUDENT DATA FORM     ID No.....[0     ]
Name......                                      Advisor...
Local Add.                                      Begin/End.          /
    ......
    ......
City......
Permanent.
    ......
City......                                           Telephone...
Birthdate.              Birthplace...                               Gender..
Ethnic....
Undergrad.                                            Type..        Code..
Degree....
Major.....                                            Graduation Year.....
```

At this point, STUENTRY enters query mode. The operator may type in an ID number, if known, or query by name or social security number. A CTRL C will take the operator back to the STUENTRY FORMS MENU where another form may be selected.

If an ID number is typed followed by a Return key, the ID information and default values supplied on the screen are filled in and the operator is left on the ID field. This is to verify that the correct ID number was entered. If it was the right record, typing the PF1 key will then cause the person to be selected and all information for that person will be read and displayed on the form. The PF1 key may be typed immediately after putting in the ID number which will select the person without first displaying the ID information.

An example of the screen after the ID number was entered followed by the Return key:

```
PF1 execute.  CTRL C abort.  <CR> verify.
                                                           ** Query Mode **
                          STUDENT DATA FORM     ID No.....[1000  ]
Name...... Jones, Mary Jo                       Advisor...
Local Add.                                      Begin/End. 00/00/00 / 00/00/00
    ......
    ......
City......
Permanent. 1000 South Locust
    ......
City...... Oxford                OH 45056            Telephone...513-523-8822
Birthdate. 00/00/00     Birthplace...                               Gender.. M
Ethnic.... WH
Undergrad.                                            Type..        Code..
Degree....
Major.....                                            Graduation Year.....
```

If the ID number is not known, typing a Return or the PF1 key with a zero or blank in the ID field will cause STUENTRY to ask the following question:

```
 Query on Program Enrollment or Id records?  (P or I).
```

Typing an 'P' will cause only people with a Program Enrollment Record to be displayed when querying, while typing an 'I' will bring up anyone with an ID record. The routine that does the query is called DUFIND, which is explained in the document, 'Using the ID Record Lookup Screen'. Basically, queries based on ID number, name or social security number can be done. If a Control C is typed while in DUFIND, STUENTRY will attempt to add an ID record for the person. Permission for addition may be selectively removed from STUENTRY so that a Control C would put the operator back in query mode on the form selected.

After either selecting a person or typing a Control C to add ID information, our screen will look like the following (except if ID is being added, all the fields will be blank, ready to be filled in):

```
PF1 execute  CTRL C abort  CTRL O check field                      Screen 1 of 2
TAB next screen  CTRL U scroll screens  CTRL W help            ** Update Mode **
                          STUDENT DATA FORM     ID No.....[1000  ]
Name...... Jones, Mary Jo                       Advisor...
Local Add. 845 West College                     Begin/End. 09/11/86 / 05/10/87
    ......
    ......
City...... Oxford                  OH 45056
Permanent. 1000 South Locust
    ......
City...... Oxford                  OH 45056          Telephone...513-523-8822
Birthdate. 00/00/00     Birthplace...                               Gender.. F
Ethnic.... WH
Undergrad.                                            Type..        Code..
Degree....
Major.....                                            Graduation Year.....
```

The right side of the top 2 lines (command panel) will vary according to the mode and the number of screens on the form. The mode, displayed on the second line from the top of the screen, refers to whether or not the person selected has an ID Record. If they do not have an ID record, STUENTRY will be in 'Add' mode. If the person selected already has an ID Record, STUENTRY will be in 'Update' mode, which is shown in our example.

The other difference depends on how many screens are defined for this form. How to set up and modify STUENTRY screens is defined in section 8 (CARS Entry Library Program) and section 9 (Creating Screens and Form Definition Files) of the *CX System Reference Technical Manual*. If there is more than one screen, the option 'TAB next screen' is also displayed. Also, above the mode is the number of the screen currently displayed as well as the total number of screens. 'Screen 1 of 2' is an example.

STUENTRY will now allow the operator to update or add information to any field displayed on the screen, as well as the scroll files which will be discussed later. The field that the operator is on will always be delimited by an open and close bracket, '[' and ']'. Typing the Return key causes STUENTRY to go to the next field defined in the screen file as the next field. Thus, it may move down to the next field in the column or move across to the next column depending on the form. The arrow keys may also be used to go directly to a field, i.e., up, down, left, or right.

PF1 causes all files and fields displayed on the form to be either added, updated, or deleted depending on information entered on the screen. Permissions for addition, update and deletion are also checked for each file and field. All additions, modifications and default values are considered. Thus, a record will be added, (if a record does not already exist), if any field from that file is added or defaulted in from the screen. The files and fields in any of the scroll screens will also be updated at this time.

CTRL C will abort the add, discarding all additions or modifications of any field on the form, including any scroll files accessed. The operator will be placed on the ID field in query mode on the same form.

Typing PF1 or CTRL C will clear the screen and assume that another record wants to be entered using the same form. The ID number of the previous record is displayed at this point. Two reasons for this are in case an ID record was added and the ID number was not known (it can now be written on the hard-copy form) and in case the operator wanted to verify the data just entered. To return to the STUENTRY FORMS MENU screen, another CTRL C while in query mode must be typed. This defaulting to query mode saves time when processing multiple people having the same form, i.e. multiple student data sheets.

Typing the TAB key (if form has more than 1 screen) will cause STUENTRY to skip to the first modifiable field on the next screen of the form. Also, STUENTRY will automatically go to the next screen after the last field on a screen has been accessed. The TAB key is provided if the operator does not wish to input data on the whole first screen or the operator desires to simply query on the information on the screen without going through the fields.

CTRL W brings up help screen(s) describing the various commands available to the user.

CTRL O allows the operator to do either of the following: A table lookup on a code or a lookup on a name or social security number when an ID number is required. If STUENTRY is currently on a field that does not have a table lookup feature or is not an ID number, nothing will happen.

CTRL U may be typed at any time to bring up a scroll screen menu.

The operator may type the letter associated with any file desired to be accessed. For the sake of clarity, these files will be referred to as "scroll files" since the scrolling feature is used to display and update them. Scrolling refers to having more than 1 record displayed on the screen at one time. After a file is selected, the bottom lines on the screen will be replaced with the file selected, depending on how large the scroll screen was designed. Using our example, the screen will look like this, (after typing a 'C' to bring up Contacts):

```
PF1 execute.  CTRL C abort.  CTRL F fast-forward.  CTRL B fast-backward.
CTRL O add record. CTRL E erase line. CTRL U scroll screens.
                          STUDENT DATA FORM     ID No.....[1000  ]
Name...... Jones, Mary Jo                       Advisor...
Local Add. 845 West College                     Begin/End. 09/11/86 / 05/10/87
    ......
    ......
City...... Oxford                  OH 45056
================================== CONTACT ================ Record   1 of  12 =
  Contact                          St Contacted     Due  Corresp   Added  Rt CGC
 SDS      Student Data Sheet        C  05/04/85  00/00/00 0      05/03/85  O  Y
 GRDRPT   Final Grade Report        C  06/02/85  06/02/85 0      06/02/85  O  N
 TRANS    Transcript                C  06/19/85  00/00/00 0      06/19/85  O  Y
 SDS      Student Data Sheet        C  07/19/85  00/00/00 0      07/19/85  O  Y
 DEANLIST Dean's List Letter        C  08/13/85  08/12/85 0      08/13/85  O  Y
 HOUSING  Housing Letter            C  09/15/85  09/14/86 0      09/15/85  O  Y
 SDS      Student Data Sheet        C  11/23/85  00/00/00 0      11/23/85  O  N
 TRANS    Transcript                C  02/03/86  00/00/00 0      02/03/86  O  N
```

STUENTRY will load all records that already exist on this scroll file for this person. The message 'Record x of y', which appears on the same line as the name of the scroll file, identifies the current record and the total number of scroll records that exist for this person. Note that not all 12 fit in the area provided. A CTRL F or the down arrow may be used to display the rest of the records. Also note that if none of these records exists for this person, STUENTRY defaults to adding a record and the message at the top will be 'Record 1 of 1'. If nothing is typed in the fields, no record will be added upon exit from the scroll file. Typing anything over a record that is already displayed will modify that record, but will NOT add a new record. A CTRL O must be typed to add a new record.

PF1 brings the operator back to the screen from which this option was called.

CTRL C brings the operator back to the scroll file level, allowing another scroll file to be entered, as well as all additions or modifications made within this scroll file to be discarded.

CTRL F will display another screen of records for the person, if any more records exist than can be displayed on the bottom of this screen.

CTRL B will display the previous screen of records, after a CTRL F was used.

CTRL O must be typed to add an additional record for this person. If operator does not have permission to add this record, this option will not work.

CTRL E will erase the current record of this scroll file, (i.e., the record on which the cursor is resting when CTRL E is typed). If operator does not have permission to delete this record, this option will not work. Also, if the operator deletes the last scroll file, STUENTRY will put the operator back at the scroll file level, allowing another scroll file to be accessed.

CTRL U will cause the command panel to change to the prompt for which scroll file is desired, allowing the operator to select another scroll file.

Parameters

If STUENTRY is to be used only for query purposes, the "-d" parameter may be given to run STUENTRY in display only mode. This mode restricts the available commands to querying on records.

The "-t" parameter is for passing a date to STUENTRY to override the default value of today being the system date. This is used internally when STUENTRY adds records and checks for valid dates, but it will not override the defaults to 'today' declared in the SCR files.

The "-m" parameter tells STUENTRY which menu file to use for this session. If no "-m" is passed, the default menu name, "csmenu", will be used.

The "-f" parameter is for restricting the operator to a specific form. If the "-f" parameter is passed with a valid form, STUENTRY will not bring up the STUENTRY FORMS MENU with the list of forms, but go into query mode on the form specified.

The "-T" parameter is for overriding the default tickler code. This restricts the operator to only displaying and updating tickler records and contacts with the specified tickler code.

The "-D" parameter is for causing STUENTRY to print debugging messages about which tables and files are being updated. This is normally only used by the programmer for debugging purposes.

Compilation Values

The normal constituent entry process may be modified using a compilation flag defined in /usr/include/CARSV/custom/stuentry.h. After any modification to this file, do a "make reinstall" in the STUENTRY source directory (/usr/src/CARSV/develop/stuentry).

Program Flow

The following diagram summarizes the logic flow followed by STUENTRY.

```
    --------------------------
(1) | Perform Initialization |
    | and Process parameters |
    --------------------------
               |
               |    If form parameter passed
               |---------------------------------------------------------->
               |                                                          |
               |<------------------------------------------------------<  |
               v                                                       |  |
    --------------------------------                                   |  |
(2) | Optionally Load Menu Screen  |                                   |  |
    | Display Menu Screen          |                                   |  |
    | Select Form to use           |                                   |  |
    --------------------------------                                   |  |
               |                                                       |  |
               |   Bye   ----------------                              |  |
               |-------->| Exit Program |                              |  |
               |         ----------------                              |  |
               |<---------------------------------------------------------<
               |                                                       |
               v                                                       |
    ----------------------------------                                 |
(3) | Load Form Selected or Passed   |                                 |
    | Initialize File Record Buffers |                                 |
    ----------------------------------                                 |
               |                                                       |
               | If form not passed                                    |
               |-------------------------------------->                |
               |                                      |                |
               |<-------------------------------------|--------------< |
               v                                      v              | |
    ----------------------------           -----------------------   | |
(4) | Display Command Panel    |       (5) | Entry Routine -     |   | |
    | Display Screen 1 of Form |           |   (Expanded below)  |   | |
    ----------------------------           -----------------------   | |
               |                                      |              | |
               |                                      |              | |
               |                                      -----------------^
               |                                                     |
               |                                                     |
               |                       -----------------------       |
               |  'E' entered      (5) | Entry Routine -     |       |
               |-----------------------|   (Expanded below)  |-------^
               |                       -----------------------
               |
               |  'B' entered   ----------------
               >--------------->| Exit Program |
                                ----------------
```

```
Expansion of Entry Routine:
               |<------------------------------------------------<
               v                                                 |
        --------------                                           |
    (6) | Query Mode |                                           |
        --------------                                           |
               |                                                 |
               |  SCR_ABORT returned     -------------------     |
               |------------------------>| Exit Entry Mode |     |
               |                         -------------------     |
               |                                                 |
               |  If display only        -------------------     |
               |------------------------>| Display Screens |-----^
               |                         -------------------     |
               |                                                 |
               |                         -------------------     |
               >------------------------>| Add/Update Mode |-----^
                                         -------------------
Expansion of Query Mode:
        ---------------------
        | Clear Rec Buffers |
        ---------------------
               |
               |<------------------------------------------------<
               v                                                 |
        -------------------                                      |
        | Enter Id Number |                                      |
        -------------------                                      |
               |                                                 |
               |                                                 |
               |  SCR_ABORT returned   --------------------      |
               |---------------------->| Exit Query Mode  |<--   |
               |                       --------------------  |   |
               |                                             |   |
               |  ID Number of 0       --------------------  |   |
               |---------------------->| Find Id function |--^   |
               |                       | Load related info|  |   |
               |                       --------------------  |   |
               |                                             |   |
               |  SCR_DONE returned    --------------------  |   |
               |---------------------->| Display Id Info  |--^   |
               |                       | Load related info|      |
               |                       --------------------      |
               |                                                 |
               |  SCR_ENTER returned   --------------------      |
               |---------------------->| Display Id Info  |------^
               |                       --------------------      |
               |                                                 |
               >-------------------------------------------------^
```

(1) Perform Initialization and Process Parameters

(2) Optionally load and display STUENTRY FORMS MENU screen and allow user to enter form desired or exit STUENTRY by typing B(ye). This would be skipped if the '-f' parameter was passed with a valid form.

(3) Load and display form selected (or passed) and initialize all the record buffers for each file displayed on the form.

(4) Display Command Panel for what action is desired. This only applies if the form is passed.

(5) 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 which is discussed next.

(6) Query Mode. The user enters an ID 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 is put in Query mode again.

Program Errors

The following error messages may occur while running STUENTRY:

```
Aborting due to unexpected return from scr_getset of STATUS
```

This may only occur as additional return statuses are implemented in the screen handling routines and indicates that the program needs to be updated to handle the new return status.

```
Add field 'FIELDNAME' not found in FILENAME file.
```

Requires program modification. The given fieldname as 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 given fieldname as specified in the commonfld array did not appear in the view of the given filename.

```
Field 'FIELDNAME' specified in tblchk array was not found in the FILENAME file
```

Requires program modification. An inconsistency exists in the tblchk array in which 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 tblchk array
```

Requires program modification. In inconsistency exists in the tblchk array in which the field specified does not exist in the current view of the file specified for the field.

```
File updating has been aborted.
```

This message is printed if an error occurs in some of the first files, which are updated causing the updates of the subsequent files to be aborted.

```
Filename 'FILENAME' from updateorder array was unrecognized.
```

Requires program modification. A filename exists in the updateorder array which does not exist in the filename array.

```
Getkey field 'FIELDNAME' not found in FILENAME file.
```

Requires program modification. Either the key or one of the components of the key does not appear in the view of the given filename.

```
Help routine has not yet been implemented.
```

Message is printed any time that the help feature is invoked.

```
Invalid command
```

Message is printed if an invalid key is entered when waiting for a command.

```
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 doesn't know what to do with this record. Possibly due to a change in the definition of the record statuses that has not been fully implemented or due to the buffer unexpectedly being overwritten.

```
Libentry: Parameters not found: auto_mode
```

Program modification required. The entry library requires these variables to be defined in the prog_params array.

```
Libentry: Parameters not found: critical_files
```

Program modification required. The entry library requires these variables to be defined in the prog_params array.

```
Libentry: Parameters not found: debug_level
```

Program modification required. The entry library requires these variables to be defined in the prog_params array.

```
Libentry: Parameters not found: menuname
```

Program modification required. The entry library requires these variables to be defined in the prog_params array.

```
Libentry: Parameters not found: scr_path
```

Program modification required. The entry library requires these variables 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
```

The user either does not have the database permissions to add this type of record to the system or the program specifically does not allow the addition of this record type.

```
Permission is not granted to delete records
```

The user either does not have the database permissions to delete this type of record to the system or the program specifically does not allow the deletion of this record type.

```
Status occurred on scr_get of STATUS                        entmain.c
```

This error may indicate that the menu form does not have a screen field with the name of 'a'.

```
Table 'TABLENAME' specified in tblchk array was not found in tablename array.
```

Requires program modification. A tablename is used in the tblchk array which is not already defined in the tablename array.

```
The FILENAME file's putkey, 'KEYNAME', is not unique. STATUS
```

The putkey specified in the filename array for the given filename is not a unique key. All putkeys are required to be unique keys.

```
The FILENAME record appears to have been deleted. Please check.
```

This error may occur on an update or delete of a record, if the program retrieved a record which it did not lock and before updating/deleting the record someone else deleted the record from the database.

```
The FILENAME record was modified before your changes. Please check.
```

The error may occur on an update or delete of a record, if one of two conditions exist. First, if the program retrieved a record which it did not lock and before updating/deleting the record someone else modified the record. Secondly, if one of the programmer supplied routines modifies the records previous buffer.

```
The form 'FORMNAME' was not found.
```

The specified form expected by the program could not be located. Verify that the form exists and has been installed.

```
The value of the FIELDNAME code, 'FIELDVALUE', was not found in 'TABLENAME'.
```

The operator entered a value which was not found in the table.

```
Update field 'FIELDNAME' not found in FILENAME file.
```

Requires program modification. The given fieldname as specified in the updatefld array did not appear in the view of the given filename.

```
dbadd: status = STATUS on file FILENAME.
```

Refer to the Informix manual as to the meaning of the status. These errors may or may not case termination of the program depending upon their severity; which, for the most part, is dependent upon whether the error occurred on a table or a file.

```
dbdelete: status = STATUS on file FILENAME.
```

Refer to the Informix manual as to the meaning of the status. These errors may or may not case termination of the program depending upon their severity; which, for the most part, is dependent upon whether the error occurred on a table or a file.

```
dbfile: status = STATUS on file FILENAME.
```

Refer to the Informix manual as to the meaning of the status. These errors may or may not case termination of the program depending upon their severity; which, for the most part, is dependent upon whether the error occurred on a table or a file.

```
dbfind: status = STATUS on file FILENAME.
```

Refer to the Informix manual as to the meaning of the status. These errors may or may not case termination of the program depending upon their severity; which, for the most part, is dependent upon whether the error occurred on a table or a file.

```
dbfind: status = STATUS on table TABLENAME.
```

Refer to the Informix manual as to the meaning of the status. These errors may or may not case termination of the program depending upon their severity; which, for the most part, is dependent upon whether the error occurred on a table or a file.

```
dbindex: status = STATUS on file FILENAME and key KEYNAME.
```

Refer to the Informix manual as to the meaning of the status. These errors may or may not case termination of the program depending upon their severity; which, for the most part, is dependent upon whether the error occurred on a table or a file.

```
dbselect: status = STATUS on database DBNAME.
```

Refer to the Informix manual as to the meaning of the status. These errors may or may not case termination of the program depending upon their severity; which, for the most part, is dependent upon whether the error occurred on a table or a file.

```
dbselect: status = STATUS on file FILENAME.
```

Refer to the Informix manual as to the meaning of the status. These errors may or may not case termination of the program depending upon their severity; which, for the most part, is dependent upon whether the error occurred on a table or a file.

```
dbselfield: status = STATUS on file FILENAME.
```

Refer to the Informix manual as to the meaning of the status. These errors may or may not case termination of the program depending upon their severity; which, for the most part, is dependent upon whether the error occurred on a table or a file.

```
dbstructview: status = STATUS on file FILENAME.
```

Refer to the Informix manual as to the meaning of the status. These errors may or may not case termination of the program depending upon their severity; which, for the most part, is dependent upon whether the error occurred on a table or a file.

```
dbupdate: status = STATUS on file FILENAME.
```

Refer to the Informix manual as to the meaning of the status. These errors may or may not case termination of the program depending upon their severity; which, for the most part, is dependent upon whether the error occurred on a table or a file.

```
dmm_add: error on form FORMNAME and screen SCREENNAME.
```

Refer to the "Out of memory" error.

```
dmm_add: error on form FORMNAME, 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.                           formload.c
```

Requires program modification. The given fieldname is a component of the getkey which 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 the getkey which does not appear in the view of the given filename.

```
ent_writerec.c: Positioning in FILENAME failed with a STATUS status.
```

Refer to the Informix manual as to 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".

Crash Recovery

If STUENTRY exits unexpectedly, the database may need to be updated. The only time the database is updated by STUENTRY is after a person has been selected and a subsequent PF1 has been typed. All records in that form for the person are then added or updated. STUENTRY then goes into Query mode for selecting the next person. The critical time is after the PF1 is typed and before operator is put back in Query mode for entering the next person. The records for that person should all be checked for accuracy, and if some were not added, they should be added again. An example might be the alumni record was modified but the interest record just put in was not added.

Database Input

The following database files are read by STUENTRY:

```
Major Files:        id_rec
                    profile_rec
                    progenr_rec
                    stustat_rec
                    tick_rec
                    church_rec
                    sch_rec
Scroll Files:       aa_rec
                    accomp_rec
                    addree_rec
                    ctc_rec
                    ed_rec
                    empr_rec
                    exam_rec
                    hold_rec
                    int_rec
                    involve_rec
                    rel_rec
                    relsec_rec
                    stuacad_rec
                    stupers_rec
                    stuserv_rec
Tables:             aa_table
                    acadstat_table
                    accomp_table
                    cl_table
                    conc_table
                    ctc_table
                    ctry_table
                    cty_table
                    deg_table
                    denom_table
                    ethnic_table
                    exam_table
                    hand_table
                    hold_table
                    holdact_table
                    int_table
                    involve_table
                    major_table
                    minor_table
                    occ_table
                    ofc_table
                    prog_table
                    rel_table
                    sch_table
                    sess_table
                    st_table
                    stupers_table
                    subprog_table
                    tick_table
                    title_table
```

Database Output

The following database files are updated by STUENTRY:

Major Files:	id_rec
	profile_rec
	progenr_rec
	stustat_rec
	tick_rec
	church_rec
	sch_rec
Scroll Files:	aa_rec
	accomp_rec
	addree_rec
	ctc_rec
	ed_rec
	empr_rec
	exam_rec
	hold_rec
	int_rec
	involve_rec
	rel_rec
	relsec_rec
	stuacad_rec
	stupers_rec
	stuserv_rec

Output Samples

There is no hardcopy output generated by STUENTRY.

Registrar - Using Registration Audit

Introduction

Registration Audit (REGAUDIT) audits the counts for registered and wait list students stored in the Section record. REGAUDIT produces a report showing the registered and wait list student counts from the Section record with the actual counts from the Course Work records. The audit report generated by REGAUDIT is normally sent to the terminal display unless the output is redirected to an output file. The program automatically updates registered and wait listed values in any incorrect Section records.

It is recommended that REGAUDIT be run during off-hours of heavy registration and before running enrollment reports. REGAUDIT should also be run at a time when the REGIST program is not running to prevent both programs from updating the Section record at the same time.

Procedures

REGAUDIT does not require any operator interaction while running since it is a background process.

Parameters

You may set several parameter options within REGAUDIT. REGAUDIT can be run to audit all courses in a session, audit individual courses, audit individual sections for a course, update audit errors automatically, and skip printing of the audit report. The following list shows each of the valid parameter options available:

-C catalog
Catalog code (optional)

-N course
Course number (optional)

-S section
Section number (optional)

-q
Do not print audit report (optional)

-s sess
Session code (eg. FA) (required)

-u
Update errors found (optional)

-y year
Calendar year, eg. 199X (required)

The REGAUDIT program requires the "-y" and "-s" options; all other parameter options are optional.

The following are examples of valid parameter combinations used to run the different versions of REGAUDIT.

- The version of REGAUDIT that updates every sec_rec for an academic year and session, and that also prints the sections that are updated is as follows:
  - regaudit -y YYYY -s SSSS -u
  - where "YYYY" is an academic year such as "199X" and "SSSS" is a session code from the sess_table.
- If the operator chooses only to print the report without updating the records, REGAUDIT should be run with the following parameters:
  - regaudit -y YYYY -s SSSS
- To update all the records in the sec_rec for a year and session, but not print the report use the following version:

  - regaudit -y YYYY -s SSSS -u -q
  - The "-q' parameter turns off the printing.
- To update a specific course, the course catalog and course number parameters are required as shown in the following example:
  - regaudit -y YYYY -s SSSS -C CCCC -N NNNNNNNN -u
  - The course catalog is shown as "CCCC" and must be valid according to the cat_table. The course number, "NNNNNNNN", must be a course in that catalog. Only the Section records for that course in that catalog will be updated.
- If it is known that one section of a course has incorrect req_num and/or wait_num values, the following version can be run:
  - regaudit -y YYYY -s SSSS -C CCCC -N NNNNNNNN -S ss -u
  - here "ss" is the section value.
  - Note: If a course number is in the catalog with blanks as characters, enter the course number with those blank spaces. If a section is right justified, a space must be entered before the section number. For example, if section "1" for Art 100 was right justified and Art 100 contained a space, the previous example should be entered as: regaudit -y 199X -s FA -C CC9X -N "ART 100" -S " 1" -u

The menu options for REGAUDIT are located in $CARSPATH/menuopt/regist/programs/rgau. Refer to this file, and the *CX Implementation and Maintenance Technical Manual*, when making changes to menu options.

Compilation Values

REGAUDIT does not use any macro constants or compilation values customizable per installation.

Program Flow

REGAUDIT processes the records in the sec_rec one record at a time for the records matching the parameter values. If the "-y" and "-s" parameters are the only parameters, all Section records for the session (sec_rec.sess) and year (sec_rec.yr) are processed. If the "-N", "-C", or "-S" parameters are used, only the Section records matching the parameter values for course (sec_rec.crs_no), catalog (sec_rec.cat), or section number (sec_rec.no) are processed for the session and year specified.

After a Section record is found, the program searches for Course Work records (cw_rec) matching the Section record. As Course Work records are found, the program checks cw_rec.stat value and totals the registered and wait list counts. The registered count is incremented when records with cw_rec.stat value of "R" are found. The wait list count is incremented when a record with a cw_rec.stat equal to "L" is found.

When all the Course Work records have been found for a course section, the registered and wait list counts are compared to the reg_num and wait_num fields in the sec_rec found. If the "-u" option was given as a parameter, the reg_num and wait_num values are updated if they are incorrect. If the "-q" parameter was not used, REGAUDIT will write out a line for the audit report for the course with incorrect counts.

Program Errors

REGAUDIT does not use mail to send program processing errors to the operator. Error messages are sent directly to the terminal display , or, if the program's output is redirected, error messages are placed in the output file.

Database CC.
DBSELECT:status = NN

An attempt to open the database "CC" failed resulting in the error number "NN". If the error number is 6001, rerun the program. The 6001 error can occur as a result of a dbstatus or dbbuild process running at the same time.

"file". DBSELECT:status = NN

The database file "file" could not be opened because of the error indicated by the "NN" error number.

"file". DBSTRUCTVIEW:
status = NN
If this error occurs, contact the CISC Response Center. This error results from a dbstructview function call on the file "file".

"key". DBSELFIELD:
status = NN
The database record index "key" cannot be selected by REGAUDIT because of the error indicated by the error number "NN". Verify that the database index "key" exists in the database through the INFORMER "print status for ..." command.

Bad parameter: "PP".
An invalid parameter argument "PP" was passed to REGAUDIT. See the "Parameters" section above for valid parameter options and rerun the program.

Usage: regaudit -y year -s sess [-C cat] [-N crs_no] [-S sec] [-u] [-q]
This message is printed when an invalid parameter option or parameter combination was used when running REGAUDIT. See the "Parameters" section above for valid parameter options and rerun the program.

**** ERROR on "crs_no" "sec_no" -- DBUPDATE status=NN ****
A database update error "NN" occurred when REGAUDIT attempted to update the sec_rec where the crs_no equals "crs_no" and sec_no equals "sec_no".

Crash Recovery
If REGAUDIT exits unexpectedly or an error is given by the program, the program can be rerun after the error has been resolved.

Database Input
The only database tables used for input to REGAUDIT are the sec_rec and the cw_rec.

Database Output
Only the sec_rec table is updated by REGAUDIT.

If the REGAUDIT Report is routed to a printer or to the screen display, you will receive an electronic mail message that indicates the number of incorrect records and the number of errors.

If the report is routed to a file, you will receive two mail messages: one indicating the file location of the report, and another indicating the number of incorrect records and the number of errors.

Registrar - Using Registration Lists

Introduction

REGLIST can produce a class list for any section of any course in any course catalog. Class lists may be generated for all registered and withdrawn students for each section sorted by name. Missing grade lists may also be created containing only those students who have not received grades for that course section. Wait lists may be created with all wait listed students sorted by the date and time they were added to the list. Once the lists have been created, you can run FPS to print them.

Running REGLIST

REGLIST is normally run as a background process, since it does not require any operator interaction.

Parameters

Following is a summary of the parameters available and their usage:

-a
: Create lists for all sections, even those with no students

-B
: Print both confirmed and financially cleared students

-c class
: Select sections for a specific class (freshmen, sophomores, etc.)

-C catalog
: Select sections for the specified catalog code

-d date
: Beginning of section end date range

-D department
: Select sections for the specified department

-e date
: End of section end date range

-f
: Sort lists by faculty name abbreviation

-F form
: Use the specified FPS form file for formatting

-G scantype
: Type of scanning form to be used (for grade scanning)

-i
: Ignore cross-listed courses (for cross-listed class lists)

-I
: Print only students with financial clearance

-j
: Include students who dropped on or after the first day of class

-k
: Exclude withdrawn students

-l campus
: Select sections for the specified campus

-L site

Select sections for the specified site

-m
Create missing grade lists only

-M
Print only matriculated students

-N course
Select sections for the specified course number

-O
Allow reprint of class lists for scanning application (for grade scanning)

-r
Sort students by random number (for anonymous grading)

-R
Print only confirmed students

-s sess
Select sections for the specified session code

-S section
Select sections for the specified section number

-t text
Up to 80 characters text

-u subsess
Select sections for the specified subsession code

-w
Create only wait lists

-W weekday
Select sections for a specified day of the week

-X
Sort students by course number (for cross-listed class lists)

-y year
Select sections for the specified numeric year

-z
Output blank student information lines

By using various combinations of parameters, it is possible to produce lists for many different situations.

Examples of parameters passed to the program at the menu are in $CARSPATH/menuopt/regist/programs. All the files in this subdirectory that begin with "regl" pertain to REGLIST.

The first two parameters, year and session identify the course schedule to be used. The year is specified by "-y", followed by the numeric year value, and the session is "-s" followed by the session code from the sess_table as follows:

```
reglist -y 199X -s FA
```

Here, only the sections offered in the fall session of 199X will be used by REGLIST.

Using only these two parameters results in printing a class list for every section of every course in the Fall 199X schedule. To restrict the selection further, any combination of the other selection parameters may be used. For example, to create a class list for one particular section, the following could be used:

```
reglist -y 199X -s FA -C UG9X -N ENG100 -S 1
```

This would create one class list for section number "1" of the course "ENG100" in the "UG9X" catalog for the "FA" session of 199X.

To allow greater selection flexibility, the course number parameter may include a wild card comparison using the asterisk (*). If class lists were needed for every English course instead of just one section in the previous example, the following would be appropriate:

```
reglist -y 199X -s FA -N ENG*
```

The course number value of "ENG*" will match every course number that begins with "ENG". Therefore, lists would be generated for "ENG100", "ENG101", "ENG200", etc.

The wild card comparison may only be used at the end of the value to match. The value "*100", then, cannot be used to print lists for all "100" level courses. Since the asterisk must be at the end of the value, "*100" means the same as "*", which matches anything.

In addition to the course number values, wild card comparisons are also allowed on subsession, catalog, and section number values.

Note:

Refer to the *Course Entry Program* in the *Course/Class Schedule Technical Manual* for more information about cross-listed courses.

Form Definitions

Several FPS form definition files are provided with CARS Solution. The following list describes the characteristics of each form.

aclasslist
Double spaced with graduating students identified. No grades are listed. This list is for anonymous grading: does not include student names.

afinallist
Double spaced with grades and graduating students identified. This list is for anonymous grading: does not include student names.

agradelist
Double spaced with graduating students identified. This form provides underlines for the entry of grades. This list is for anonymous grading - does not include student names.

amhrslst
Positive attendance hours format.

amidlist
Midterm Grade List. Double spaced with grades and graduating students identified. This list is for anonymous grading: does not include student names.

classlist
Double spaced without grades printed and graduating students identified.

contactlst

Class Contact List.

finallist
Final Class List: double spaced with grades printed and graduating students identified.

gradelist
Double spaced without grades printed and graduating students identified. Underlines are provided to be used for entry of grades.

midlist
Midterm List: double spaced with grades printed and graduating students identified.

permroll
Permanent Class Rollsheet.

roster
Enrollment Verification Roster.

tmproll
Temporary Class Rollsheet.

waitlst
Class Wait List.

Any of these forms may be locally customized, or you may create new forms to meet your institution's requirements. Review these definition files for examples of the data fields you can use to customize your forms.

To select a particular form, the "-F" parameter must be given followed by the name of the form to be used.

Deciding which form to use depends on the purpose of the class lists. If it is the beginning of the session, a list may be created for every section of every course using the "tmproll" form as follows:

```
reglist -y 199X -s FA -F tmproll
```

If the selection of sections to be listed results in more than one list to be created, the lists will be sorted in course number, catalog code, then section number order. It may be desirable to have the lists sorted by the faculty members teaching the sections. If the "-f" parameter is used, the lists will be sorted by the abbreviated name for the faculty member of each section.

The following example uses this sorting option to create lists with blank grades to be distributed to the faculty members:

```
reglist -y 199X -s FA -F gradelist -f
```

After grades have been entered for students, it may be helpful to list all students who did not receive final grades. The "-m" parameter will allow only those students who have a final grade of "IP" (In Progress) or "NR" (Not Reported) to be selected. This option may be used with any of the other parameters.

If the wait list is being used in the REGENT program, it is possible to print lists of only those students who are waiting for a seat to become available in a section. Since wait lists are treated on a first come, first serve basis, a wait list should be sorted by the date and time the student was added to the list. This is accomplished using the "-w" parameter. Only students with a stat of "L" will appear on the lists.

REGLIST normally does not create blank lists. If it is necessary to create a list when no students appear on the list, the "-a" option may be used. This may be most helpful when a few sections are being selected, to avoid the possibility of receiving no output at all.

Note:

In the form definition files, the header field "date" will contain the system date and time that the REGLIST program was started. The scrolling field "underline" will contain five underscores (_____) and may be used to provide blanks for the entry of grades on the printed list. The scrolling field "no" will contain a sequential number for each student on a list, beginning at "1". The scrolling field "graduating" will contain an asterisk (*) if the year and session parameters match the planned graduation year and session fields in the prog_enr_rec. Otherwise, the field will contain a blank ( ).

Program Errors

The following lists the error messages that are generated by REGLIST.

Bad parameter: 'parameter'.

This message occurs when an invalid parameter option has been passed to REGLIST, or when the session or year parameter are left blank or zero. See the above *Parameters* section for valid parameter options.

FPS_OPEN error on reglist form

This error occurs when an invalid FPS form name has been passed with the -F parameter option. To validate and load forms, REGLIST checks the forms installed in the $FPSPATH/regist/reglist install directory. Verify that the form desired has been correctly installed through the make processor, and that the correct permissions are used on the directory and file.

Crash Recovery

In case the REGLIST program exits abnormally, watch the screen or review your electronic mail for any error messages from the program. After you correct the error, remove the FPS file and restart the program.

Follow these steps to restart REGLIST:

1. Type the following at the shell prompt to change to the forms directory.

   cd /usr/spool/forms

2. Type the following at the shell prompt to list in descending order the last modified time of in-progress class list FPS files.

   ls -lt formname.[fi]*

   where "formname" is the same as the -F parameter form.

3. Review the list and identify the in-progress FPS file (which is incomplete) that has the same time as the program crash. Remove the file.

4. Restart the REGLIST program, watching closely for any error messages if the crash is not due to system failure.

Database Input

The following database files are read by REGLIST:

- adm_rec
- classscan_rec (for grade scanning)
- crs_rec
- ctc_rec (for grade scanning)
- cw_rec
- cweval_rec (for grade scanning)
- fac_rec
- grdg_table
- id_rec

- mtg_rec
- profile_rec
- prog_enr_rec
- reg_rec
- sec_rec
- stu_acad_rec
- stuscan_rec (for grade scanning)
- svc_rec (for grade scanning)

Database Output

The grade scanning application updates various files.

Registrar - Using Registration No-Show

Introduction

The REGNOSHOW program creates a list of unconfirmed students and/or students without financial clearance and can optionally drop these students from their classes and set the student's registration status to "no-show" status. REGNOSHOW can assist in monitoring unconfirmed students and/or students without financial clearance for student billing purposes. The program also helps in creating available slots in classes for other students.

Procedures

You should run REGNOSHOW just after registration to produce a list of students who have not been confirmed. This list should be reviewed by the registrar and coordinated with Financial Aid, Student Accounts, and your institution's instructors to ensure that the list does not include any students who should have been confirmed, but were missed. If any students on the list need to be confirmed, use REGENT to confirm the students. You may also confirm students through Student Accounts if financial clearance constitutes clearance at your institution. REGNOSHOW can be run at the registrar's discretion to produce a list of students who are without financial clearance and/or not confirmed.

Once the list has been narrowed down to students who are considered "no-shows," REGNOSHOW can update all students on the list to "no-show" status and drop the students from their classes. The registrar should notify the Student Billing Office that the "no-show" students need to receive refunds.

REGNOSHOW is a background process that does not require operator interaction.

Parameters

REGNOSHOW requires parameters that provide academic year, academic session, and academic program. If multiple academic programs are in use at an institution, REGNOSHOW must be run separately for each program. The parameter options available to REGNOSHOW are shown below:

-b
: Check for both registration status and financial clearance (optional)

-f
: Check for financial clearance (optional)

-p prog
: Academic program code e.g. UNDG (required)

-q
: Do not print report (optional)

-r
: Check for registration status (optional)

-s sess
: Academic session code e.g. FA (required)

-u
: Update registration status to "no-show" (optional)

-y year
: Calendar year e.g. 199X (required)

The "-p", "-s", and "-y" parameters are required and must always be used with the program; the "-b", "-f", "-q", "-r", and "-u" parameters are optional.

The following examples show how to run REGNOSHOW to generate a list of unconfirmed students, i.e. students with a registration status of "R":

- regnoshow -p "UNDG" -s "FA " -y 199X
- regnoshow -p "UNDG" -s "FA " -y 199X -r

In the first example, REGNOSHOW would print an unconfirmed student roster for students in the "UNDG" program for the "FA" "199X" session. Since the "-r", "-f" and "-b" options were not passed, the "-r" option is assumed. The second example illustrates the use of the "-r" option. This causes REGNOSHOW to operate in the same manner as when no parameters are passed.

Examples of the parameters used to cause REGNOSHOW to update the registration status and drop students from their classes follows:

- regnoshow -p "UNDG" -s "FA " -y 199X -u
- regnoshow -p "UNDG" -s "FA " -y 199X -u -q

The first example drops courses and produces a list of students updated. The second example performs the update without printing a list of students updated (because of the"-q" parameter). Please note, that before the update option is used, the registrar should review the unconfirmed student list.

Following are examples of the use of the "-r", "-f" and "-b" options:

- regnoshow -p "UNDG" -s "FA " -y 199X -f
- regnoshow -p "UNDG" -s "FA " -y 1989X-r -f
- regnoshow -p "UNDG" -s "FA " -y 199X -b

If the "-f" option is given as in the first example above, , then only those students who have financial clearance of "N" will be listed. However, if you select the "-r" and the "-f" options, then students that are either unconfirmed with a registration status of "R" OR have financial clearance of "N" will be listed. The "-b" option provides a way to list only those students who are both unconfirmed (with status of "R") and also have financial clearance of "N". If the "-b" option is given with the "-r" and "-f" options, only the "-b" option will take effect.

Compilation Values

REGNOSHOW does not use any conditional compilation values or flags that can be changed at individual installation sites.

Program Flow

In the first processing step, REGNOSHOW finds unconfirmed students and/or students without financial clearance and processes each student one at a time. The stu_acad_rec is searched for records where:

- the yr equals the "-y" parameter value
- the sess equals the "-s" parameter value
- the prog equals the "-p" parameter value
- the reg_stat is not "N"

Once a stu_acad_rec with the given parameters is found, the program determines if the student should be counted as a no-show according to the parameters that have been passed. The student is counted as a no-show if:

- the "-b" option is passed, and the reg_stat is "R" and fin_clr is "N"
- the "-r" option is passed and the reg_stat is "R"
- the "-f" option is passed and the fin_clr field is "N"

If the "-q" option was not used, the program locates the id_rec for the student so that the name and social security information can be printed on the report.

In the second processing step, the program locates each cw_rec for the student ID number, session, year, and program. The number of courses that the student is registered for and waiting for are accumulated for printing on the report. The registered course count is incremented for each cw_rec with a stat equal to "R", and the wait list course count is incremented for each cw_rec with a stat equal to "L".

If the "-u" parameter is passed to the program during the third processing step, the program updates each cw_rec for the unconfirmed student so that the stat equals "D" (dropped). A reg_rec is then added to record the drop. As each cw_rec is found, the sec_rec for the course is updated to decrease the wait list and registered counts according to the stat ("L" = waiting, "R" = registered). After each course has been dropped, the reg_stat changes to "N", and the reg_upd sets to the current system date.

In the final processing step, if the "-q" parameter is not used, unconfirmed students and/or students without financial clearance are printed to the audit report. This report contains name, social security number, current registration status, financial clearance flag, and the registered and wait list course counts.

Program Errors and Mail Messages

Although the program does send mail which indicates that the program has completed, program errors are printed on the standard output. The most common program usage error is as follows:

```
Usage:  regnoshow -p prog -y year -s sess [-u] [-q] [-r] [-f] [-b]

where:  prog: Academic program code
               year: Academic year
               sess: Academic session code
               -u: Update registration status to 'no-show
               -q: Do not print out no-show list
               -r: Check registration status for no-shows
               -f: Check financial clearance for no-shows
               -b: Check both financial and registration status for  no-shows
```

If you receive this message, it indicates that an invalid parameter or parameter combination was used when running REGNOSHOW. See the "Parameters" section above for valid parameters. After you correct the parameters, rerun the program.

When you use REGNOSHOW, you receive either of two kinds of mail messages. One is the usage message listed above, while the other is a message which indicates that the program has completed. The number of "no-shows" and the number of errors that occurred are indicated in this mail. This message could appear as follows:

```
               Regnoshow completed successfully with 10 noshows and 0 errors
```

Crash Recovery

If REGNOSHOW exits unexpectedly, rerun the program after the cause of the error has been resolved. If REGNOSHOW was running using the "-u" parameter, the students not updated from the previous run will be updated.

If you ran REGNOSHOW using the "-u" parameter, run REGAUDIT to verify that the sec_rec registered and wait list counts are correct.

Database Input

REGNOSHOW uses the following database files as input:

- cw_rec
- id_rec
- reg_rec
- sec_rec
- stu_acad_rec

Database Output

REGNOSHOW updates or adds the following database files:

- cw_rec
- reg_rec
- sec_rec
- stu_acad_rec

REGNOSHOW Report Output

On the No Show Report, , the "Reg Stat" (registration status) column represents the current value in the reg_stat field for each student listed. Both the "R" for "registered" and the "C" for "confirmed" appear in the "Reg Stat" column if you select the "-r" and "-f" parameter options.

Even though a particular student may have a status of "C", he or she may not have financial clearance for one reason or another. The "-f" option tells REGNOSHOW to list any students who do not have financial clearance. The "-r" option tells REGNOSHOW to list any students who are not confirmed.

If you select the "-b" option, the output contains only students who are both unconfirmed and without financial clearance.

If you select the "-u" parameter option, the "Reg Stat" field on the report displays the registration status of each student prior to the "no-show" update of "N" in the stu_acad_rec._reg_stat field.

The audit reports generated by REGNOSHOW also indicate which parameters you selected at run time.

Student Billing - Automated Holds Script

Purpose

The purpose of the Autoholds script is to create new holds and re-activate or terminate existing holds. The script re-activates holds by removing the ending date and changing the beginning date to the date the script is run. To terminate holds, the script places an ending date in the record.

Before you use this script

All subb_recs need to be forwarded to the most current subb_rec. If any subb_rec remains under the Balance Amount indicated in the Autohold process, an ending date will be assigned to the hold_rec. Credit balance and zero balance subb_recs will be ignored by the script.

Menu Option

The menu option for this script is Automatic Holds - SUBS, found in the Fiscal Management: Student Accounts Billing: Student Billing Menu. The directory path for the script is $CARSPATH/modules/stubill/scripts/autoholds.

Student Billing - Data Flow Diagram

Overview

The area of Student Accounts includes the Student Billing, Financial Aid, Registration, Voucher, and Statement applications programs as well as some table maintenance and reporting.

Explanation of Diagram

The following diagram outlines all the functions involved in maintaining student accounts. Processes such as REGENT, labeled in all caps, are application programs that cause the next step to occur. Each process is underlined. Subprocesses are listed under a process. For example, "Assessments" is a process. "Tuition" is a subprocess of "Assessments". "BILLING" is the application program which assesses charges against a student.

Overview Of The Student Accounting Process

```
                            Assessments
                            -----------
        Tuition     Course Fees         Room              Student Activity Fee
                                        Board             Health Insurance
          |             |                  |                          |
          |             |                  |                          |
          v             v                  v                          v
          ------------------------------------------------------------
                    |
                    |
                    | (to Charges)
```

```
              Charges            Financial Aid Awarded
              -------            ---------------------
                 |                         |
              BILLING                FAENTRY -------------|
                 |                         |              |
                 v                         v              |
                 ---------------------------              |
                                        |                 |
                                        |                 |
                                        v                 |
 SDS <---------- REGENT/BILLING <----Student Billed       |
                        |            --------------       |
                        |                                 |
                        |                                 |
                        v                                 |
                        -------------------|              |
                                           v              v
Add Additional Charges              Post SB Voucher   Post FA Voucher
----------------------              ---------------   ---------------
     |                                     |                 |
     |                                     |                 |
    SA                                    SB                FA
  VOUCHER                                VOUCHER          VOUCHER
     |                                     |                 |
     |                                     |                 |
     v                                     v                 v
     ---------------------------------------------------------
                        |
                        v
           Subsidiary Total and Balance Records
           ------------------------------------
                        |
                        |
            CH          |
          VOUCHER       |
  Payments ---------->  |
  Posted                |
  ------                |
                        v
              Update Balance Records
              ----------------------
                        |
                        |
                        |
                        v
                Monthly Statements ------STMT----------> Subsidiary Account
                ------------------                           Statement
                                                         ------------------
```

Automatic Charges
Students may be assessed charges automatically depending upon their Registration, Student Services, Program Enrollment, Profile, or ID record. The charges are based upon values maintained in those records. The conditions under which those charges are created are found in the Assessment table. Records in the Charge table place dollar amounts against the assessments. The application program, BILLING, assesses charges for each student for a particular billing run. The program, REGENT, through its 'Output' command, produces a Student Data Sheet which lists each of the charges assessed as student as a result of BILLING.

Assessments
Assessments, as was mentioned previously, can be created because of certain values in one of five student records. These assessments fall into four general categories, tuition, course fees, room and/or board, miscellaneous charges. Tuition is usually assessed based on the number of credit hours a student is registered for during a specific academic session. Course fees are assessed against a student if he/she is registered for a course with a fee code assigned to it. Room or dorm fees may be assessed based on the dorm code, the number of individuals within the dorm room, or if the student is not a resident assistant within a dorm. Meal plan types determine the fees assessed for board charges. Student activity and health insurance fees are two of the most common miscellaneous assessments.

Charges
Each record in the Assessment table relates to one or more record in the Charge table. BILLING looks at both the Assessment and Charge tables to determine which charges are to be assessed against a student and then the amount of the charge based on the following equation:

```
Charge = Minimum charge + Rate * ((UNITS per student - offset) / step)
```

Billing
BILLING can be used for testing, to see if a student will be billed for everything they are supposed to be billed for. The 'Process' command within BILLING processes bills for each student meeting the conditions built into the assessment and charge records for the billing run period. Another command, 'Verify', will provide a printout of each student billed in that run and the charges each was assessed. A student billing report, 'Student Charges', will provide similar data for the student accounts office.

Financial Aid
There are two functions within the financial aid area, determining a student's financial aid need, and awarding the aid to the student. Both of these functions can be accomplished through the program, AIDENTRY.

Financial Aid Need
A student's financial aid need is determined for the entire academic calendar year. The values maintained in the Financial Aid Need record for the student are displayed on the screen used for aid entry so that the operator is continually informed of what portion of the student's budget is need and what percentage of that need is unmet.

Awarding Aid
Although financial aid is awarded for specific sessions, the encumbered amounts for each session of the academic year can be entered at the same time.

Student Billing
The Student Bill is the initial notification to the student of all the charges assessed against the student, the financial aid awarded, and the total amount owed. A Subsidiary Account Statement is a periodic notification to the student of the status of the student's account; i.e., amount paid since last statement, amount owed to the institution or the amount owed to the student.

For example, a student might receive a bill for the Fall session in August. Once he/she has paid part or all of the total amount, he/she might receive monthly statements on the status of his/her account balance.

Printing a Student Bill
The Student Data Sheet (SDS) with the billing data included is considered the student bill. It is created by REGENT which in turn calls BILLING to find all charges and aid for a student. The SDS can be printed for an individual student interactively through REGENT, at the time of registration, or for a group of selected students through the Forms Production System. Those institutions which allow the student to pay towards the bill soon after registration, may prefer to print the SDS directly from REGENT when the student's registration is displayed on the terminal screen. Other institutions mail bills to all students several weeks prior to the beginning of classes. A version of REGENT is available to all in the latter situation which will create SDS's for all students within a specific academic program. The REGENT and FPS documentation further describe the above mentioned processes.

Posting Charges Against a Student's Account
To post the necessary Subsidiary Total and Balance records against a student's account, the student accounts operator must run the program VOUCHER for voucher type 'SB'.

Posting Additional Charges Against a Student's Account
An SA or student account voucher is used to post miscellaneous charges or payments against a student's account. Examples of such may be dorm damage charges or a credit balance forward. This

same voucher may be used to query the student account, financial aid, key deposit, or room deposit subsidiary accounts for a particular student.

Updating Subsidiary Accounts
SB and SA vouchers are used to create and/or modify subsidiary total and balance records that influence a student's account. Payments entered through a CH or cashier voucher will also update the appropriate S/A (student accounts), K/D (key deposit), R/D (room deposit) subsidiary balances.

Periodic Subsidiary Account Statements
Subsidiary Account Statements are created through the program, STMT, and printed through the Forms Production System. As was mentioned previously, statements are to be sent to students on a periodic basis after the initial bill has been sent.

Student Billing - How to Calculate Minimum Payment

Overview
This document explains in detail the calculations that STMT does in computing minimum payments, past amounts, and current amounts due.

Introduction
In setting up the Deferred Payment tables, it is useful to know exactly what STMT does when it computes minimum payments.

The Defpay Table
The Deferred Payment table, in conjunction with the Payment Term table (pay_term_table), controls the computation of deferred payments for any given subsidiary balance. If, for example, the pay_term_table entry for "DEFER " has specified "INCR" for pterm_defpay, the computations will be controlled by the INCR entries in the defpay_table.

‘tdefpay_tbcode' is the code for this deferred payment plan.

‘tdefpay_mo' is either the number of months from the subb_due_date this record will become effective, or a flag that specifies that this record will become effective in a certain number of days.

‘tdefpay_days' is either the day of the month this record is to become effective, or the number of days from the subb_due_date.

‘tdefpay_pct' is the percentage of the balance that is due on the date specified by tdefpay_mo and tdefpay_days.

‘tdefpay_min' is the minimum payment allowed. If the calculations give a number less than this minimum, the minimum (or the remaining balance due, if the balance is less than the minimum) is due.

‘tdefpay_inc' is the incremental amount or minimum amount that can remain due in the account. If the payment computed by tdefpay_pct and tdefpay_min would leave less than this amount in the account, the entire balance is due. Often, small balances just aren't worth the time and effort needed to collect them.

‘tdefpay_inc' helps prevent the formation of small balances.

Sample Deferred Payment Table
The sample table below is an example of a Deferred Payment table that will be used in subsequent examples:

	tdefpay_tbcode	tdefpay_mo	tdefpay_day	tdefpay_pct	tdefpay_min	tdefpay_inc
1.	INCR	-1	0	25	$25.00	$10.00
2.	INCR	1	31	50	$25.00	$10.00
3.	INCR	2	31	75	$25.00	$10.00
4.	INCR	3	31	100	$0.00	$0.00
5.	INCR	4	31	100	$0.00	$0.00

The first table entry with the tdefpay_mo of -1 and tdefpay_day of 0 is designed to produce a minimum payment of 25% on the subb_due_date. The next three entries make successively higher percentages due on the last day of the next three months, until the entire balance is due on the last day of the third month. Note that the tdefpay_min and tdefpay_inc fields are zero for the fourth defpay_table entry; it does not matter what they are, as 100% of the balance is due at that time.

The last entry in the table may seem confusing, but it serves the purpose of determining when the total amount should be considered past due. Without the last entry, STMT has no way of knowing when to move the last 25% of the balance due from "current due" to "past due."

STMT uses the program run date, instead of the statement end date, to determine the minimum payment due. If, for example, the program was run on October 5 with a statement beginning date of September 1 and a statement ending date of September 30, any charges, credits, or financial aid to that individual's account on October 1st through 5th would not show up on the statement but would still be taken into account when computing the minimum payment.

The minimum payment computations do not easily handle a mix of balances without minimum payment plans and balances with minimum payment plans. The outstanding balance from any non-minimum payment balance will be added to the past due amount for those that have a minimum payment plan.

Month and Day Example

Let us assume that the subb_due_date is 08/30/86, the first day of the fall semester. Students register for the fall classes, and their fall balances are automatically added with the 8/30/86 due date. Defpay_table entry #1 will produce a statement showing 25% of the balance due on August 30. (Special month code of -1 indicates number of days from the subb_due_date, and days value of 0 indicates the subb_due_date.) On September 30, the second table entry becomes effective. 50% of the balance is due, and if the student has not paid the first 25% of his bill, it is considered past due. 75% of the balance is due on October 31 (with the first 50% being past due if it was not paid), and the rest of the balance (100%) is due on November 30. On December 31, anything the student has not paid becomes past due.

Computation Description

Below is a description of the computations for the minimum payment.

Formulas

Two table entries are used in the computations. The first table entry, used to compute the minimum payment, is the table entry that matches the current date. The second table entry used is the one immediately before the one used to compute the minimum payment. It is used to compute the portion of the balance that is past due.

```
1.  Compute the minimum due based on percentage:
    Due = ((Charges - Financial Aid) * % from table) - Payments
2.  Check the Minimum Payment Due:
    If Due <= Min payment from table then
        Due = Minimum Payment
    Note that if payments result in a credit balance in step 1, no
    payment is due, and the credit amount will be applied to other
    sessions.
3.  Incremental check:
    If ((Charges - Aid - Payments) - Amount Due <= Increment value then
        amount due = Charges - Financial Aid - Payments
```

Example of Calculations

```
Key for diagram:
BF - Balance Forward From Previous Month
MO - Activity in Current Month
C  - Charges
P  - Payments
F  - Financial Aid Posted
Terms of Payment:
SP86 and FA86:  Both Bal periods use a DEFER code which is linked as
                described above to the deferred payment code of INCR.
                The payment due date for each BAL period is:
                SP86 - 01/05/86
                FA86 - 08/30/86
```

		July.		Aug.		Sept.		Oct.	
		BF	MO	BF	MO	BF	MO	BF	MO
	C	200 a		200 b					
SP86	P				-200 b				
	F								
	C			0	2000 b	2000 c	200 c	2200 d	
FA86	P			0	-300 b	-300 c	0 c	-300 d	-725 d
	F		-100 a	-100 b	-50 b	-150 c	0	-150 d	

This example covers two fiscal calendar periods. The first period, SP86**,** had a previous balance of $200, which was eventually paid off in July. The second period, FA86, had $100 in financial aid credited to the account in July. Charges totaling $2000 were added in August. Payments totaling $300 and financial aid of $50 was also credited to the student's account in August. Another $200 in charges was made in September, and finally, the student made a $725 payment in October. November is not shown; there was no activity to the account.

```
a.) July Statement: July 31, 1986
     Minimum Payment Calculation:
        SP86  -
            Formula 1: ((200 - 0) * 100.00%) - 0 = 200
            Formula 2: Is (200 <= 25)? Due = 200
            Formula 3: Is ((200 - 0 - 0) - 200) <= 10)? Due = 200
         FA86 -
            Formula 1: ((0 - 100) *  0.00%) - 0 = 0
     Past Due Minimum Payment Calculation:
        SP86  -
            Formula 1: ((200 - 0) * 100.00%) - 0 = 200
            Formula 2: Is (200 <= 25)? Past Due = 200
            Formula 3: Is ((200 - 0 - 0) - 200) <= 10)? Past Due = 200
         FA86 -
            Formula 1: ((0 - 100) *  0.00%) - 0 = 0
     Statement Ending balance would show 100.00 dr
     Past Due    Current Due    Minimum Payment
      200.00           0.00          200.00
```

Note that although the ending balance only gives the student a $100 debit, he still owes $200 on his previous balance. The financial aid he received in July cannot be credited toward his Spring balance, and the entire Spring balance is considered past due.

```
b.) August Statement: August 31, 1986
    Minimum Payment Calculation:
        SP86  -
            Formula 1: ((200 - 0) * 100.00%) - 200 =   0
        FA86  -
            Formula 1: ((2000 - 150) * 25.00%) - 300 =   162.50
            Formula 2: Is (162.50 <= 25)? B = 162.50
            Formula 3: Is ((2000 - 150 - 300) - 162.50) <= 10)? A = 162.50
    Past Due Minimum Payment Calculation:
        SP86  -
            Formula 1: ((200 - 0) * 100.00%) - 200 =   0
        FA86  -
            Formula 1: ((2000 - 150) * 0.00%) - 300 =   0
     Statement Ending Balance would show 1550.00 dr
     Past Due    Current Due    Minimum Payment
        0.00       162.50          162.50
```

Note that although there is a debit balance of $1550, the student only owes the minimum payment of $162.50.

```
c.) September Statement: September 30, 1986
    Minimum Payment Calculation:
        FA86  -
            Formula 1: ((2200 - 150) * 50.00%) - 300 =   725.00
            Formula 2: Is (725.00 <= 25)? B = 725.00
            Formula 3: Is ((2200 - 150 - 300) - 725.00) <= 10)? A = 725.00
    Past Due Minimum Payment Calculation:
        FA86 -
            Formula 1: ((2200 - 150) * 25.00%) - 300 =   212.50
            Formula 2: Is (212.50 <= 25)? B = 212.50
            Formula 3: Is ((2200 - 150 - 300) - 212.50) <= 10)? A = 212.50
     Statement Ending Balance would show 1750.00 dr
     Past Due    Current Due    Minimum Payment
      212.50        512.50        725.00
```

Note that the past due increased by $50 because of additional charges totaling $200 in September. The minimum payment calculations do not consider when a given charge was added to the account. All charges made to an account for a given period are due as of the due date for that balance period, regardless of the date it was actually added to the account. So although the $200 charge was added sometime in September, the calculations still use the August 31 date for computing the minimum payment.

```
d.) October Statement: October 31, 1986
    Minimum Payment Calculation:
        FA86  -
            Formula 1: ((2200 - 150) * 75.00%) - 1025 =   512.50
            Formula 2: Is (512.50 <= 25)? B = 512.50
            Formula 3: Is ((2200 - 150 - 1025) - 512.50) <= 10)? A = 512.50
    Past Due Minimum Payment Calculation:
        FA86 -
            Formula 1: ((2200 - 150) * 50.00%) - 1025 =   0
     Statement Ending Balance would show 1025.00 dr
     Past Due    Current Due    Minimum Payment
        0.00       512.50          512.50
```

The payments made in October paid the past due and the September amounts in full.

```
e.) November Statement: November 30, 1986 (NO TRANSACTIONS IN MONTH)
    Minimum Payment Calculation:
        FA86  -
            Formula 1: ((2200 - 150) * 100.00%) - 1025 =   1025.00
            Formula 2: Is (1025.00 <= 25)? B = 1025.00
            Formula 3: Is ((2200 - 150 - 1025) - 1025.00) <= 10)? A = 1025.00
    Past Due Minimum Payment Calculation:
        FA86 -
            Formula 1: ((2200 - 150) * 75.00%) - 1025 =   512.50
            Formula 2: Is (512.50 <= 25)? B = 512.50
            Formula 3: Is ((2200 - 150 - 1025) - 512.50) <= 10)? A = 512.50
     Statement Ending Balance would show 1025.00 dr
     Past Due    Current Due    Minimum Payment
      512.50       512.50         1025.00
```

Student Billing - How to Prepare for a New Billing Session

Overview

Preparation for a new billing session is made simpler with the aid of several INFORMER Adds to create new records.

Introduction

Before BILLING can be run for a new session, the appropriate Billing Parameter, Assessment, Charge, and Refund records are needed. Also, the new billing period needs to be included in the fiscal calendar. Subsidiary Balance and Total records are also needed for the new session. Records created each session that vary little between sessions, such as the stu_serv_recs, can be added using an INFORMER Add.

Fiscal Calendar

The new billing period must be in the fiscal calendar. It is recommended that the records for each billing period be created as soon as the dates are known. This procedure will need to be accomplished using the PERFORM data entry screen for the fiscal calendar.

Billing Tables

Where many records are added to a file or table, an INFORMER will make the initial addition process faster. PERFORM will be needed to make any adjustments, additions, deletions, updates.

Billing Parameter File

Any new Billing Parameter records for the new session should be entered using the PERFORM data entry screen for the billing parameter file.

Assessment and Charge Tables

Because most of the Assessment and Charge records will be duplicated each term, INFORMER Adds exist to create the new entries in the tables from a previous session. The following is the INFORMER.

```
prompt for cursess using "Enter Current Session Code >> ";
prompt for curyr using "Enter Current Year (eg 1984) >> ";
prompt for newsess using "Enter New Session Code >> ";
prompt for newyr using "Enter New Year (eg 1985) >> ";
read into x assess_table
     where assess_sess = cursess
     and assess_yr = curyr;
add noprompt assess_table
assess_sess = newsess
assess_yr = newyr
assess_code = x.assess_code
assess_unit = x.assess_unit
assess_prog = x.assess_prog
assess_subsess = x.assess_subsess
assess_desc = x.assess_desc
assess_billing = x.assess_billing
```

```
assess_crit_file = x.assess_crit_file
assess_crit_field = x.assess_crit_field
assess_crit_stat = x.assess_crit_stat
assess_crit_rel_op = x.assess_crit_rel_op
assess_pcc_file = x.assess_pcc_file
assess_pcc_field = x.assess_pcc_field
assess_pcu_file = x.assess_pcu_file
assess_pcu_field = x.assess_pcu_field
assess_pbc_file = x.assess_pbc_file
assess_pbc_field = x.assess_pbc_field
assess_pbu_file = x.assess_pbu_file
assess_pbu_field = x.assess_pbu_field
assess_scc_file = x.assess_scc_file
assess_scc_field = x.assess_scc_field
assess_scu_file = x.assess_scu_file
assess_scu_field = x.assess_scu_field
assess_sbc_file = x.assess_sbc_file
assess_sbc_field = x.assess_sbc_field
assess_sbu_file = x.assess_sbu_file
assess_sbu_field = x.assess_sbu_field
assess_rfnd_file = x.assess_rfnd_file
assess_rfnd_field = x.assess_rfnd_field
assess_rfnd_stat = x.assess_rfnd_stat
assess_rfnd_rel_op = x.assess_rfnd_rel_op
assess_date_file = x.assess_date_file
assess_date_field = x.assess_date_field
assess_rfnd_schd = x.assess_rfnd_schd
;
read into y chg_table
        where chg_sess = cursess
        and chg_yr = curyr;
add noprompt chg_table
chg_sess = newsess
chg_yr = newyr
chg_code = y.chg_code
chg_unit = y.chg_unit
chg_prog = y.chg_prog
chg_subsess = y.chg_subsess
chg_desc = y.chg_desc
chg_br1 = y.chg_br1
chg_br2 = y.chg_br2
chg_br_code1 = y.chg_br_code1
chg_br_code2 = y.chg_br_code2
chg_br_code3 = y.chg_br_code3
chg_eng1 = y.chg_eng1
chg_max1 = y.chg_max1
chg_eng2 = y.chg_eng2
chg_max2 = y.chg_max2
chg_eng3 = y.chg_eng3
chg_max3 = y.chg_max3
chg_incr = y.chg_incr
chg_min1 = y.chg_min1
chg_rate1 = y.chg_rate1
chg_offset1 = y.chg_offset1
chg_step1 = y.chg_step1
chg_min2 = y.chg_min2
chg_rate2 = y.chg_rate2
chg_offset2 = y.chg_offset2
chg_step2 = y.chg_step2
chg_min3 = y.chg_min3
chg_rate3 = y.chg_rate3
chg_offset3 = y.chg_offset3
chg_step3 = y.chg_step3
;
b
```

Any modifications to either of these tables, updates, deletions, additions, can be made using the PERFORM data entry screens for these tables. If any new records are added to the Charge table, the codes should also be added to the Subsidiary Total table. The General Ledger account numbers for any new Subsidiary Total records should be verified with the G/L Account record. An ACE report, 'subwogl', can be used for the verification process.

Refund Table
Because the entries in the Refund table are date dependent, it is recommended that any additions to the table be made through PERFORM. Query the codes for each program first. Then use the Control 'P' function to duplicate data from some of the fields during the Add.

Subsidiary Balance and Total Records
INFORMERS exist to add new Subsidiary Total records for subt_code "PAID" and to add new Subsidiary Balance records for the "SB" subb_code. Any subsidiary periods to be used must be in the fiscal calendar first. The following INFORMER Adds are used to create the subt_recs and subb_recs.

```
prompt for oldprd using "Enter Current Subsidiary Period (eg FA84) >> ";
prompt for newprd using "Enter New Subsidiary Period (eg SP85) >> ";
read into x subb_rec
    where subb_subs = "S/A "
    and subb_code = "SB"
    and subb_prd = oldprd
;
add noprompt subb_rec
subb_subs = x.subb_subs
subb_code = x.subb_code
subb_prd = newprd
subb_subs_no  = x.subb_subs_no
subb_date = today
subb_pay_tms  = x.subb_pay_tms
subb_stat = "O"
;
read into y subt_rec
    where subt_subs = "S/A "
    and subt_code = "PAID"
    and subt_prd = oldprd
;
add noprompt subt_rec
subt_subs = y.subt_subs
subt_code = y.subt_code
subt_prd = newprd
subt_subs_no = y.subt_subs_no
subt_date = today
subt_stat = "O"
;
b
```

Student Service Records
New Student Service records are needed for each student. The following INFORMER will create new records based on the existing ones. Any changes in housing, meal plans, or any other billable field will need to be made using PERFORM.

```
prompt for cursess using "Enter Current Session Code >> ";
prompt for curyr using "Enter Current Calendar Year >> ";
prompt for newsess using "Enter New Session Code >> ";
prompt for newyr using "Enter New Calendar Year >> ";
read into x stu_serv_rec
    where stusv_sess = cursess
    and stusv_yr = curyr
;
add noprompt stu_serv_rec
stusv_sess = newsess
stusv_yr = newyr
stusv_id = x.stusv_id
stusv_rsv_stat  = x.stusv_rsv_stat
emergency_phone = x.emergency_phone
local_phone = x.local_phone
off_campus_res_appr = x.off_campus_res_appr
intend_hsg = x.intend_hsg
campus  = x.campus
dorm = x.dorm
room = x.room
suite   = x.suite
no_per_room = x.no_per_room
asb_fee_wvd = x.asb_fee_wvd
campus_box = x.campus_box
late_reg = x.late_reg
health_ins_wvd  = x.health_ins_wvd
meal_plan_type  = x.meal_plan_type
meal_plan_wvd = x.meal_plan_wvd
res_asst = x.res_asst
park_prmt_no = x.park_prmt_no      (** If new permits are issued each term
veh_type = x.veh_type              (** do not include these three
veh_license = x.veh_license        (** fields
stusv_stat = "I"
;
b
```

If it is known that any student will not be returning for the new term, delete the student's stu_serv_rec for the new term. If the field 'stusv_stat' is being maintained, the where clause might contain the following:

```
     and stusv_stat <> "N";
```

Student Billing - How to Use Student Billing Tables

Overview

Several database files are used by the Student Billing process to automate student charges. This document describes each file and how it is used by the Student Billing process.

Introduction

The Student Billing program, BILLING, uses information entered into several database files to calculate student charges based on registration and student services information. This section explains each file and identifies how each database field is used by the BILLING program in calculating charges.

Billing Parameter Record (pbilling_rec)

The Billing Parameter record defines several parameters required by the BILLING program to calculate student charges. This information determines the academic program and session for which charges are to be calculated and identifies posting information. The INTEREST program also shares this file for obtaining information used in calculating interest charges on past due account balances.

The following shows the format of the PERFORM screen used to enter information into the Billing Parameter record and a description of each field.

```
                              BILLING Parameters
Billing Parameter.....[a   ][aa                    ] Session.....[i   ]
Program...............[b   ]                         Year........[j   ]
Billing Code..........[d   ]                         Subsession..[k ]
Bursar Calc...........[c]
Subsidiary Code.......[e   ]
Subsidiary Entry......[f   ][ff                    ]
Bal Code..............[g   ]
Bal Period............[h     ]
```

Billing Parameter - (pbill_run_code)
Identifies a unique billing run code used to reference information in a Billing Parameter Record. This value is used by the BILLING program to look up and read information in a Billing Parameter Record. The field [aa] (pbill_desc) is used to enter a long description that describes the run code.

Program - (pbill_prog)
Identifies the academic program, i.e., "UNDG" that is to be billed. This field cannot be blank; therefore, there must be a separate Billing Parameter Record for each academic program to be billed.

Bursar Calc - (pbill_bursar_calc)
Is a "Y" or "N" value that allows charge or financial aid adjustments to be calculated for display in the BURSAR program. If a "N" value is used, the indicated session in the record can be displayed by the BURSAR program as long as there are posted charges or credits, with the exception that charge and financial aid adjustments will not be calculated. A "Y" value allows charge and aid adjustments to be calculated and displayed in the BURSAR program. This field allows the bursar to turn off the automatic calculation of billing adjustments for sessions where additional billing charges are not to be posted.

Billing Code] - (pbill_bill_code)
Is used as one of the links into the Assessment record. Generally, this field is either "SB" for Student Billing or "INT" for Interest Parameters. The BILLING program uses this field along with the session, year, program, and subsession, to join to the Assessment table for lookup purposes.

Subsidiary Code - (pbill_subs)

Identifies the student accounts subsidiary (e.g., "S/A") where student charges are to be posted. This must be a subsidiary that uses balance (subb_rec) and total (subt_rec) records.

Entry Type - (pbill_ent_type)
Is the entry code (e.g., "BILL") used in posting student billing charges. This code must be a valid entry code according to the ent_table.

Subsidiary Entry Desc - (pbill_sube_desc)
Becomes the General Ledger Entry record description when charges are posted to student accounts. This description is the text that is printed on student accounts statements as the heading for student billing charges.

Bal Code and Bal Period - (pbill_bal_code, pbill_bal_prd)
Are used to determine the Subsidiary Balance record to post charges to. The "Bal Code" (e.g., "SB") identifies the type of Subsidiary Balance record, and the "Bal Period" identifies the balance period to post charges to. Since the "Bal Period" and the academic session fields contain different values, this field is used by BILLING to link academic session codes to balance periods.

Subsession - (pbill_subsess)
Identifies the subsession code to be billed. This field should be left blank unless it has been determined to set up totally separate billing runs for each subsession within a session. Generally, this field is left blank, and subsessions are billed together in one billing run for a session.

Year - (pbill_yr)
Contains the academic calendar year (format 19XX) where the session to be billed occurs.

Session - (pbill_sess)
Identifies the academic session code for the session to be billed.

Assessment Table (assess_table)
The Assessment table defines criteria used to determine if a student is to be charged and defines what information the charge is to be based on. Assessment tables must be set up for each session and academic program where automated student billing is to be done.

Before the Assessment table is explained in detail, the function of the Assessment and Charge tables in automated student billing must be understood. The Charge table explained in more detail below is used to determine the dollar amount of a charge, while the Assessment table determines the charges that apply to a student and part of the information used in the Charge table to calculate the amount.

The BILLING program calculates charges to be posted by first loading all the Assessment tables that join to the Billing Parameter record selected. The join is based on the fields in each record for session, year, subsession, and bill code (e.g., "SB"). Then, based on information within the Assessment table and on information found in other files referenced by the Assessment table, a charge code is selected. If the charge code is found matching the session and academic program of the billing run, the dollar amount is then calculated.

The following shows the format of the screen used to enter assessments and describes each field in the Assessment table.

```
                                ASSESSMENT TABLE
Code.............[a   ][b                       ]            Program......[c    ]
Billing Code.....[d   ] Session...[e   ]  Year........[f   ] Subsession...[g ]
Primary Test 1.......[h].[i   ].[j   ]    Primary Charge Code.............[pa  ]
[k                                   ]    [pb                                  ]
Primary Test 2.[l  ].[m].[n   ].[o   ]    Primary Charge Units............[pc  ]
[p                                   ]    [pd                                  ]
Primary Test 3.[q  ].[r].[s   ].[t   ]    Primary Break Code..............[pe  ]
[u                                   ]    [pf                                  ]
Primary Test 4.[v  ].[w].[x   ].[y   ]    Primary Break Units.............[pg  ]
[z                                   ]    [ph                                  ]
                                          Secondary Charge Code...........[sa  ]
----------REFUND INFORMATION----------    [sb                                  ]
                                          Secondary Charge Units..........[sc  ]
Refund Conditions...............[ra  ]    [sd                                  ]
Refund Date.....................[rb  ]    Secondary Break Code............[se  ]
[rc                                  ]    [sf                                  ]
Refund Schedule.................[rd  ]    Secondary Break Units...........[sg  ]
as of Date.. 00/00/00    Pct..      0     [sh                                  ]
```

Assessment Identification Fields
The identification section of the Assessment table are fields used to link the Assessment table to the Billing Parameter record and the Charge table. A description of each of these fields follows:

Code - (assess_code)
Identifies a four-character code used to reference an Assessment Table. The value in this field is not used as a direct link to the Charge Code. It is a value used simply to reference an Assessment table and must be unique within each session, subsession, and program.

Long description - (assess_desc)
Used to explain the purpose of information within an assessment

Program - (assess_prog)
Identifies the academic program code to be billed. This field must not be blank. Each academic program must be billed separately.

Bill Code - (assess_billing)
Identifies a value that is used to link from the Billing Parameter record to the Assessment table along with the session, year, subsession, and program code. If this field contains any value defined in the macro 'INT_BILL_CODE' in the file 'CARSPATH/include/custom/billing', the BILLING program will calculate interest charges instead of student billing charges.

Session - (assess_sess)
Identifies the academic session to be billed.

Year - (assess_yr)
Is the calendar year (format 19XX) where the session occurs.

Subsession - (assess_subsess)
Identifies the subsession within a session to be billed. Generally, this field is left blank since all subsessions within a session may be billed at the same time.

Functions of the Primary and Secondary Sections
Before each field in this section is described, the function of the primary and secondary sections must be explained. Basically, the primary and secondary sections correspond to an "if then else" statement; or, in other words, "if test conditions are true, then use primary side, else use secondary side". If the conditions presented in the Primary Test 1 and/or Primary Test 2 and/or Primary Test 3 and/or Primary Test 4 fields are met, the primary side is used and the secondary side is ignored. If the conditions listed in the Primary Test fields are not met, the primary side is ignored and the secondary side is used.

Each section is made up of fields from a set of special codes that reference other database files, accumulations of courses or hours (audit, registered, or total) or constant values. When the BILLING program reads an assessment, these values are translated by information already in the database for a student. Once the values have been found, the primary test is evaluated. If the condition is met, the primary side is then used to determine the charge. If the condition fails, the secondary side is used to determine the charge.

It is possible to leave an entire primary or secondary section's fields blank if the section is not needed. However, be sure to fill in all required fields when using a section. The Charge Code and Charge Units fields must be filled in along with only one of the Break Code or Break Units fields.

Primary Test Fields
Primary Test 1 [h - k] and **Primary Test 2 [m - p]** and **Primary Test 3 [r - u]** and **Primary Test 4 [w - z]** are the fields that make up the "if" test for the primary section. Up to four test conditions can be specified. If only one test condition is needed, use Primary Test 1 fields and leave the Primary Test 2, 3 and 4 fields blank. If more than one condition is needed, they can be logically related by using "AND" or "OR" in the test. Using "AND" requires that the Primary Test conditions be met to use the primary parameters [pa - ph], while using "OR" requires that only one of the primary tests be met to use the primary parameters [pa - ph]. If there are no primary test conditions, leave the primary parameters [pa -ph] blank; the secondary parameters [sa - sh] will be used since the conditions are blank.

To use a combination of AND/OR conditions, Primary Test 1 and Primary Test 2 are considered to be parenthetically linked. Primary Test 3 and Primary Test 4 are similarly linked. The AND/OR parameter entered for [q] is considered the logical link between the two parenthetical tests. Thus a combination of AND/OR tests as follows would be interpreted properly:

Case 1: (Test 1 OR Test 2) AND (Test 3 OR Test 4)

Case 2: (Test 1 AND Test 2) OR (Test 3 AND Test 4)

Case 3: (Test 1 OR Test 2) AND (Test 3 AND Test 4)

The **[l], [q]** or **[v]** fields can contain either an "AND" or "OR" value. This value is used in relating the Primary Test tests. If Primary Test 2, 3 or 4, is not needed, leave this field blank. The **[h], [m], [r]** or **[w]** fields identify a relational operator that determines how the test's constant value and a value found for a student are to be compared. Valid relational operators are:

```
        "="     Equal To                "!"     Not Equal To
        "<"     Less Than               ">"     Greater Than
```

[i], [n], [s] and [x] fields identify a fixed or constant value used for comparison against student information. This value cannot exceed 4 characters, but can be either a numeric or 1-4 alphabetic code.

[j] and [k], [o] and [p], [t] and [u], [y] and [z] are fields used to enter special keywords that are read by BILLING and translated to actual values found in a student's database records. If the 4 character keyword or "file" field ([j], [o], [t], [y]) is filled in, there must be a corresponding value in the longer field ([k], [p], [u], [z]) used to further identify the value being looked up.

See the section on "File/Field Keywords" to reference valid codes that can be used in the Primary Test fields. It should be noted that the keywords "CODE, SAME, SUBS, or UNIT" cannot be used in the Primary Test fields. Also, if a database record keyword is used, e.g., TFAC, SERV, etc. and the referenced database record cannot be found for a student, the primary test condition will fail.

Primary Section Fields
The primary section fields are used to determine information used to charge students when the Primary Test conditions are met. If the primary section or side does not apply for any charge, leave all the primary fields completely blank.

Primary Charge Code [pa], [pb] - (assess_pcc_file, assess_pcc_field) are fields used to determine the actual charge code used post a student billing charge to a student's account. This value becomes the link from the Assessment table to the Charge table. Any valid file keyword can be used in this field except for "SAME".

Primary Charge Units [pc], [pd] - (assess_pcu_file, assess_pcu_field) are fields that determine the charge units used in calculating a fee. When the Charge table for the assessment is found, the charge units are multiplied by the rate specified in the Charge table. In the following equation used to calculate the charge fee, the charge units are substituted for the "UNITS" variable in the equation. Any valid keyword can be used in this field except for "SUBS" or "SAME". The Charge Units field must contain a valid keyword even though the charge rate may be 0.

```
          Fees = Base + Rate * ((UNITS - Offset) / Step)
```

Primary Break Code [pe], [pf], Primary Break Units [pg], [ph] - (assess_pbc_file, assess_pbc_field, assess_pbu_file, assess_pbu_field) are used to determine which charge break level is to be used to calculate the fee. The Charge table (see below) allows up to three separate break levels that can be used to charge up to three different fee amounts within the same charge record. A break level can be selected either by a 1 to 4 character code value or by a numeric value (using Break Units), but not both. One of the Break Code or Break Units fields must be used; if there is only one break level being used in the Charge table, the keyword "SAME" can be used as shorthand to say that the Break Code or Break Units are the same as the Charge Units. Any valid keyword can be used in the Break Code or Break units except for "SUBS".

Secondary Section Fields
The secondary section fields for **Secondary Charge Code [sa], [sb], Secondary Charge Units [sc], [sd], Secondary Break Code [se], [sf],** and **Secondary Break Units [sg], [sh]** function in the same manner as the corresponding primary fields. The main difference is that the secondary side is only used by BILLING when the Primary Test fails.

Refund Section Fields
The Refund Section is used to reference a refund schedule from the Refund table (rfnd_table). This allows percentage refunds to be given back to students based on date values. For percentage refunds to work, there must be a date value that can be used to find the correct refund schedule and percentage that applies on that date. If there is not any date that can be referenced, a refund schedule cannot be used, and BILLING will refund 100% of the fee.

Refund dates can be specified in two ways. Since the registration program (REGENT) automatically adds a Registration record (reg_rec) for each course add or drop with the date of the add or drop, the BILLING program can access these dates and refund students appropriately based on the actual drop dates. To use this method, the Primary Charge Units or Secondary Charge Units must use one of the accumulator keywords, e.g., THRS, RHRS, etc. In addition, the **Refund Conditions [ra]** field must contain the keyword "TUIT". If the Assessment Charge Units fields do not contain one of the accumulator keywords, the Refund Conditions field must be blank. There are only two valid values for this field, "TUIT" or blank.

The second method involves charges not based on course adds or drops, but still depends on a date value. The **Refund Date [rb], [rc]** fields allow a date from a database record to be specified. Any 4 character keyword that references a database record can be used, e.g., ACAD, SERV, PENR, etc. The field **[rc]** must be a database field in the given record that is a date type field.

The **Refund Schedule [rd]** field is where the refund schedule code is entered. This should be a valid code from the Refund table that matches the session and academic program specified in the Assessment table.

File/Field Keywords
The Assessment table field described above allow the use of certain keywords that are translated by BILLING when a student is selected. The keywords are translated to a value usually corresponding to a value from the student's database records. For example, if registered hours are specified, the registered hours for the selected student are calculated. Similarly, if the database record containing the dorm field is given, the actual code for the dorm the student is signed up for is found and used by BILLING.

Each of the primary or secondary fields listed above, contain two fields, a four character code field, and a 24 character field that is used to further describe the code. The 4 character field is referred to as the "file" or "keyword" field, and the longer field is referred to as the "field". There must be an entry in both the "file" and "field" for every type of keyword, except for the keyword "SAME".

A description of the three types of valid keywords and how they can be used follows:

```
Database File Keywords:
    These keywords reference a list of database records that the BILLING
    program can look up for a student.  For each of the following
    records, the "field" value must be non-blank and must be a valid
    database field name from the database record specified.
    "ACAD"              Student Academic Record (stu_acad_rec)
    "CUST"              Student Billing Customization Record (sbcust_rec)
    "ID"                ID Record (id_rec)
    "PENR"              Program Enrollment Record (prog_enr_rec)
    "PROF"              Profile Record (profile_rec)
    "SERV"              Student Services Record (stu_serv_rec)
    "SUBA"              Subsidiary Account Record (suba_rec).
    "TFAC"              Facility Table (facil_table) - based on values
                        for campus, dorm, and room in the stu_serv_rec.
Keywords Based on Accumulated Hours or Courses:
    When using the accumulator values, the "field" must contain either
    the code "ALL", which causes BILLING to total all of the hours or
    courses for the keyword type, or a code that is a valid tuition,
    fee, or bill code.  When the latter is used, BILLING only totals
    the number of hours or courses a student is registered for that contain
    the tuition, fee, or bill code specified.
    "THRS"              Total number of registered and audit hours
    "TCRS"              Total number of registered and audit courses
    "RHRS"              Total number of registered hours
    "RCRS"              Total number of registered courses
    "AHRS"              Total number of audit hours
    "ACRS"              Total number of audit courses
    "FHRS"              Total number of hours for courses with a fee code
    "FCRS"              Total number of courses for a fee code
    "BHRS"              Total number of hours for courses with a bill fee
                        code
    "BCRS"              Total number of courses with a bill fee code
Other Keywords:
    "CODE"              BILLING will reference the "field" value entered
                        as literal text.
    "SAME"              Used when the break code or break units are the
                        same as charge code or charge units.  This code
                        requires a blank "field" value.
    "SUBS"              Used when the "field" contains a subsidiary
                        code other than the main student receivables
                        account code (usually "S/A").  This must be
                        a subsidiary without balance and total records.
    "UNIT"              BILLING will use the value in the "field" as
                        a numeric type.
```

Charge Table (chg_table)
The Charge table is used to calculate the actual dollar amount of a fee based on information determined from the Assessment table. For every Assessment table that references a Charge Code, there must be a Charge table.

The following shows the layout of the PERFORM screen used to enter information into the Charge table with an explanation of the fields following:

```
                                 CHARGE TABLE
Charge Code.[a   ][b                       ]   Session.....[d   ]
Program.....[c   ]                             Year........[e   ]
                                               Subsession..[f ]
------------------------------ BREAK DEFINITIONS ------------------------------
 Level    Break Unit   OR   Break Code               Description
 -----    ----------        ----------     -----------------------------------
  #1      < [g      ]         [i   ]      [l                                   ]
  #2   between #1 and #3      [j   ]      [m                                   ]
  #3      > [h      ]         [k   ]      [n                                   ]
------------------------------- FEE CALCULATIONS ------------------------------
 Level   Base Amt      Rate      Offset      Step       Minimum     Maximum
 -----   --------    --------    -------    -------     --------    --------
  #1    [o       ]  [p       ]  [q      ]  [r      ]  |  [s      ]  [t       ]
  #2    [        ]  [        ]  [       ]  [       ]  |  [       ]  [        ]
  #3    [        ]  [        ]  [       ]  [       ]  |  [       ]  [        ]
  Fees = Base + Rate * ((UNITS - Offset) / Step)  rounded to nearest [u       ]
```

Charge Table Identification Fields

Charge Code [a], [b] identifies the charge code used for posting the charge. This code must have a corresponding entry in the Subsidiary Total table (subt_table). A description for reference purposes can be added to **[b]** to describe the charge code. This description is not the charge description that is printed on the SDS Bill; it is only used for reference purposes in the Charge table. The descriptions printed on the statements and SDS Bill come from the Subsidiary Total table (subt_table).

Program, Session, Year, Subsession [c-f] fields are used to identify which academic program the charge applies to and the session for which the charge is to be calculated. These fields are used by BILLING to join to the Assessment table.

Charge Table Break Definition Fields
Each entry in the Charge table can have a maximum of three different methods for computing a charge based on the break codes or break units. Note that break codes and break units cannot be used at the same time within the same Charge table entry. Values in the Assessment table determine whether break units or break codes are used.

Break Units [g], [h] are used if the Assessment table is using break units instead of break codes to determine the charge level. In this instance, the first charge level is applied if the break units established by the Assessment table are less than the first break unit value in the charge record. The second level is applied if the assessment break units fall between the first and second charge break unit values. The last level is applied when the assessment break units are greater than the second charge break units value.

Break Codes [i-k] are used to determine which charge level is to be used in calculating the fee. If no match exists, the last charge level will be used. Matches are done character by character except in the case of a wild card position, which implies that any character may be substituted for the wild card character.

A wild card comparison can be defined in any of the Charge table break code fields. A wild card can be set up by substituting an "*" for any character in the break code. For example, to charge all break codes beginning with a "M" using the same equation, the Charge table's break code can be defined as "M***".

The **Description [l-n]** fields can be used for descriptive comments explaining to the user how the charge is calculated.

Charge Table Fee Calculation Fields
Charges or fees are calculated by the following formula which allows either a fixed amount calculation, or a variable rate calculation:

Fees = Base + Rate * ((UNITS - Offset) / Step)

Each of the variables in the equation except for "UNITS" are defined within the Charge table. The "UNITS" variable comes from the value determined by the Primary Charge Units or the Secondary Charge Units in the Assessment table. The **Base [o]** amount is a fixed amount that is added to an optionally calculated rate. The **Rate [p]** is a variable amount based on the "UNITS" less any **Offset [q]** amount divided by a **Step [r]** amount. If the offset value is a zero value, BILLING will default a value of 1 to avoid division by zero errors.

The following are a few examples showing how this equation can be used to calculate a billing fee:

```
            For a basic fee of $50.00 per student:
                Fee = 50.00 + 0 * (UNITS - 1) / 1
            For a tuition charge of $100.00 per credit (UNITS):
                Fee = 0 + 100.00 * (UNITS - 0) / 1
            For a tuition charge of $1000.00 plus $100.00 for every
            hour over 18 credits (UNITS):
                Fee = 1000.00 + 100.00 * (UNITS - 18) / 1
            For a fee of $10.00 for every three credit hours:
                Fee = 0 + 10.00 * (UNITS - 0) / 3
            For a tuition charge of $1000 plus $300 for every 3
            credits over 18:
                Fee = 1000 + 300 * (UNITS - 18) / 3
        (Note:  The above calculation is not for every fraction of 3
                credits, since BILLING only uses whole number or integer
                values.)
```

Charges can be rounded to the nearest cent or dollar by placing a value in the **increment [u]** ("rounded to nearest") field in the charge table. Increments within the charge table are type money with a default of $0.01 causing the charge to be rounded to the nearest cent. Any amount can be used for rounding. Note that refunds are always rounded to the nearest penny regardless of the increment field value.

The **Minimum [s]** field allows a minimum charge amount to be specified. No matter what fee amount is calculated by BILLING, if the charge minimum is non-zero, the fee will not be less than the minimum value.

The **Maximum [t]** field allows a maximum fee amount to be specified. If the maximum field is non-zero, the calculated fee will not exceed the maximum amount.

Refund Table (rfnd_table)
The Refund table establishes a refund schedule for any charge to be refunded back to students, and must be created for each session, subsession, year, and program along with the Assessment and Charge tables. The Refund table allows a refund schedule to be entered that lists refundable percentages valid for date ranges.

All assessment codes should have a refund schedule specified even if there is not to be any refund granted. If there is not to be any refund, a refund schedule can be added with a refund percentage of "0" and a refund date one less than the beginning date of the billing session (e.g., session begins on 08/15/85, the refund date equals 08/14/85). Note that charges will be refunded at 100% regardless of the refund schedule used if the assessment Charge Units are not based on one of the accumulator keywords (RHRS, TCRS, FHRS, etc.).

Students marked as "no-shows" (stuac_reg_stat = "N") will receive a 100% refund of all charges regardless of what is in the Assessment Table for the refund schedule.

Classification Table (cl_table)
The Classification table along with defining the various student classifications, determines which classifications are billed through BILLING. Any classification to be billed must contain a "Y" value in the

billing field (cl_bill). If any classification is not to billed by BILLING, this field should be set to "N". The latter case should be used if, for various reasons, it is easier or more efficient to manually charge students under any classification.

Subsidiary Total Table (subt_table)
The Subsidiary Total table defines the General Ledger expense account for every charge. Every code in the Charge table must have an equivalent Subsidiary Total table code.

A **Lost Tot** refers to any charge that was posted at one time by BILLING, but in a subsequent run, the BILLING program cannot justify why the charge was posted to a student's account. It may be is easier to think of a "lost tot" in terms of an "unlinked tot". In order for BILLING to calculate a charge, there must be an Assessment and a Charge table for the charge. In addition, the Assessment table must allow the student to be charged a given fee. If at some time, an Assessment table is changed or the student no longer satisfies conditions within an Assessment table, the BILLING program cannot link from the Assessment to the Charge table to calculate a fee. In this case, the once calculated charge becomes a "lost" or "unlinked" tot. In other words, BILLING cannot determine based on current assessment information why the charge was ever posted. BILLING will then refund the charge at 100% if the tsubt_lost flag is set to "Y".

Example Assessment and Charge Tables
This section gives a few examples showing how the Assessment and Charge tables can be set for certain types of fees.

Lab Fees
Charge a computer lab fee of $30.00 assessed per course registered with a fee code of "COMP". This fee is to be charged any student who signed up for a course with the fee code "COMP". In addition, the refund schedule "RFND" is to be used to calculate non-refundable amounts when students drop courses. The fee code must be placed in the Section record's (sec_rec) fee code (sec_fee_code) field before registration.

The assessment will be translated as follows by BILLING, The Charge Code used will be "COMP", the Charge Units will be the actual number of courses a student is registered for with the fee code of "COMP". The Break Units will be the same value as the Charge Units.

The Charge Level selected will be the 3rd level, since there will be no match for levels 1 and 2. The fee is a rate of $30.00 multiplied by the Charge Units value.

```
                                ASSESSMENT TABLE
Code.............[COMP][Computer Fee            ]            Program......[UNDG]
Billing Code.....[SB  ] Session...[FA  ]  Year........[19XX] Subsession...[  ]
Primary Test 1.......[ ].[    ].[    ]    Primary Charge Code.............[    ]
[                                    ]    [                                    ]
Primary Test 2.[   ].[ ].[    ].[    ]    Primary Charge Units............[    ]
[                                    ]    [                                    ]
Primary Test 3.[   ].[ ].[    ].[    ]    Primary Break Code..............[    ]
[                                    ]    [                                    ]
Primary Test 4.[   ].[ ].[    ].[    ]    Primary Break Units.............[    ]
[                                    ]    [                                    ]
                                          Secondary Charge Code...........[CODE]
----------REFUND INFORMATION----------    [COMP                                ]
                                          Secondary Charge Units..........[FCRS]
Refund Conditions...............[TUIT]    [COMP                                ]
Refund Date.....................[    ]    Secondary Break Code............[    ]
[                                    ]    [                                    ]
Refund Schedule.................[RFND]    Secondary Break Units...........[SAME]
as of Date.. 00/00/00    Pct..      0     [                                    ]
```

```
                                 CHARGE TABLE
Charge Code.[COMP][Computer Lab Fee        ]   Session.....[FA  ]
Program.....[UNDG]                             Year........[19XX]
                                               Subsession..[  ]
------------------------------ BREAK DEFINITIONS ------------------------------
 Level    Break Unit   OR   Break Code                Description
 -----    ----------        ----------     -----------------------------------
  #1      < [0      ]         [    ]      [                                   ]
  #2   between #1 and #3      [    ]      [                                   ]
  #3      > [0      ]         [    ]      [$30.00 per course                  ]
------------------------------- FEE CALCULATIONS ------------------------------
 Level   Base Amt      Rate       Offset      Step        Minimum     Maximum
 -----   --------    --------    -------    -------       --------    --------
  #1    [        ]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  #2    [        ]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  #3    [        ]  [   30.00]  [       ]  [       ]  |  [        ]  [        ]
  Fees = Base + Rate * ((UNITS - Offset) / Step)  rounded to nearest [    0.01]
```

Drop/Add Fees
Charge students $10.00 per course added or dropped. The REGENT program maintains the database field used in the assessment containing a count of each course added or dropped. A refund schedule will not be used for this assessment, since the Charge Units are not one of the accumulator keywords. Any refund schedule specified will be ignored and the student refunded at 100%.

The Charge Code used to calculate and post the fee is "ADRP", the rate is calculated by the value in the stu_acad_rec field "stuac_no_adrps". If this field is 0, there is no charge (10.00 * rate of 0 = 0).

```
                                ASSESSMENT TABLE
Code.............[ADRP][Add/Drop Fee            ]            Program......[UNDG]
Billing Code.....[SB  ] Session...[FA  ]  Year........[19XX] Subsession...[  ]
Primary Test 1.......[ ].[    ].[    ]    Primary Charge Code............[    ]
[                                    ]    [                                   ]
Primary Test 2.[   ].[ ].[    ].[    ]    Primary Charge Units...........[    ]
[                                    ]    [                                   ]
Primary Test 3.[   ].[ ].[    ].[    ]    Primary Break Code.............[    ]
[                                    ]    [                                   ]
Primary Test 4.[   ].[ ].[    ].[    ]    Primary Break Units............[    ]
[                                    ]    [                                   ]
                                          Secondary Charge Code..........[CODE]
----------REFUND INFORMATION----------    [ADRP                               ]
                                          Secondary Charge Units.........[ACAD]
Refund Conditions..............[    ]    [tuac_no_adrps                      ]
Refund Date....................[    ]    Secondary Break Code...........[    ]
[                                    ]    [                                   ]
Refund Schedule................[    ]    Secondary Break Units..........[SAME]
as of Date.. 00/00/00    Pct..      0     [                                   ]
```

```
                                 CHARGE TABLE
Charge Code.[ADRP][Add/Drop Charge         ]   Session.....[FA  ]
Program.....[UNDG]                             Year........[19XX]
                                               Subsession..[  ]
------------------------------ BREAK DEFINITIONS ------------------------------
 Level    Break Unit   OR   Break Code                Description
 -----    ----------        ----------     -----------------------------------
  #1      < [      0]         [    ]      [                                   ]
  #2   between #1 and #3      [    ]      [                                   ]
  #3      > [      0]         [    ]      [$10.00 PER ADD/DROP                ]
------------------------------- FEE CALCULATIONS ------------------------------
 Level   Base Amt      Rate       Offset      Step        Minimum     Maximum
 -----   --------    --------    -------    -------       --------    --------
  #1    [        ]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  #2    [        ]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  #3    [        ]  [   10.00]  [       ]  [       ]  |  [        ]  [        ]
  Fees = Base + Rate * ((UNITS - Offset) / Step)  rounded to nearest [    0.01]
```

Dorm Charge
The dorm charge is set up to use a Charge table with a Charge Code equal to the dorm code. This method allows a separate price to be attached to each dorm and each dorm's fees will be in a unique

total code for reporting purposes. In addition, the actual fee calculated depends on number of students that can occupy the room. The example shows how one Assessment can reference multiple Charge tables depending on a value found in a database field. Note that the assessment code is not the same as the charge code; the link from the assessment to the charge is by the code determined by the Primary or Secondary Charge table fields. The Charge table shown will be for the "HOPE" dorm which has the fee of $500 for a double, $750 for a single room, and $800 for rooms with 3 or more students. The occupancy is defined in the Facility Table (facil_table).

When the assessment is translated, BILLING will attempt to find a stu_serv_rec for the "FA" "19XX" year. If the record cannot be found, or the stusv_dorm field is blank, the student will not be charged. The Charge Units and Break Units are based on the maximum occupancy field in the facility table corresponding to the dorm code, campus code, and building code in the stu_serv_rec.

```
                                ASSESSMENT TABLE
Code.............[DORM][Resident Hall Charges   ]            Program......[UNDG]
Billing Code.....[SB  ] Session...[FA  ]  Year........[19XX] Subsession...[  ]
Primary Test 1.......[ ].[    ].[    ]    Primary Charge Code.............[    ]
[                                    ]    [                                    ]
Primary Test 2.[   ].[ ].[    ].[    ]    Primary Charge Units............[    ]
[                                    ]    [                                    ]
Primary Test 3.[   ].[ ].[    ].[    ]    Primary Break Code..............[    ]
[                                    ]    [                                    ]
Primary Test 4.[   ].[ ].[    ].[    ]    Primary Break Units.............[    ]
[                                    ]    [                                    ]
                                          Secondary Charge Code...........[SERV]
----------REFUND INFORMATION----------    [stusv_dorm                          ]
                                          Secondary Charge Units..........[TFAC]
Refund Conditions...............[    ]    [facil_max_occ                       ]
Refund Date.....................[    ]    Secondary Break Code............[    ]
[                                    ]    [                                    ]
Refund Schedule.................[    ]    Secondary Break Units...........[SAME]
as of Date.. 00/00/00    Pct..      0     [                                    ]
```

```
                                 CHARGE TABLE
Charge Code.[HOPE][HOPE Dorm              ]   Session.....[FA  ]
Program.....[UNDG]                            Year........[19XX]
                                              Subsession..[  ]
------------------------------ BREAK DEFINITIONS ------------------------------
 Level    Break Unit   OR   Break Code                Description
 -----    ----------        ----------     ------------------------------------
  #1      < [      2]         [    ]      [$500 PER DOUBLE ROOM                 ]
  #2   between #1 and #3      [    ]      [$750 PER DOUBLE ROOM                 ]
  #3      > [      2]         [    ]      [$800 PER MULTIPLE OCCUPANCY          ]
------------------------------- FEE CALCULATIONS ------------------------------
 Level   Base Amt      Rate       Offset      Step        Minimum     Maximum
 -----   --------    --------    -------    -------      --------    --------
  #1    [  500.00]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  #2    [  750.00]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  #3    [  800.00]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  Fees = Base + Rate * ((UNITS - Offset) / Step)  rounded to nearest [    0.01]
```

Tuition Charges
The example shows a tuition assessment that distinguishes between in-state and out-of-state fees. The example also shows how the Charge table Break Levels are used to calculate part-time and full-time fees. For in-state students, tuition is calculated to be $250 per credit for part-time students (less than 12 hours), and $3300 for full-time students (12-18 hours). Students that carry over 18 hours are charged an additional $100 per hour over 18 hours. Out-of-state students are charged $400 per credit hour full or part-time.

The Assessment below shows how the Primary Test is used. If the res_st field in the Profile record (profile_rec) contains a "OH" value, the primary side will be used to charge the student. If the Profile record cannot be found for a student, or the res_st field is not equal to "OH", the secondary side will be used to charge the student.

The primary side will use a charge code of "OTUI" and calculate the charge units to be the total number of registered hours. The secondary side will use a charge code of "NTUI" and the Charge Units will also be the total number of registered hours.

```
                                ASSESSMENT TABLE
Code.............[TUIT][Tuition Assessment      ]            Program......[UNDG]
Billing Code.....[SB  ] Session...[FA  ]  Year........[19XX] Subsession...[  ]
Primary Test 1.......[=].[OH  ].[PROF]    Primary Charge Code.............[CODE]
[res_st                              ]    [OTUI                               ]
Primary Test 2.[   ].[ ].[    ].[    ]    Primary Charge Units............[RHRS]
[                                    ]    [ALL                                ]
Primary Test 3.[   ].[ ].[    ].[    ]    Primary Break Code..............[    ]
[                                    ]    [                                   ]
Primary Test 4.[   ].[ ].[    ].[    ]    Primary Break Units.............[SAME]
[                                    ]    [                                   ]
                                          Secondary Charge Code...........[CODE]
----------REFUND INFORMATION----------    [NTUI                               ]
                                          Secondary Charge Units..........[RHRS]
Refund Conditions...............[TUIT]    [ALL                                ]
Refund Date.....................[    ]    Secondary Break Code............[    ]
[                                    ]    [                                   ]
Refund Schedule.................[RFND]    Secondary Break Units...........[SAME]
as of Date.. 00/00/00    Pct..      0     [                                   ]
```

```
                                 CHARGE TABLE
Charge Code.[OTUI][Ohio Tuition           ]   Session.....[FA  ]
Program.....[UNDG]                            Year........[19XX]
                                              Subsession..[  ]
------------------------------ BREAK DEFINITIONS ------------------------------
 Level    Break Unit   OR   Break Code              Description
 -----    ----------        ----------     -----------------------------------
  #1      < [   12.0]         [    ]      [PART TIME FEE = $250 PER HOUR       ]
  #2   between #1 and #3      [    ]      [FULL TIME FEE 12-18 HOURS = $3300   ]
  #3      > [   18.0]         [    ]      [> 18 HOURS = $3300 + ($100/HR > 18) ]
------------------------------- FEE CALCULATIONS ------------------------------
 Level   Base Amt      Rate      Offset      Step        Minimum     Maximum
 -----   --------    --------    -------    -------      --------    --------
  #1    [        ]  [  250.00]  [       ]  [       ]  |  [        ]  [        ]
  #2    [ 3300.00]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  #3    [ 3300.00]  [    0.00]  [   18.0]  [       ]  |  [        ]  [        ]
  Fees = Base + Rate * ((UNITS - Offset) / Step)  rounded to nearest [    0.01]
```

```
                                 CHARGE TABLE
Charge Code.[NTUI][Non-Ohio Tuition       ]   Session.....[FA  ]
Program.....[UNDG]                            Year........[19XX]
                                              Subsession..[  ]
------------------------------ BREAK DEFINITIONS ------------------------------
 Level    Break Unit   OR   Break Code              Description
 -----    ----------        ----------     -----------------------------------
  #1      < [       ]         [    ]      [                                    ]
  #2   between #1 and #3      [    ]      [                                    ]
  #3      > [       ]         [    ]      [$400 PER HOUR                       ]
------------------------------- FEE CALCULATIONS ------------------------------
 Level   Base Amt      Rate      Offset      Step        Minimum     Maximum
 -----   --------    --------    -------    -------      --------    --------
  #1    [        ]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  #2    [        ]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  #3    [        ]  [  400.00]  [       ]  [       ]  |  [        ]  [        ]
  Fees = Base + Rate * ((UNITS - Offset) / Step)  rounded to nearest [    0.01]
```

Meal Plan Charges
A meal plan charge that is based on a full-time or part-time rate of $700.00 or $400.00. The rate is determined by a field in the stu_serv_rec which contains the meal plan code. If the meal plan code is not "F" or "P", do not charge the student.

The decision for the meal plan type is accomplished by using the charge break levels. The secondary break code references the stu_serv_rec and causes BILLING to lookup the stu_serv_rec for the program and session and read the contents of the meal_plan_type field. If the stu_serv_rec is not found, the student will not be charged. The secondary charge units, even though not needed to calculate a rate in the Charge table, must contain a valid value. This is done by using "UNIT" of 1, which is effectively ignored by the charge calculation since the rate is zero.

```
                                ASSESSMENT TABLE
Code.............[MEAL][Meal Plan               ]            Program......[UNDG]
Billing Code.....[SB  ] Session...[FA  ]  Year........[19XX] Subsession...[  ]
Primary Test 1.......[ ].[    ].[    ]    Primary Charge Code............[    ]
[                                    ]    [                                   ]
Primary Test 2.[   ].[ ].[    ].[    ]    Primary Charge Units...........[    ]
[                                    ]    [                                   ]
Primary Test 3.[   ].[ ].[    ].[    ]    Primary Break Code.............[    ]
[                                    ]    [                                   ]
Primary Test 4.[   ].[ ].[    ].[    ]    Primary Break Units............[    ]
[                                    ]    [                                   ]
                                          Secondary Charge Code..........[CODE]
----------REFUND INFORMATION----------    [MEAL                               ]
                                          Secondary Charge Units.........[UNIT]
Refund Conditions..............[    ]    [1                                  ]
Refund Date....................[    ]    Secondary Break Code...........[SERV]
[                                    ]    [meal_plan_type                     ]
Refund Schedule................[    ]    Secondary Break Units..........[    ]
as of Date.. 00/00/00    Pct..      0     [                                   ]
```

```
                                 CHARGE TABLE
Charge Code.[MEAL][Meal Plan              ]   Session.....[FA  ]
Program.....[UNDG]                            Year........[19XX]
                                              Subsession..[  ]
------------------------------ BREAK DEFINITIONS ------------------------------
 Level    Break Unit   OR   Break Code              Description
 -----    ----------        ----------     -----------------------------------
  #1      < [       ]         [F   ]      [FULL TIME PLAN = $700               ]
  #2   between #1 and #3      [P   ]      [PART TIME PLAN = $400               ]
  #3      > [       ]         [    ]      [NO CHARGE                           ]
------------------------------- FEE CALCULATIONS ------------------------------
 Level   Base Amt      Rate       Offset      Step        Minimum     Maximum
 -----   --------    --------    -------    -------      --------    --------
  #1    [  700.00]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  #2    [  400.00]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  #3    [    0.00]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  Fees = Base + Rate * ((UNITS - Offset) / Step)  rounded to nearest [    0.01]
```

Course Audit Fees

Audit fees are charged only for part-time students or for any non-resident student. The fee is $65.00 per course audited for resident part-time students, and $85.00 per course for any non-resident student.

In the assessment, the primary test is used to test for all students registered for less than 12.0 hours *or* students who are non-Ohio residents. If neither of these conditions are met, the secondary side is used by BILLING to charge the student. Since the secondary side does not have any information, no charge will be calculated if the primary test conditions are not met.

The charge units are the total number of audited courses, and the break code will be the value in the profile_rec for resident state. If there is no Profile record, the student will not be charged any audit fee.

The Charge table uses the break code levels to determine the fee amount. If the res_st field is blank, there is no charge, if the field contains an "OH", there is a $65.00 charge, and if the field is anything other than blank or "OH" the $85.00 fee is charged. In this case, it does not matter what value or code is specified in the 3rd break level since it is always the default level when there is no match in the first or second level break codes.

```
                                ASSESSMENT TABLE
Code.............[AUDT][Audit Charge           ]           Program......[UNDG]
Billing Code.....[SB  ] Session...[FA  ]  Year........[19XX] Subsession...[  ]
Primary Test 1.......[<].[12.0].[RHRS]    Primary Charge Code.............[CODE]
[ALL                                 ]    [AUDT                              ]
Primary Test 2.[OR ].[!].[OH  ].[PROF]    Primary Charge Units............[ACRS]
[res_st                              ]    [ALL                               ]
Primary Test 3.[   ].[ ].[    ].[    ]    Primary Break Code..............[PROF]
[                                    ]    [res_st                            ]
Primary Test 4.[   ].[ ].[    ].[    ]    Primary Break Units.............[    ]
[                                    ]    [                                  ]
                                          Secondary Charge Code...........[    ]
----------REFUND INFORMATION----------    [                                  ]
                                          Secondary Charge Units..........[    ]
Refund Conditions...............[TUIT]    [                                  ]
Refund Date.....................[    ]    Secondary Break Code............[    ]
[                                    ]    [                                  ]
Refund Schedule.................[RFND]    Secondary Break Units...........[    ]
as of Date.. 00/00/00    Pct..      0     [                                  ]
```

```
                                 CHARGE TABLE
Charge Code.[AUDT][Audit Charge            ]   Session.....[FA  ]
Program.....[UNDG]                             Year........[19XX]
                                               Subsession..[  ]
------------------------------ BREAK DEFINITIONS -----------------------------
 Level    Break Unit   OR   Break Code               Description
 -----    ----------        ----------     -----------------------------------
  #1      < [       ]         [    ]      [NO CHARGE IF BLANK                  ]
  #2   between #1 and #3      [OH  ]      [$65.00 FOR OHIO RESIDENTS           ]
  #3      > [       ]         [    ]      [$85.00 FOR NON-OHIO RESIDENTS       ]
------------------------------- FEE CALCULATIONS -----------------------------
 Level   Base Amt      Rate       Offset      Step        Minimum     Maximum
 -----   --------    --------    -------    -------       --------    --------
  #1    [        ]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  #2    [        ]  [   65.00]  [       ]  [       ]  |  [        ]  [        ]
  #3    [        ]  [   85.00]  [       ]  [       ]  |  [        ]  [        ]
  Fees = Base + Rate * ((UNITS - Offset) / Step)  rounded to nearest [    0.01]
```

Pre-Payment Credits
BILLING can automatically apply the balance of a subsidiary pre-payment type account to the student's account. An example of this, is a student who prepays a $150.00 payment which is then to be automatically credited to his bill when charges are posted through the BILLING program. The following is an example showing how this is set up in the Charge and Assessment tables using a subsidiary code of "PRE". Note that the subsidiary code and the charge code must be the same value. Using this method, BILLING will credit the entire amount in the pre-payment subsidiary to the student accounts subsidiary and debit the pre-payment subsidiary to wipe out the balance for a student. If the "PRE" subsidiary balance is zero for any student, no credit can be applied to the student's account.

In the assessment, the secondary charge code "file" value is "SUBS". This has special meaning to BILLING, in that the charge code will be the subsidiary code specified in the "field", i.e., "PRE". The Charge Units are set to a zero amount since the charge is actually not going to have any rate or amounts specified. BILLING actually "refunds" the "PRE" subsidiary balance since the charge fee calculation results in a zero fee.

```
                                ASSESSMENT TABLE
Code.............[PRE ][Pre-payment credit      ]            Program......[UNDG]
Billing Code.....[SB  ] Session...[FA  ]  Year........[19XX] Subsession...[  ]
Primary Test 1.......[ ].[    ].[    ]    Primary Charge Code.............[    ]
[                                    ]    [                                    ]
Primary Test 2.[   ].[ ].[    ].[    ]    Primary Charge Units............[    ]
[                                    ]    [                                    ]
Primary Test 3.[   ].[ ].[    ].[    ]    Primary Break Code..............[    ]
[                                    ]    [                                    ]
Primary Test 4.[   ].[ ].[    ].[    ]    Primary Break Units.............[    ]
[                                    ]    [                                    ]
                                          Secondary Charge Code...........[SUBS]
----------REFUND INFORMATION----------    [PRE                                 ]
                                          Secondary Charge Units..........[UNIT]
Refund Conditions...............[    ]    [0                                   ]
Refund Date.....................[    ]    Secondary Break Code............[    ]
[                                    ]    [                                    ]
Refund Schedule.................[    ]    Secondary Break Units...........[SAME]
as of Date.. 00/00/00    Pct..      0     [                                    ]
```

```
                                 CHARGE TABLE
Charge Code.[PRE ][Pre-payment credit      ]   Session.....[FA  ]
Program.....[UNDG]                             Year........[19XX]
                                               Subsession..[  ]
------------------------------ BREAK DEFINITIONS ------------------------------
 Level    Break Unit   OR   Break Code               Description
 -----    ----------        ----------     ------------------------------------
  #1      < [       ]         [    ]      [                                    ]
  #2   between #1 and #3      [    ]      [                                    ]
  #3      > [       ]         [    ]      [CREDIT ENTIRE BALANCE               ]
------------------------------- FEE CALCULATIONS ------------------------------
 Level   Base Amt      Rate       Offset      Step        Minimum     Maximum
 -----   --------    --------    -------    -------      --------    --------
  #1    [        ]  [        ]  [       ]  [       ]  |  [       ]  [        ]
  #2    [        ]  [        ]  [       ]  [       ]  |  [       ]  [        ]
  #3    [        ]  [       0]  [       ]  [       ]  |  [       ]  [        ]
  Fees = Base + Rate * ((UNITS - Offset) / Step)  rounded to nearest [    0.01]
```

Holding Accounts
Non-student account subsidiaries can also be used as holding accounts. For example, a graduation fee of $50.00 is required to be paid by all seniors which is held until graduation. If students do not have $50.00 in the account, a charge is generated to bring the graduation subsidiary balance up to $50.00. If the graduation subsidiary balance is over $50.00, any amount over $50.00 is refunded back to the student. The example uses a subsidiary code of "GRAD".

The assessment tests for students with a classification code of "SR" in the Program Enrollment record. If this record does not exist or the classification code is not "SR", the student will be refunded any balance in the "GRAD" subsidiary.

```
                                ASSESSMENT TABLE
Code.............[GRAD][Graduation Fee          ]            Program......[UNDG]
Billing Code.....[SB  ] Session...[FA  ]  Year........[19XX] Subsession...[  ]
Primary Test 1.......[=].[SR  ].[PENR]    Primary Charge Code.............[SUBS]
[prog_cl                             ]    [GRADS                               ]
Primary Test 2.[   ].[ ].[    ].[    ]    Primary Charge Units............[UNIT]
[                                    ]    [1                                   ]
Primary Test 3.[   ].[ ].[    ].[    ]    Primary Break Code..............[    ]
[                                    ]    [                                    ]
Primary Test 4.[   ].[ ].[    ].[    ]    Primary Break Units.............[SAME]
[                                    ]    [                                    ]
                                          Secondary Charge Code...........[SUBS]
----------REFUND INFORMATION----------    [GRAD                                ]
                                          Secondary Charge Units..........[UNIT]
Refund Conditions...............[    ]    [0                                   ]
Refund Date.....................[    ]    Secondary Break Code............[    ]
[                                    ]    [                                    ]
Refund Schedule.................[    ]    Secondary Break Units...........[SAME]
as of Date.. 00/00/00    Pct..      0     [                                    ]
```

```
                                 CHARGE TABLE
Charge Code.[GRAD][Graduation Fee          ]   Session.....[FA  ]
Program.....[UNDG]                             Year........[19XX]
                                               Subsession..[  ]
------------------------------ BREAK DEFINITIONS ------------------------------
 Level    Break Unit   OR   Break Code                Description
 -----    ----------        ----------     ------------------------------------
  #1      < [       ]         [    ]      [                                    ]
  #2   between #1 and #3      [    ]      [                                    ]
  #3      > [       ]         [    ]      [$50.00 FEE FOR SENIORS              ]
------------------------------- FEE CALCULATIONS ------------------------------
 Level   Base Amt      Rate       Offset      Step        Minimum     Maximum
 -----   --------    --------    -------    -------      --------    --------
  #1    [        ]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  #2    [        ]  [        ]  [       ]  [       ]  |  [        ]  [        ]
  #3    [        ]  [   50.00]  [       ]  [       ]  |  [        ]  [        ]
  Fees = Base + Rate * ((UNITS - Offset) / Step)  rounded to nearest [    0.01]
```

Process Flow of Student Accounting

Overview
This diagram illustrates the integration of activities between registration, student services and financial aid and their impact upon the student billing process.

Introduction
The following diagram outlines all the functions involved in maintaining student accounts. Processes such as REGENT, labeled in all caps, are application programs that cause the next step to occur. Each process is underlined. Subprocesses are listed under a process. For example, "Assessments" is a process. "Tuition" is a subprocess of "Assessments". "BILLING" is the application program which assesses charges against a student

```
                         Assessments
                         -----------
     Tuition     Course Fees          Room              Student Activity Fee
                                      Board             Health Insurance
       |             |                  |                         |
       |             |                  |                         |
       v             v                  v                         v
       ------------------------------------------------------------
                 |
                 |
                 v
              Charges           Financial Aid Awarded
              -------           ---------------------
                 |                        |
              BILLING                 FAENTRY -------------|
                 |                        |                |
                 v                        v                |
                 ---------------------------               |
                                        |                  |
                                        |                  |
                                        v                  |
 SDS <---------- REGENT/BILLING <----Student Billed        |
                        |            --------------        |
                        |                                  |
                        |                                  |
                        v                                  |
                        -------------------|               |
                                           v               v
Add Additional Charges              Post SB Voucher   Post FA Voucher
----------------------              ---------------   ---------------
     |                                     |                 |
     |                                     |                 |
    SA                                    SB                FA
  VOUCHER                               VOUCHER           VOUCHER
     |                                     |                 |
     |                                     |                 |
     v                                     v                 v
     ---------------------------------------------------------
                        |
                        | To Subsidiary Total and Balance Records
                        v
           Subsidiary Total and Balance Records
           ------------------------------------
                        |
                        |
            CH        |
          VOUCHER     |
  Payments ---------->  |
  Posted              |
  ------              |
                        v
              Update Balance Records
              ----------------------
                        |
                        |
                        |
                        v
                Monthly Statements ------STMT----------> Subsidiary Account
                ------------------                           Statement
                                                         ------------------
```

Student Billing - Using Reports and Updates to Close a Billing Period

Overview

Closing a billing period includes reconciling student data with session billing data.

Introduction

A series of ACE reports and INFORMER updates have been provided allowing the Bursar or Controller to reconcile student data with amounts charged and paid for the session. Reports providing registration data are located in "$CARSPATH/regist/reports". Housing and other student services data reports are located in "$CARSPATH/stuserv/reports". Several accounting reports are also available to provide summary data per subsidiary tot code, specifically those used for manual or automatic charges. An INFORMER Update updates subsidiary balance statuses.

Registration Data

Counts of registered students or registered hours, separated by part-time, full-time, and overloads and by classification are provided through the reports "enrregstu" and "enrreghrs", respectively. Tuition figures can be estimated using these counts.

To estimate income from course fees, use "enrcrsfees" to provide counts per course where course fees were assessed.

Student Services Data

Student services data includes housing, meal plans, any waivers, and any additional student situations not related to registration. Three reports exist providing counts from the stu_serv_rec.

Counts of students by intended housing, classification, and part-time or full-time registration status are provided using "enrres". This report can be used for estimates of dorm charges. This same report also provides an overview of dorm usage.

Mealplan types and waivers by classification and part-time or full-time registration status are counted by "enrmeal". Estimates of board charges can be calculated from these values. The same counts can be used to justify the existence of current mealplans.

Other fields are called by "enrserv" to provide additional counts from the stu_serv_rec also by classification and part-time or full-time status. These fields include "asb_fee_wvd" (Y or N), "health_ins_wvd"(Y or N), "res_asst" (Y), and "park_prmt_no" (> " "). The report may be expanded to include any additional fields used in the Assessment table for automatic or manual billing.

Accounting Data

The Subsidiary Total records need to be reconciled with the General Ledger accounts. Run "subwogl" to verify that the account numbers in the subt_table are also in the gla_rec. This verification should be run whenever many entries are made to subt_table.

To have a listing of each subsidiary tot code where subt_post = "B" or "C" with the encumbered and actual amounts from the corresponding glamt_rec, run "totbal". The fiscal year and up to two subt_post values are the report parameters.

Closing Current Subsidiary Balance Records

The following INFORMER update will close all Subsidiary Balance records for subsidiary "S/A" and balance code "SB".

```
prompt for subbprd using "Enter Subsidiary Period (eg FA84) >> ";
update noprompt subb_rec
  subb_stat = "C"
  where subb_prd = subbprd
  and subb_subs = "S/A "
  and subb_code = "SB  "
  and subb_amt_act = 0
  ;
b
```

It should be run periodically after the last time BILLING is run for the session.

Telephone Billing - How to Set Up Tables & Records

Overview

This document describes the tables and records involved in the Phone Billing module. An explanation of each field is included.

Introduction

There are six files that belong exclusively to the Phone Billing module. The Phone Call record (phcall_rec) contains the detailed information of each phone call placed. The Responsible Person record (phresp_rec) identifies a person with a phone extension number or access code. The Phone Account table (phacct_table) defines which account number to charge for calls of a responsible person. The Area Code table (ac_table) and the Zone table (zone_table) work together to identify the rates to be charged for calls. The Phone Run record (phrun_rec) is a summary record for a group of calls.

```
m4_center(`Telephone Billing Record Flow', 80)
                       ----------------------------------
                       | Phone Call Record (phcall_rec) |
                       ----------------------------------
                       ||              ||              ||
-----------------------------------    ||   -----------------------------------
|   Area Code Table (ac_table)    |    ||   | Resp Person Record (phresp_rec) |
-----------------------------------    ||   -----------------------------------
                ||                     ||                   ||
-----------------------------------    ||   -----------------------------------
|     Zone Table (zone_table)     |    ||   | Phone Acct Table (phacct_table) |
-----------------------------------    ||   -----------------------------------
                                       ||
                       ----------------------------------
                       |  Phone Run Record (phrun_rec)  |
                       ----------------------------------
```

Phone Call Record (phcall_rec)

The Phone Call record contains the detailed information of each phone call placed. The information originally comes from a telephone switch or is downloaded from a personal computer that is attached to a phone system. The record may then be updated to reflect the cost of the telephone call by running the program "Compute Call Charges". This information then serves as a basis for billing individuals or charging campus departments.

Phone call number index (phcall_no)
The number is serially assigned to call records as they are added to the system. It is used for identification only.

Extension number (phcall_ext_no)
The extension number of the call origin.

Call date (phcall_call_date)
The date a call was placed.

Call time (phcall_call_time)
The time a call was placed, format: HHMM. Time is computed using military time, a 24 hour clock.

Call duration (phcall_call_dur)
The total time of a call. Time is recorded in seconds.

Number called (phcall_called_no)
The number of the call destination.

Call charge amount (phcall_chg_amt)

The amount of the charge for a call. The amount is computed by the program "Compute Call Charges", if necessary. Some personal computers that interface with phone systems also compute call costs. If this is the case, the program "Compute Call Costs" does not need to be run.

Call recalculation (phcall_recalc)
If the charges need to be calculated the field should be yes. If the record already contains charge amounts from the phone system then the field should be no. Yes also causes charges to be recalculated if call rates have changed since the original calculation.

Call account number (phcall_acct_no)
The account number used to place a call. This is typically an additional series of numbers that are dialed when a call is placed.

Call access number (phcall_access_ no)
The access number used to place a call, if access codes are used. This is typically an additional series of numbers that are dialed when a call is placed.

Call billing code (phcall_bill_code)
The billing code of a call. This will join a call with a person based on entries in the responsible person record.

Call key1 (phcall_key1)
A composite field that identifies a call as unique. It combines the call date, call time and number called to form a unique combination.

Phone Responsible Person Record (phresp_rec)

The Phone Responsible Person record defines a caller with a billing code. The billing code may be the callers extension number, access number or any defined code. The billing code must be in the call record to identify calls with a caller. The billing code must also be downloaded from the telephone switch or personal computer attachment.

Responsible person billing code (phresp_bill_code)
The billing code joins a person or extension to a phone call.

Responsible person access number (phresp_access_ no)
The access code used to place a phone call.

Responsible person id (phresp_id)
ID number of the responsible person.

Debit code (phresp_dr_code)
Identifies the debit side of the general ledger entry for calls placed with this code. The account number comes from the Phone Account table.

Credit code (phresp_cr_code)
Identifies the credit side of the general ledger entry for calls placed with this code. The account number comes from the Phone Account table.

Phone Account Table (phacct_table)

The Phone Account table identifies entries that are made for the general ledger. After the table is set up, the phone account code is entered in the Responsible Person record. Calls for that person are then charged based on values in this table.

Phone account code (tphacct_tbcode)

The phone account code when placed in the responsible person record identifies the general ledger entry for calls of a person. The code may be any alpha-numeric sequence. Codes are needed for both the debit and credit side of the entry.

Phone account text (tphacct_text)
The description of the phone account code.

Phone account fund (tphacct_fund)
Fund for a phone account code.

Phone account center (tphacct_cntr)
Center for a phone account code.

Phone account account (tphacct_acct)
Account for a phone account code.

Phone account project (tphacct_proj)
Project for a phone account code.

Phone account subsidiary code (tphacct_subs)
Subsidiary code for a phone account code.

Phone account subsidiary balance code (tphacct_bal_ code)
Subsidiary balance code for a phone account code.

Phone account total code (tphacct_tot_code)
Subsidiary total code for a phone account code.

Area Code Table (ac_table)

The Area Code table defines the area code and exchange numbers that will be used to assess phone charges. Each record is assigned a zone which determines the rates to be charged for calls to an area code and exchange combination.

Area code (tac_no)
The area code for this record. The same area code may be entered more than once with different exchanges.

Exchange (tac_exch)
The exchange associated with the area code. A blank exchange will cause all exchanges within the area code to be billed at the same rate.

Zone (tac_zone)
The type of zone to be used for this record. This code joins to the zone table which defines the rates for charging calls.

In state (tac_in_st)
Logical definition of in-state calls. If this field is yes, the call may be billed a premium or discount based on rates defined in the Zone table.

Area code prim (tac_prim)
A composite field that joins the area code and the exchange number to make it a unique record.

Zone Table (zone_table)

The Zone table specifies the rates to be charged for calls within a zone. Rates are assigned for time blocks of the day. Different rates may apply for weekday, Saturday or Sunday calls. A beginning effective date may be specified so that new rates may be put in ahead of time. Many area codes are mapped to single zones. Conceptually, zones are concentric billing circles around the local calling area. A single zone may contain many area codes.

```
============================== Area Code Table ================================
Area Code-Exch....[316]-[365]                     In-State....[N]
================================ Zone Table ===================================
Zone.............[  8] 926-1910 mi                Incr (Seconds) Rate  State Add
Begin Date........ 06/01/85           Minimum....   1    60      56.0     0.0
Holiday discount..  40.0              Additional.   1    60      39.0     0.0
Operator Charge...   0
Increment.........  60
            --- First ---  -- Second ---  --- Third ---  -- Fourth ---
             Time   Disc    Time   Disc    Time   Disc    Time   Disc
Weekdays:       0   60.0     800    0.0    1700   40.0    2300   60.0
Saturday:       0   60.0     800   60.0    1700   60.0    2300   60.0
Sunday:         0   60.0     800   60.0    1700   40.0    2300   60.0
```

Zone number (tzone_no)
The zone number joins a zone to an area code.

Zone date (tzone_date)
The date that this zone rate table becomes effective.

Zone description (tzone_desc)
A description of this zone such as mileage bands.

Increment (tzone_incr)
The seconds that must pass for each level of charge. Commonly billing is done either by the minute or for each 6 seconds. Enter 60 to bill by the minute.

Minimum call time (tzone_min)
Minimum number of increments to charge. If minimum call charge is for one minute and the increment is 60 seconds, enter 1.

Additional call time (tzone_add)
Number of increments to charge once the minimum is exceeded. Enter 1 if charges are assessed for each additional minute and if the increment is 60.

Minimum rate (tzone_rate_min)
Rate to be charged for minimum call increment. If you bill by the minute, the first minute may be charged at 55 cents per minute. Amounts should be entered in cents. Fractional cents are allowed.

Additional rate (tzone_rate_add)
Rate to be charged for additional call increments. If you bill by the minute, additional minutes may be charged at 45 cents per minute. Amounts should be entered in cents. Fractional cents are allowed.

In-state minimum adjustment (tzone_st_min)
Adjustments to be made to the minimum charge increment for in-state calls. Amounts should be entered in cents. Fractional cents are allowed.

In-state additional adjustment (tzone_st_add)
Adjustments to be made to the additional charge increments for in-state calls. Amounts should be entered in cents. Fractional cents are allowed.

Service charge (tzone_serv_chg)
Additional charge for operator assistance. Amounts should be entered in cents. Fractional cents are allowed. Operator assistance calls are defined as calls that have a dialing sequence that begins with zero.

Holiday discount (tzone_disc_hol)
Percentage discount for a holiday. There are 5 holidays currently supported. Discounts will be applied to calls on New Years Day, July 4th, Labor Day, Thanksgiving Day and Christmas Day. The holiday discount will be applied if it is greater than the usual call

discount. {labor day is the first Monday after September 1.} {thanksgiving is the previous Thursday from November 30.}
Time Blocks
Billing rates are determined based on the time of day and the day of the week. There are three categories for days. They are Saturday, Sunday and weekdays. Within these three categories, four different rates may be specified for time blocks.
First time (tzone_time1_wk, tzone_time1_sat, tzone_time1_sun) The beginning time to apply the midnight call discount. A beginning time is needed for Saturday, Sunday and weekday calls. This is entered in military time. If the associated discount goes into effect at midnight enter zero. This discount will be in effect beginning with the time enter in this field and ending with the time entered as second time. time >= time1 and time < time2.
First Discount (tzone_disc1_wk, tzone_disc1_sat, tzone_disc1_sun) The discount to be applied to calls made during time 1. A discount of 40 1/2 % is entered as (40.5).
Second time (tzone_time2_wk, tzone_time2_sat, tzone_time2_sun) The second time starts the day time discount. This discount will be in effect beginning with the time entered in this field and ending with the time entered as third time. time >= time2 and time < time3.
Second discount (tzone_disc2_wk, tzone_disc2_sat, tzone_disc2_sun) The discount to be applied to calls made during time 2.
Third time (tzone_time3_wk, tzone_time3_sat, tzone_time3_sun) The third time starts the evening discount. This discount will be in effect beginning with the time entered in this field and ending with the time entered as fourth time. time >= time3 and time < time4.
Third discount (tzone_disc3_wk, tzone_disc3_sat, tzone_disc3_sun) The discount to be applied to calls made during time 3.
Fourth time (tzone_time4_wk, tzone_time4_sat, tzone_time4_sun) The fourth time starts the late evening discount. This discount will be in effect beginning with the time entered in this field and ending at midnight. time >= time4 and time < 2400.
Fourth discount (tzone_disc4_wk, tzone_disc4_sat, tzone_disc4_sun) The discount to be applied to calls made during time 4.

Zone prim (tzone_prim)
A composite field that make the record unique. The field is a combination of the zone number and zone date.

Phone Run Record (phrun_rec)
A Phone Run record is added each time phone billing is done. It contains information relative to the group of calls billed within a date range.

phrun_no
Serial number uniquely assigned by the system when a Phone Run record is added.

phrun_beg_date
Beginning date of a billing period.

phrun_end_date
Ending date of a billing period.

phrun_post_date
Posting date of a billing period

phrun_chg_amt
Total amount charged for a billing period.

phrun_no_calls
Number of calls in a billing period.

INDEX