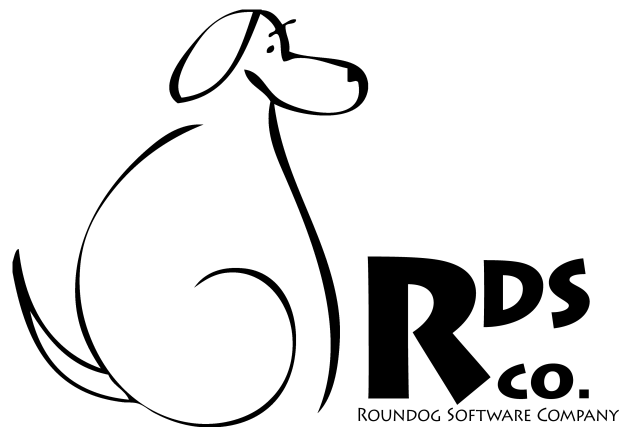


# *CHARLES*®

**College Housing and Residence Life  
for Education Systems**

**CONCEPTS MANUAL**





# Table of Contents

<b>1 Forward</b>	<b>6</b>
<b>1.1 THEORY</b>	<b>6</b>
<b>1.2 FUNCTIONALITY</b>	<b>6</b>
1.2.1 Acquisition of Student Data	6
1.2.2 Assignment Processing	7
1.2.3 Charge Generation - Billing and Payment	7
<b>1.3 BACK END RELATIONAL DATABASE MANAGEMENT SERVERS (RDMS)</b>	<b>7</b>
<b>1.4 CLIENT WORKSTATIONS</b>	<b>8</b>
<b>2 System Requirements</b>	<b>9</b>
<b>2.1 CLIENT WORKSTATIONS</b>	<b>9</b>
<b>2.2 DATABASE SERVER</b>	<b>9</b>
<b>2.2 WEB SERVER</b>	<b>9</b>
<b>3 Using the System</b>	<b>10</b>
<b>3.1 GETTING FAMILIAR WITH THE INTERFACE</b>	<b>11</b>
<b>3.2 AVAILABLE FUNCTIONS LIST</b>	<b>16</b>
<b>4 The CHARLES Theory of Housing</b>	<b>18</b>
<b>4.1 THE OVERALL HOUSING CYCLE</b>	<b>18</b>
<b>4.2 BEGINNING THE CYCLE</b>	<b>20</b>
<b>4.3 TRACKING PAYMENTS</b>	<b>22</b>
<b>4.4 MATCHING FOR FALL ASSIGNMENTS</b>	<b>23</b>
4.4.1 Some issues about assignments:	25
<b>4.5 ROLLING STUDENTS FROM FALL TO SPRING</b>	<b>25</b>
<b>4.6 PROCESSING FINAL SPRING CHECK OUT</b>	<b>26</b>
<b>4.7 STUDENT ELIGIBILITY - CONTRACTS AND TRACKING</b>	<b>26</b>
<b>5 Working with Assignments</b>	<b>29</b>
<b>5.1 ASSIGNING A STUDENT</b>	<b>30</b>
<b>5.2 MOVING STUDENTS</b>	<b>31</b>
<b>5.3 CANCELING STUDENTS</b>	<b>32</b>
<b>5.4 HANDLING ROOM CHANGE REQUESTS</b>	<b>33</b>
<b>6. Halls, Suites, Rooms and Bed Spaces</b>	<b>36</b>
<b>6.1 ADDING SPACES</b>	<b>37</b>

6.2	COMPLETING SPACE IDENTIFICATION	39
6.3	BUILDING A TERM'S SPACES FROM A PRIOR TERM	41
6.4	INDICATING LOTTERY PARAMETERS FOR ROOM RE-SELECTION	42
6.5	VIEWING A HALL'S SPACE USAGE OR ACTIVITY	42
7.	Viewing Student Records: Correspondence & Inquiry	44
7.1	STUDENT INQUIRY	44
7.2	eMAIL GENERATION	49
8.	Other Convenient Functions	53
8.1	HANDLING GROUPS	53
8.2	TROUBLE TICKETS	54
8.3	DATA RETRIEVAL	57
9.	Summer Housing	63
9.1	THE WEEKLY THEORY	64
9.2	MORE ON THE WEEKLY THEORY	65
9.3	LONG TERM SUMMER HOUSING	65
9.4	SUMMER HOUSING GROUPS	76
9.5	SHORT TERM SUMMER HOUSING	79
10.	Residence Life	82
10.1	ACADEMIC YEAR INQUIRY	82
10.2	CHECK IN & CHECK OUT	83
10.3	WORK ORDERS	85
11.	The Web Portal	90
11.1	WEB PAGE HEADERS	90
11.2	WEB PAGE GENERATION	92
11.2.1	Inserting a new web page	93
11.2.2	Adding data elements to your web page	95
11.3	WEB PAGE ACCESS	99
11.3.1	A Note About Web Portal Theory	100
11.3.2	Web Page Access Creates the Forms Page	101
11.3.3	Cautions for Forms Page Generation	104
12.	Systems Administration	105
12.1	TERM TABLES	105
12.1.1	Setting Up a Term	105
12.1.2	Rate Evaluations	107



12.1.3 Rates	108
12.1.4 Assignment Behaviors	110
<b>12.2 SYSTEM TABLES</b>	<b>111</b>
12.2.1 System Supplied Tables	111
12.2.2 Customizable System Tables	113
<b>12.3 OTHER SYSTEM ADMINISTRATION FUNCTIONS</b>	<b>126</b>
12.3.1 User Groups	126
12.3.2 Users	127
12.3.3 Data Import Utility	128
12.3.4 Institution Specifics	131
12.3.5 Version Updates	134
<b>Appendix 1</b>	<b>136</b>
<b>System Provided Tables</b>	<b>136</b>
Room Change Statuses (F_rcstatuses)	140
<b>Appendix 2</b>	<b>141</b>
<b>OPTIONAL FUNCTIONS</b>	<b>141</b>
a) - Meal Plan Functions	141
b) - Smoking Parameter	146

*CHARLES* is an interface to a specifically designed housing occupancy database. In other words, it is a window into your housing information system. Any interaction you need to make with your data, you can achieve through *CHARLES*, from simple information retrieval to generating assignments and charges.

The system is configured in the client-server model, where each desktop computer in your department runs its own copy of the program but all connect to a common database. An additional module can be added to your web server for student access to web forms that you design.

## 1.1 THEORY

*CHARLES* is a term (or semester) based system which means that student records and room assignments are handled on a semester by semester basis. Semester specifics, such as starting and ending dates, are stipulated by you so that semesters can be defined in the way best suited to your business practices.

*CHARLES* has been designed to work the way your institution does. It contains many user-configurable tables to help customize the system according to your needs. By using a high level of automation and behind-the-scenes processing, it shouldn't require a high level of database knowledge from your staff and will allow you to take better care of your student's needs rather than spending your time and energy handling your database. *CHARLES* is designed to be extremely user-friendly with a low learning curve.

## 1.2 FUNCTIONALITY

### 1.2.1 Acquisition of Student Data

Student data can be entered into the system through several different methods.

**Admissions Imports:** Typically an initial import file of incoming students (first year students) would be procured from your school's admissions office or data warehouse. *CHARLES* allows you to create an import template to map admissions data fields to *CHARLES* data fields for insertion.

**Manual Application Entry:** In the old fashioned way, applications were made of paper and mailed to students to fill out. With the advent of the world wide web and web-based forms processing, many institutions no longer collect applications and enter them manually. But the ability to insert an application into *CHARLES* still exists.

**Web-Based Applications:** By using the *CHARLES* web portal, you are able to design applications and forms for students to complete on the web. They are input directly into your database and become ready for processing by your staff.

## 1.2.2 Assignment Processing

**Automated Assignment Processing:** While particularly useful for incoming students, *CHARLES* allows you to define sub-populations of students to automatically assign rooms according to your set of parameters. If you want to assign a subset of students to a particular residence hall, or disperse them among a range of halls, the choice is yours. *CHARLES* will also match students to help find the best suited roommates according to your matching definitions.

**Individual Assignment Processing:** *CHARLES* still maintains the ability to directly assign or move an individual resident. *CHARLES* also allows you to define the filtering criteria for assignments (gender, smoking status, etc) and override these if desired.

**Room Selection:** Sometimes referred to as “lottery”, *CHARLES* gives you the ability to allow returning students to choose their roommates for the coming year and find the best, suitable space for them based upon your defined prioritization rules.

## 1.2.3 Charge Generation - Billing and Payment

*CHARLES* has an extremely flexible method of fixing charges for assignments and either passing them onto your school’s bursar system, or generating bills internally.

Each room in your system has a charge-type that is associated with an entry in a rate table. Each entry in the rate table has up to 15 different charge rates that can be assessed based on student attributes that you determine. For instance, if you have a particular group of students that need to be charged a higher or lower rate than others, *CHARLES* can handle it quite easily.

*CHARLES* also gives you the ability to assess estimated charges (if your assignments are not complete at the billing deadline), as well as issue miscellaneous charges for any number of housing related items.

If your institution tracks payments, *CHARLES* has two different methods of recording them. One method is to enter the amount paid. The other is a payment flag - for when (or if) you want to indicate that payment has been received but don’t care about the amount or an overall balance. (More on this in Chapter XX.)

## 1.3 BACK END RELATIONAL DATABASE MANAGEMENT SERVERS (RDMS)

*CHARLES* is designed to work with a number of Relational Database Management Servers (RDMS). It is currently certified to work with the following:

PostgreSql  
Frontbase  
MySql  
Oracle

We are currently working to certify it with these others:

MS-SQL Server  
Openbase  
SyBase

(If there are more database servers you'd like to see in this list, please contact customer support.)

RDMS usually run on their own machine, or on one of your institution's servers. Depending on the current load to existing databases, your institution's information technology department may allow you to create the *CHARLES* tables on an existing server. If not, it is very easy to set up a machine to run as your database server. Many of the RDMS software (upon which *CHARLES* interacts) are open-source, and therefore, free\*. (\* stipulations are stated under the GNU public license.)

## **1.4 CLIENT WORKSTATIONS**

Client workstations are simply desktop or laptop machines that you decide to use to connect to *CHARLES*. There are versions of *CHARLES* that are certified to work properly under the following computer operating systems:

### **Macintosh**

MacOS 10.4 thru 10.5.6

### **Windows**

Windows XP and Windows Vista

## 2 System Requirements

---

### 2.1 CLIENT WORKSTATIONS

Since there are three types of client workstations, we will deal with each of them in turn:

#### **Macintosh Workstations:**

Intel or PPC\* chipset running operating systems 10.4 through 10.5.x, minimum 1GB RAM, 100 MB of free disk space and an ethernet or wireless connection to the internet. (\* as of this writing, *CHARLES* is still certified to run using the PPC chipset. PPC support may be dropped in future versions.)

#### **Windows Workstations:**

Windows XP or Windows Vista workstation with most recent service pack installed, Intel Celeron or better chipset with a minimum of 1 GB RAM, 100 MB of free disk space and an ethernet or wireless connection to the internet.

### 2.2 DATABASE SERVER

Your database server can be any type of machine running any of the stipulated back-end Relational Database Management Server software. As long as it has an internet address and a user account (with password), *CHARLES* will be able to connect to it.

It will be up to your IT department to secure the database by providing appropriate firewalls and port designations.

### 2.2 WEB SERVER

If your institution intends to provide student access to application forms and integrated information over the web, you will need to have a designated web server to use as a gateway to *CHARLES*' web portal functionality. The hardware required for this web server must have the ability to run a version of *CHARLES*, meaning it must meet the same requirements as your client machines. It must also have the ability to run web server software such as Apache. Therefore, your web server can be any Macintosh, Windows or Linux machine capable of running the *CHARLES* software.

We also suggest that your web server be outfitted with a secure certificate, effectively putting your web address into the HTTPS space.

### 3 Using the System

*CHARLES* has been designed to have a familiar looking user interface. When the system starts up, it presents you with a window displaying your institution's information. This indicates that *CHARLES* has connected to your database (figure 3.1).

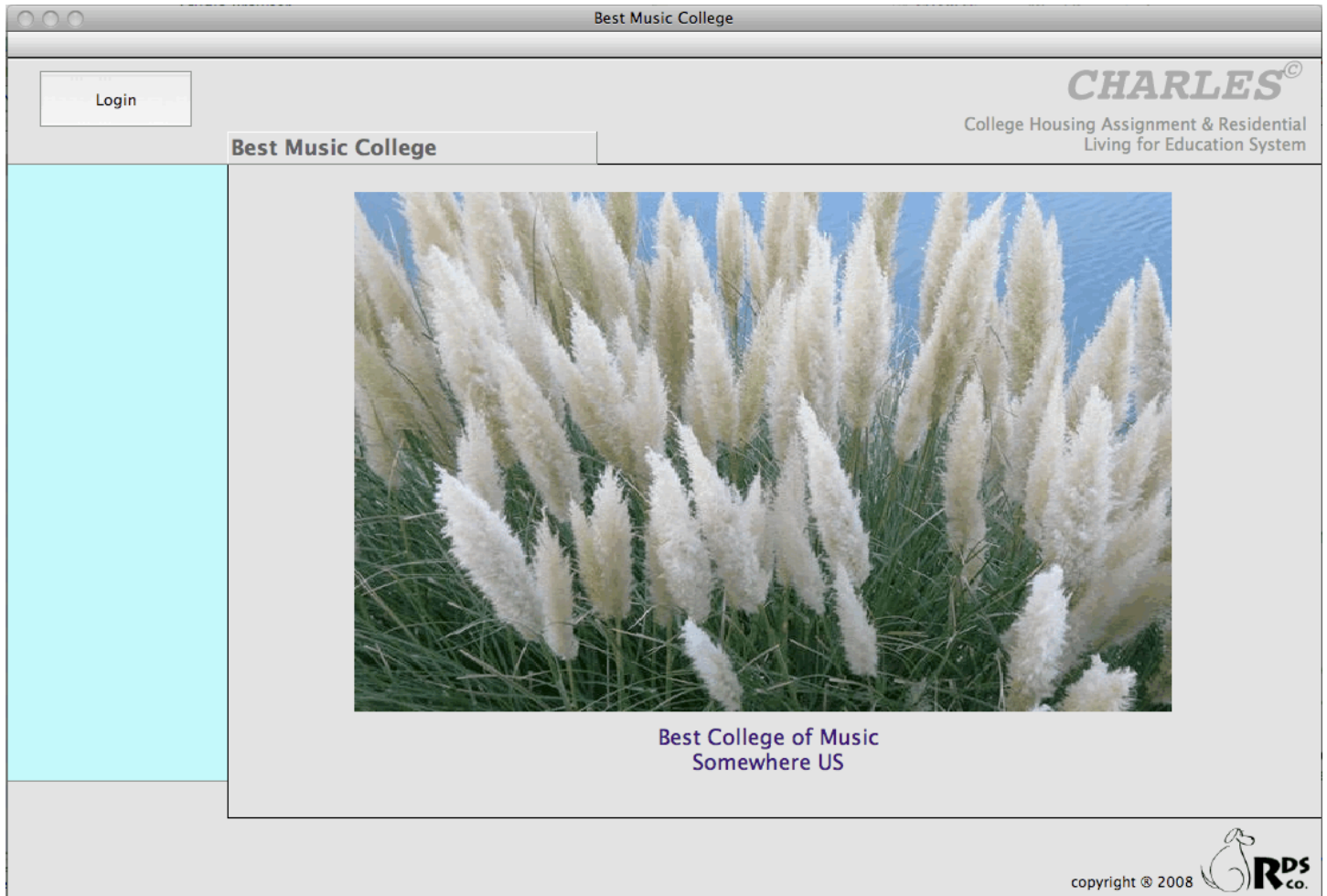


Figure 3.1 - *CHARLES* main window

In order to begin using *CHARLES*, each user must login to the database. Press the "Login" button to enter your user information (figure 3.2).

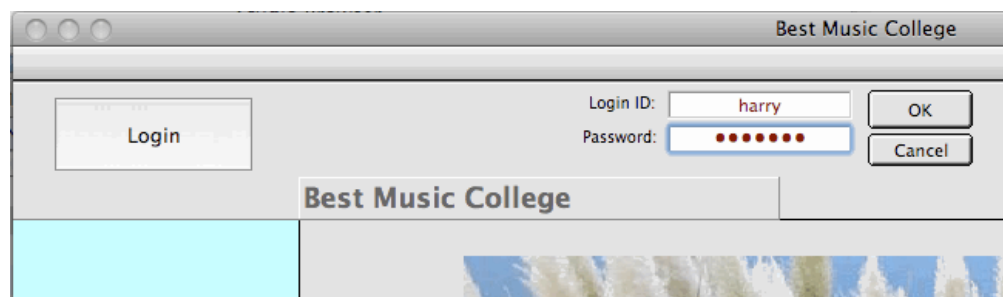


Figure 3.2 - Harry logs into your database.

### 3.1 GETTING FAMILIAR WITH THE INTERFACE

Once you are logged in, the window changes to indicate the current logged user, and shows a listing of all the functions he/she has access to. Figure 3.3 indicates the components of the *CHARLES* interface.

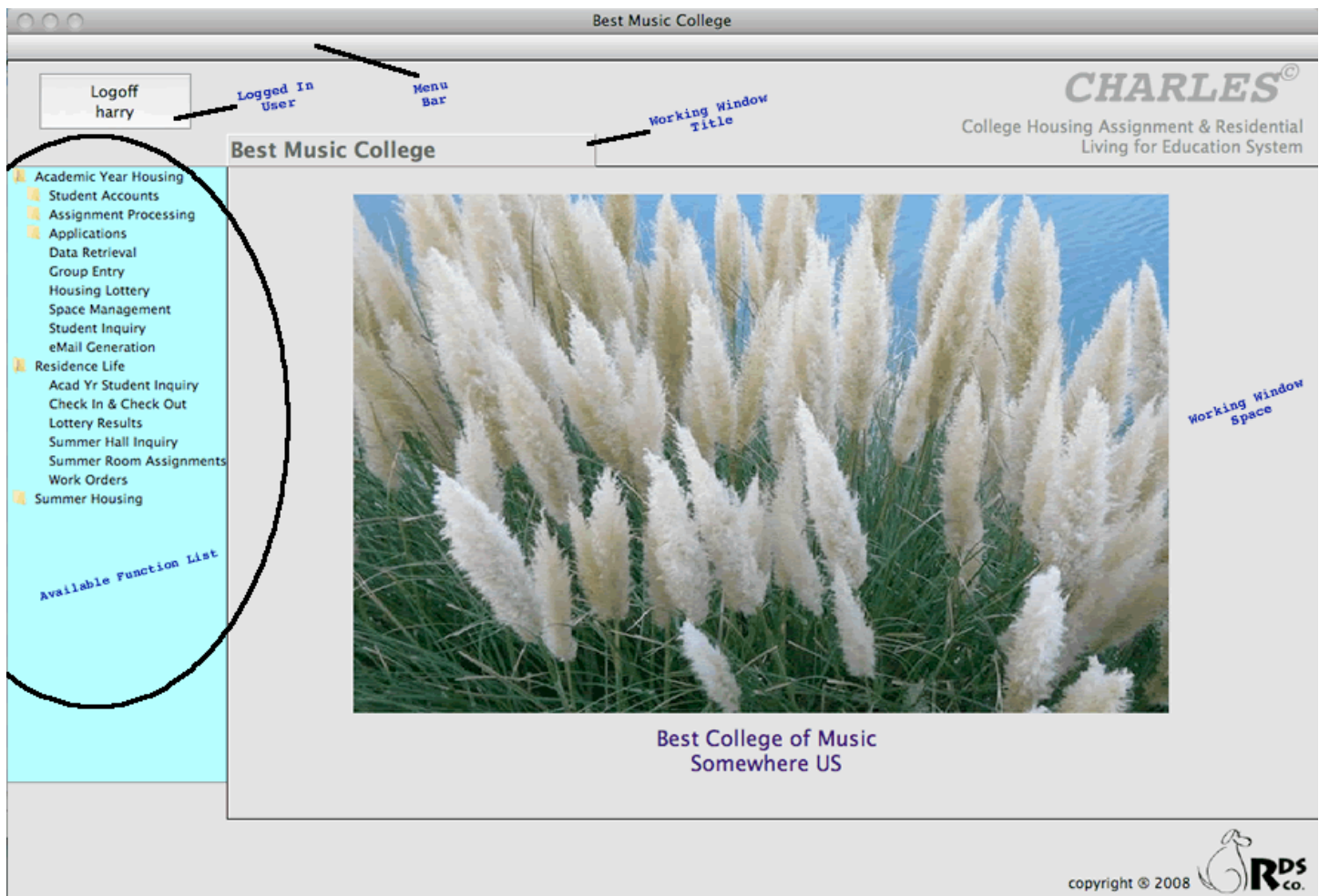


Figure 3.3 - Components of the *CHARLES* interface.

During your session with *CHARLES*, the interface remains constant allowing you to work within it. Note that the window is divided into four distinct sections:

- 1) **Function List** - Functions are assigned to you based on your particular job duties within your organization. Once you log in, you will be given access only to those functions that are available to you. These will appear in the blue section on the left-most part of the window. The function list is a hierarchical tree list and is organized according to the various areas of the system.
- 2) **Working Window Space** - the largest section of screen space on the window. This is where the various function windows appear and is where you will work on your data. This space changes for each function you select.
- 3) **Menu Bar** - for most functions you select, a menu will be installed in the menu bar containing functions and short-cuts to the behaviors of the installed working window.

- 4) **Logged In User Button** - is a button that allows you to log out of the system and indicates who the currently logged in user is.

Most working window functions are arranged with fields and buttons, allowing you to enter and manipulate data. For example, our fictitious user “harry” has access to Academic Year Housing Functions. He has expanded this grouping of functions to see what it is comprised of.

Suppose he would like to enter a student application. He would expand the Applications folder to select a type of application to enter (figure 3.4).

Harry sees Student Applications and decides that’s what he’d like to enter. Figure 3.5 shows what happens when he selects “Student Applications”.

We can see that when a function is selected from the function list, its window appears in the working window space of our interface (figure 3.5).

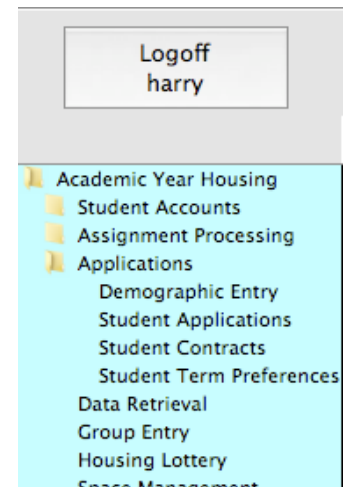


Figure 3.4 Harry expands the Applications grouping

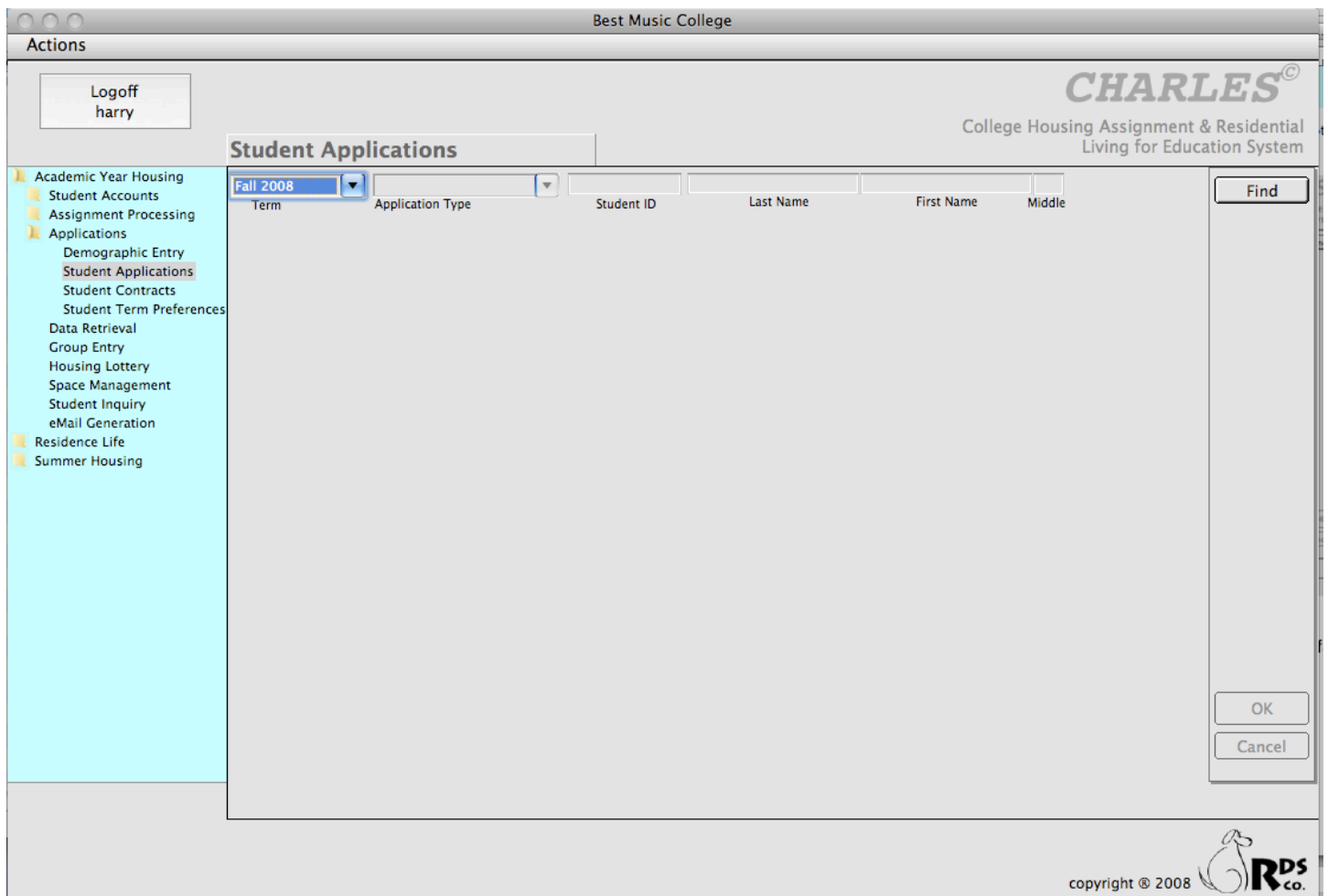


Figure 3.5 Selection of the Student Applications function changes the working window space.



Note that the selection of the function not only changes the working window space, but that it also indicates the name of the current working window, has installed a menu in the menu bar area called “Actions”, and (in this instance) has selected a default semester (or term) for us to use.

It should also be noted that most working windows in *CHARLES* behave similarly in that they contain a number of buttons along the right side (or bottom) of the working space, and they typically install an analogous menu of short-cut keys (usually named “Actions”) in the menu bar area.

Upon inspection of figure 3.5, we see that Harry does not have access to insert data. The only function available on this working window is “Find”. Harry’s system administrator or supervisor has decided that it is outside of Harry’s responsibilities to enter Academic Year Housing applications. (We will discuss user privileges further in the Administration Area.)

But suppose Harry does have access to enter student applications. Figure 3.6 shows the functions available in that scenario.

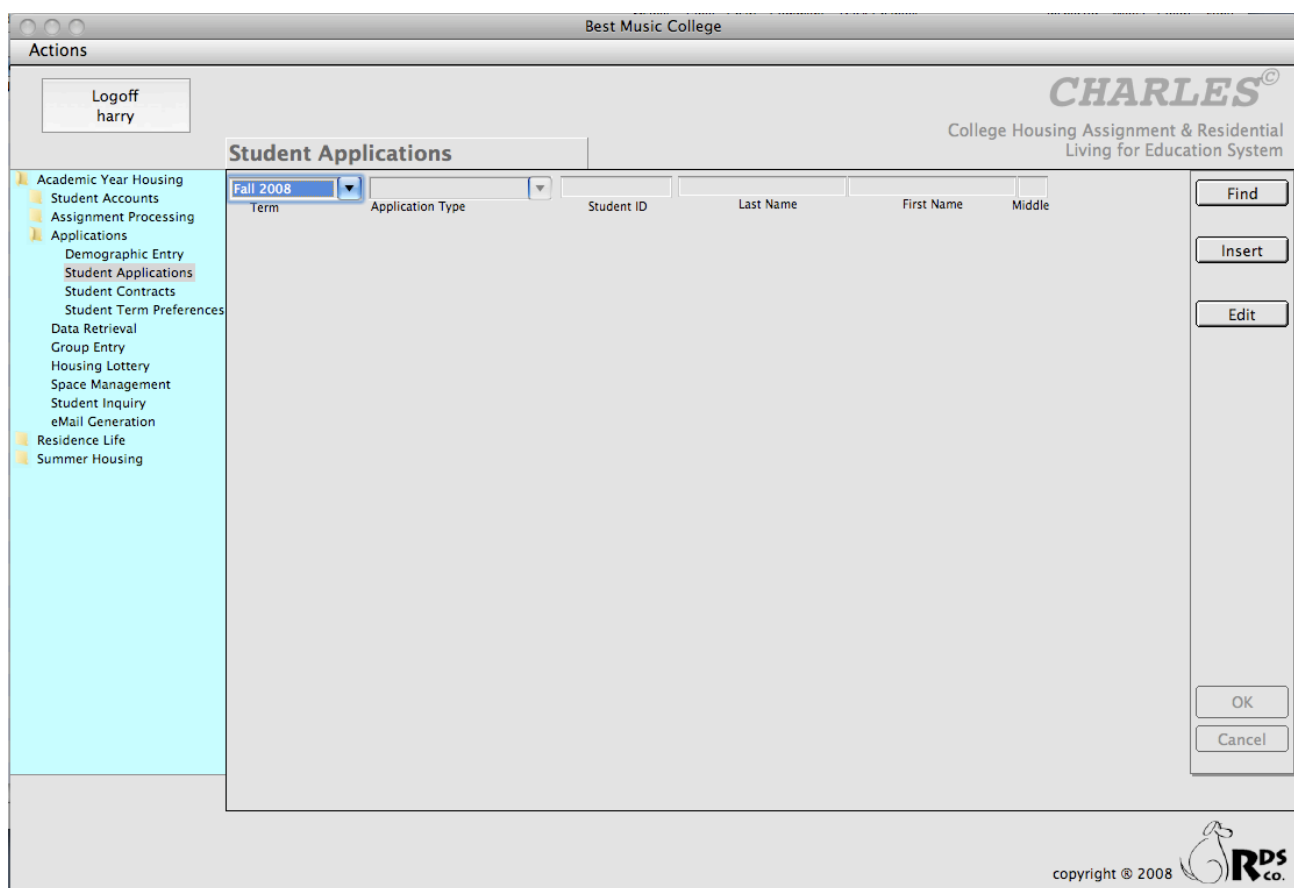


Figure 3.6 Harry has full access to Find, Insert and Edit academic year student applications.

In order for Harry to begin entering an application, all he has to do is press the “Insert” button and begin filling in the data fields.

The student application layout that appears in the working window space arranges the data fields according to the type of application that you select in the application type drop list.

In general, while entering or modifying data in the *CHARLES* interface, users initiate an action by pressing the button that corresponds to their desired task, and then tab from field to field, entering or

changing the data as they go. Note that when entering data, the data elements (or fields) on the window change color (from gray to white), indicating which elements are able to be entered or modified.

Figure 3.7 When entering data, the OK and Cancel buttons become active so you can finish or cancel the entry.

While data is being entered or modified, the OK and CANCEL buttons become enabled, allowing you to finish (OK) or cancel your entry. When you press the OK button, the system will attempt to verify your entry, making sure that all of the required data elements have been completed. *CHARLES* will not let you complete your entry until all data has been checked and verified for completeness. (figure 3.8)

Once an entry has passed its test for completeness, it is actually added to your database. Figure 3.9 shows a completed application that has passed the completeness tests.

Once the data is entered into the database, the window resets itself, disabling the OK and Cancel buttons, resetting the data field background colors and resetting the other available function buttons. The window is ready for the next entry.

Figure 3.8 *CHARLES* stops entry upon pressing OK if data is required but missing.

Finding data is handled in a similar manner. When you press the “Find” button, the indexed fields become highlighted and enterable. You may enter a value in any of the available “find” fields. Then

press OK. If the record is found in the database, it is displayed on the window. If not, you will hear an error tone and the window is reset. (Figure 3.10)

Figure 3.9 A completed application. The working window is ready for the next entry.

In general, all of the working windows in *CHARLES* behave in this manner, enabling and disabling data fields and functions as you interact with your data. *CHARLES* remains extremely watchful that data is entered in a consistent and complete manner to help insure your data's integrity.

We will deal more with the individual data processing windows, as well as assignment processing, space management, reporting and correspondence functions in later chapters.

One interesting feature in *CHARLES* is its ability to remember the last person you looked up. *CHARLES* carries that person's ID from working window to working window so you don't have to continually "find" every time you select a new working window (or function). This makes jumping from one function to another extremely quick and easy.

Figure 3.10 Performing a "Find" on the Student Application window.

## 3.2 AVAILABLE FUNCTIONS LIST

As we've shown in the preceding section, entering information into *CHARLES* is fairly straightforward. Although large amounts of data can (and perhaps should) be insert by utilizing Data Import functions, we will concentrate on individual entries in the working window space during this chapter. This should help you become more familiar with the system's data handling capabilities, and help you learn where to find things

*CHARLES* contains a number of system tables that contain commonly used pieces of data to allow you to categorize your student entries. These tables contain things like state codes, country codes, address types, school codes, degree codes, application types and many other tags that help you to specifically identify your population. Some of these system tables are provided upon delivery and cannot be modified (since they are required for the proper functioning of the system). Others can be modified by your system administrator, and we will deal with this in the systems administration chapter. We also provide an appendix indicating the system tables and their values.

We have already seen Harry log into the system in section 3.1. He has entered an application and has been able to find students using the Student Application function. But we notice from his function list that there are a lot more functions he can undertake.

Every person who logs into *CHARLES* is presented with their available functions list. It is important to note that this function list corresponds with the person's associated user group and may not include all of the functions available in the system. The available functions (the ones that appear in the list) are determined by your supervisor and/or system administrator. Each system user is given a *CHARLES* login id and password, which is then tied to a specific group. It is the group that contains the permissions for what appears in the function list. Therefore, anyone who belongs to your group has identical permissions.

Remember (from section 3.1) that Harry did not have access to enter applications in the Academic Year Housing Student Application. He could, however, "find" students on that working window. This occurred because Harry (or more correctly, the group to which Harry belongs) had only "limited" access to Student Applications. In order to give Harry the proper access to enter a student application, the system administrator had to go into Harry's group and increase the access level for applications.

In short, your function list is determined by your group and each function you see in the function list contains a level of permission. You should not worry about this unless you find that you need access to some function that you cannot see. If that happens, you should contact your supervisor and/or system administrator so they can determine whether granting you the privilege you seek is appropriate.

*CHARLES* functions are organized into several sections to help make things easy to find. Depending on your system's configuration, your function list could contain the following list of "high level" function items:

- 1) **Academic Year Housing** - contains all functions pertaining to what is probably the bulk of your institution's room assignments, accounting functions and data tracking needs.

- 2) **Residence Life** - contains functions to allow your residence halls access to student information, data rosters, check-in and check-out processing and other residence hall data.
- 3) **Summer Housing** - if your institution treats summer differently than any other academic year terms, you will see this function grouping.
- 4) **Systems Administration** - only systems administrators should be able to view this grouping of functions.
- 5) **Web Portal** - functions which allow you to design web pages and populations of students who can access those pages.

The function list is hierarchical in nature. Each of the “high level” groupings contain a number of lower level categories and/or functions. In order to expand the high level function to see what functions are included there, either click the folder icon on the left or double-click the name of the function. (see figure 3.11).

Figure 3.11 shows a fully expanded Academic Year Housing function with all sub-groupings also expanded. This listing may be different in your system according to your institution’s configuration, but you can see how the system is organized.

When you click on a function (one without a folder on the left), the working window changes allowing you to accomplish tasks belonging to that function.

We’ve already seen an application get entered with the Student Applications function. There are other application functions that we could access as well. We could modify some demographic information, the student contract or the student’s term preferences.

Some of the functions do not fall inside a subcategory. These are functions pertaining to the overall category of Academic Year Housing, but exist on their own and did not require further categorization.

Other high level functions have the same general set up and we will spend a lot of time explaining how and why these functions exist. Before we do that though, we should discuss the *CHARLES* theory of housing and why this system is organized in this way.

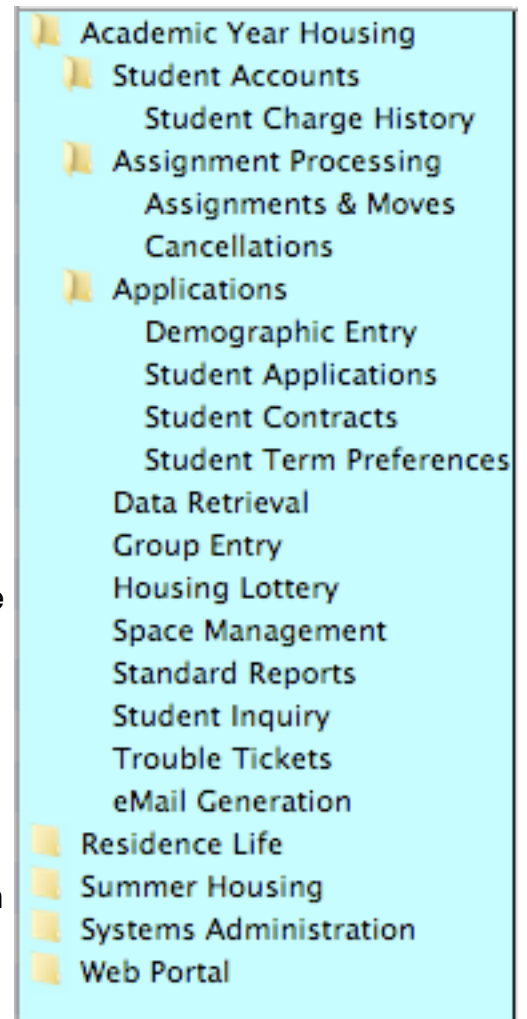


Figure 3.11 A fully expanded Academic Year Housing function

## 4 The *CHARLES* Theory of Housing

*CHARLES*, in its infancy, was developed at a large urban university at a time when there were no good methodologies or databases designed for handling college housing functions. The basic table structure was tailored around some shared resources, primarily the student information system (or SIS) that was in effect at the time.

Initially, it was an academic year based system, requiring a new database to be initiated at the start of each fall semester. Summer semesters were handled with another whole separate database. Over time, this became unwieldy and the term-based concept was developed. Once we had term-based assignments, we were able to store historical data and perform additional functions pertaining to charge generation, billing and data sharing with other offices on campus.

### 4.1 THE OVERALL HOUSING CYCLE

The theory of *CHARLES* is based upon the presumption of a housing cycle that starts in the spring when we begin to solicit applications for fall housing, and ends with final assignments and check in for the spring term. Sandwiched into this cycle is whatever your institution does with its spaces during the summer months. Figure 4.1 shows a representation of this cycle

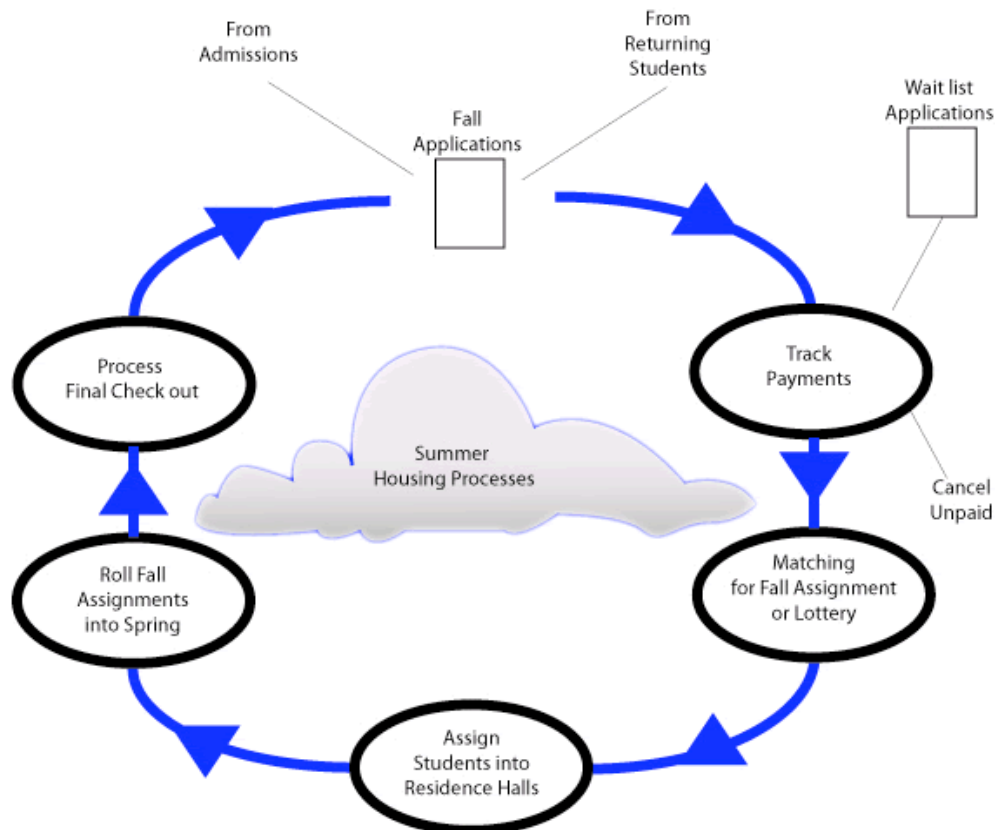


Figure 4.1 - A simplistic representation of the Housing Cycle

To jump in, let's begin with students submitting applications for housing for the fall term. These can be new students (Freshmen and Transfer Students) or returning students who were in residence during the prior year. Applications generally contain student's preferences - hall choices, meal plan choices,

roommate requests and other pertinent information. Applications may also require some payment, usually in the form of a housing deposit or an application fee (for processing). It may also simply require that a payment be made to your institution by a specific deadline date, perhaps requiring students to agree to some terms before they can submit an application.

The institution must track receipt of payments to determine eligibility (you probably wouldn't want to assign housing a person who has not either paid or promised to pay). Once the payment deadline has been reached, you know the population of students who will be needing room assignments.

The next step is to process the assignments. Returning students, being more familiar with your institution, its residence halls and room configurations, are typically given the option to participate in a room re-selection process or a lottery to determine their room assignments for the fall semester. Since they have been in residence for some period of time already and have probably made some friends they would like to have as roommates, they should be allowed to group themselves and choose their own location for the upcoming year. Often, this re-selection process happens well in advance of the close of the spring semester. This would allow the institution to inform returning students of their fall assignments before they go away for the summer, and to allow the institution to fully know what spaces they have to work with to accommodate the new students, and any other populations requiring fall housing.

During the summer months, new students have to be matched with one another and given room assignments, usually determined by their chosen application preferences.

While the assignment process progresses, students must be charged for their housing. This could occur either during the assignment process (in which case you might need to assess an estimated charge for students not yet assigned) or once all assignments have been made.

When the assignments are complete, the housing office notifies students of their check-in date and location and gives out any additional information concerning their impending stay at the institution. Near the end of the fall semester, probably around the time of finals (or the Thanksgiving break for those in the US), the spring semester should be starting. Students staying on for the spring, and their assignments from the fall, must be rolled into the spring semester. Applications for new spring admits should be sent out (or activated online) and those applications processed and assigned before spring semester check in. Your usual billing must also be handled, as well as any payment issues your institution requires.

Usually, while spring is underway, sometime shortly after spring check in, new applications for the fall must be made available, as well as the start up of any summer processes your institution has. Both summer and fall applications should be made available once spring check in is complete.

And it happily begins again and again.

The reason we've outlined this cycle is to illustrate why *CHARLES* is organized the way it is, and to show you how it can help you at each step along your housing business cycle.



## 4.2 BEGINNING THE CYCLE

To begin at the beginning, we should start our cycle with the acquisition of new student data. In most organizations, new students come to us from an admissions organization. Arrangements are made to receive a file of new student information in the form of a csv or tab delimited text file. This information should contain specific demographic and school information so we can use it to seed our database.

*CHARLES* contains an import function specifically designed to process files of this type. Once you know the layout of the admissions file, *CHARLES* allows you to set up a template for the regular importation of this data (Figure 4.2).

Order	Field Name	Table Name	Match
1	s_id	F_studentdemo	<input checked="" type="checkbox"/>
2	s_name	F_studentdemo	<input type="checkbox"/>
3	s_gender	F_studentdemo	<input type="checkbox"/>
4	sa_sid	F_addresses	<input checked="" type="checkbox"/>
5	sa_type	F_addresses	<input type="checkbox"/>
6	sa_street1	F_addresses	<input type="checkbox"/>
7	sa_street2	F_addresses	<input type="checkbox"/>
8	sa_city	F_addresses	<input type="checkbox"/>
9	sa_state	F_addresses	<input type="checkbox"/>
10	sa_postcode	F_addresses	<input type="checkbox"/>
11	sa_country	F_addresses	<input type="checkbox"/>
12	Placeholder	Placeholder	<input type="checkbox"/>
13	term_stutype	F_termstudent	<input type="checkbox"/>
14	term_id	F_termstudent	<input checked="" type="checkbox"/>
15	term_term	F_termstudent	<input type="checkbox"/>
16	asnative	F_assignment	<input type="checkbox"/>

Figure 4.2 - The Data Import function ready to import a freshman file.

*CHARLES* inserts this data into the proper tables according to your template. It also sets an initial assignment status of IN or “inactive”. An inactive assignment is one in which nothing has been done. We’ll talk more about assignment statuses, but for now, simply be aware that the cycle progresses via the tracking of assignment statuses.

If you would like to collect student applications over the web, you will have to design a web page using the *CHARLES* web portal (or any other type of webpage tool) so students can submit the information you require to properly categorize them for assignment. A nice thing about the *CHARLES* web portal is that you won’t need a lot of familiarity with web technologies or HTML in order to design a web page to collect student information and preferences. *CHARLES* allows you to define populations (for instance, new freshmen who are currently inactive in the fall term) and give those populations access to specifically created web pages, like a student preferences application.

Your system administrator can assign access to these web portal functions for any group your institution desires. Figures 4.3 and 4.4 show the working window functions for web populations and web pages. The web portal functions will also be discussed further in another section.



**Web Page Access** Life for Education Systems

Population Selection: New Freshmen Population Name: New Freshmen Term: Fall 2008

**Population Definition**

SQL Definition: Select \* from F\_assignment,F\_termstudent where term\_id=asnsid and asnterm='083' and term\_term='083' and asnactive=1 and asnstatus='IN' and term\_htype='F' and term\_stutype='UG'

☐ Override SQL

Forms Page Header: Forms Header

☐ Contract Check  
Check this box if this population requires validation of their current housing contract.

Web Pages:

- ☐ FRquestions
- ☐ NewGrads
- ☐ error
- ☒ frapp
- ☐ unactual

Pages Granted: frapp

Portal Download Forms: (links to pdf documents and web page instructions)

<li><a href="https://127.0.0.1/freshmaninstructions.pdf">Freshman Instructions</a></li>

This window allows you to build a forms page for each population you define. Enter the population name, then modify the definition of your population. Choose the web pages you'd like to make available to the population.

Describe

Insert

Edit

Delete

View Forms Page

OK

Cancel

Figure 4.3 - Web Page Access function allows you to define populations and grant web page access to them..

**Web Page Generation** Life for Education Systems

Form Name: frapp Header: Freshman Application Header Term: Fall 2008

☒ Effect Assignment Status? ☒ Page is active

Successful submission sets to: UN Unassigned

**Embedded Data Elements**

Order	Element Name	Table Name
1	s_id	F_studentdemo
2	s_firstname	F_studentdemo
3	s_lastname	F_studentdemo
4	sa_street1	F_addresses
5	sa_street2	F_addresses
6	sa_city	F_addresses
7	sa_state	F_addresses
8	sa_postcode	F_addresses
9	term_rezhal1	F_termstudent

(double-click line to remove element from form)

**Form Action**

```
</table>
<input name="OmnisClass" value="RT_forms" type="hidden">
<input name="OmnisLibrary" value="Housing" type="hidden">
<input name="OmnisServer" value="S912" type="hidden">
<input name="FormName" value="frapp-submit" type="hidden">
<input name="Snetid" value="," ipv_row.Snetid," type="hidden">
```

☐ Override HTML

**HTML**

```
<script language="JavaScript">
var submittimes = 0;
function submitonce() {
submittimes++;
if (submittimes == 1) {
document.form.submit();
}
else {
alert("You may only submit the form once. Refresh if you are having
```

**Confirm Message**

**Form Link (to appear on the Forms Page)**

```
<li><a href="http://localhost/cgi-bin/nph-omnis/cgi/?FormName=frapp-send&OmnisServer=S912&OmnisClass=RT_forms&OmnisLibrary=Housing&Snetid=",inetid,">2008-2009 Freshman Housing Application</a>
```

Find

Insert

Edit

Delete

View Web Page

Add Data Element

Copy WebPage

Next

Prior

OK

Cancel

Page SQL

Select

s\_id,s\_firstname,s\_lastname,sa\_street1,sa\_street2,sa\_city,sa\_state,sa\_postcode,term\_rezhal1 from F\_addresses,F\_studentdemo,F\_termstudent where s\_id=sa\_sid and s\_id=s\_id and s\_id=term\_id and term\_term='083'

Figure 4.4 - Web Page Generation creates web forms with embedded data fields from your database tables.

Of course, there are other methods of collecting student information, from manual (hand written) data forms and data entry, to installing another web-based system of data acquisition. Your systems administrators may be fully versed in web technologies and may wish to access your database using

any number of web methods. We simply offer these tools to jump start your interaction with your students.

### 4.3 TRACKING PAYMENTS

CHARLES is able to track payments in two different ways: either by logging the actual payment amount or by checking a flag to indicate payment received. It is up to your institution which is more appropriate.

Payments are a way your institution can ascertain whether applicants are serious about living in student housing and can be used as a mechanism to advance payment for the term, as well as minimize attrition through non-payment.

Figure 4.5 shows the Student Inquiry function with a payment being processed. Of course, this is a single payment being processed manually. In many cases, you would probably want to receive a file of payments from your bursar and import them using the data import functions.

The screenshot displays the 'Student Inquiry' application window. At the top, a header bar contains the title 'Student Inquiry' and the logo 'Life for Education Systems'. Below the header, a form displays student information: 'Fall 2008' (selected term), 'U11-22-3333' (Student ID), 'Kelley, Grace R.' (Name), 'Female' (Gender), 'JAN 1 1989' (Birthdate), '0511' (Grad Date), and 'NetID'. A 'Find Student' button is located on the right. Below this, the 'Addresses' section shows a table with one entry: 'P' (Type), '111 Monaco Way' (Street), 'Punksataunee' (City), 'PA' (State), '19222-' (Zip), 'US' (Country), and '(645) 555-1122' (Phone). A 'Find Suite' button is also present. The main area is divided into tabs: 'Current', 'History', and 'Miscellaneous'. The 'Current' tab is active, showing a 'Charges' table with columns 'Code' and 'Description'. One charge is listed: 'Est-GDNG' with description 'Est'. A modal dialog box titled 'Enter Payment' is overlaid on the screen. It contains a 'Payment Type' dropdown set to 'CHECK', a 'Payment Amount' field with '\$ 300.00', a 'Payment Description' field with 'Housing Deposit - Fall 2008', and a 'Payment Received' field with 'APR 15 2009'. 'OK' and 'Cancel' buttons are at the bottom of the dialog. Below the dialog, the 'Payments' section shows a table with columns: 'Code', 'Description', 'Paid On', 'Processed On', 'Processed By', and 'Amount'. A 'Payments' table is visible with one entry: 'DEC 19 2008' (End Date) and '650.00' (Amount). An 'Enter Payment' link is visible. At the bottom right, there are 'OK' and 'Cancel' buttons for the main window.

Figure 4.5 - Manually processing a payment for a student.

If you elect to do so, you can also simply track that a payment has been received (regardless of the amount). This, again can be handled either through a data import from your bursar, or manually by checking a field in the student's term data window. Figure 4.6 shows an example of that manual process.

Once your payment deadline has passed, you will know for certain all of the individuals who are serious about living in student housing for the upcoming term.

**Student Term Preferences** Life for Education Systems

Term: **Fall 2008**

Student ID: **U11-22-3333** Name: **Kelley, Grace R.** Net ID:  Legal Sex: **F** Birthdate: **JAN 1 1989**

Student Type: **GD - Graduate Stu...** Housing Type: **NG - New Grad St...** Year in School: **1** Confirmation Number:  Lifestyle Preference:

Emergency Contact: **Kelley, Pa** Relationship: **FA** Emergency Phone: **(645) 555-1122** Alternate Phone: **(645) 555-1122**

Degree Code: **MA** Major Code:  Application Source:  Meal Preference:

Cell Phone Number:  Comments: **another test**

☐ Squat Current Space Gender Identity: **Legal Sex** Gender ID:  Preferred Name:

Group Affiliation:  Rate: **2** Roommate Requests:

	SID	Name
1.	<b></b>	<b></b>
2.	<b></b>	<b></b>
3.	<b></b>	<b></b>

Primary Email Address: **grace@aol.com** Alternate Email Address:

App Received: **AUG 1 2008** App Entered: **AUG 1 2008**

Buttons: **Find**, **Edit**, **OK**, **Cancel**

Figure 4.6 - Indication of a payment received in the student's term preferences record.

During the payment period, there are a number of built-in reports to help you gauge the progress of the final outcome. These can be found in the Standard Reports function and are listed under the Stats reports, Alpha Rosters and the Exports subcategories.

#### 4.4 MATCHING FOR FALL ASSIGNMENTS

New students will need to be matched up with one another and assigned to their rooms. *CHARLES* has a way to do this through the use of the Assignment Processing functions.

In the menu bar, when working with the Assignment and Moves function, you will see an item called Actions - a menu containing the functions available on the working window. This menu contains a couple of additional functions - one of them being Auto Assignment. (Figure 4.7)

Auto assignment allows you to define populations of students for automatic assignment. In other words, each population you define will contain the people from your definition, it will group them according to their characteristics, and find spaces for them to live in. A picture of the population window appears in figure 4.8. There will be more about this in the ***Working in CHARLES*** manual, but for now, we wanted to point out this important step in the cycle.

You can define as many populations as you need. You can also elect to match your population to itself or to another population.

Once your population is defined, the process is to activate it, (which sets all of its members to an assignment status of 'AA') and run it (which attempts to find spaces for everyone).

A final window is displayed at the end of the process for your final approval before actually committing the assignments to the database. (Figure 4.9)

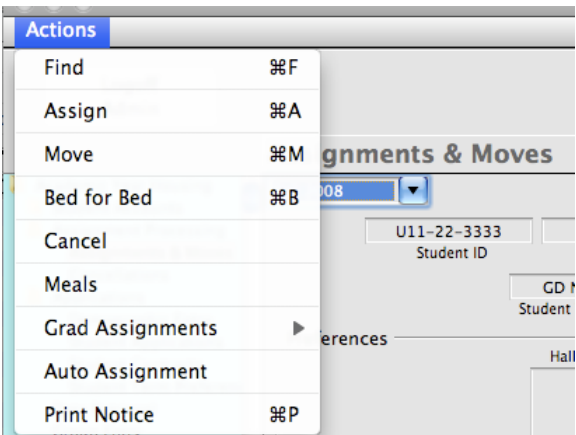


Figure 4.7 - Actions menu in Assignments and moves allows access to Auto Assignment function.

**Population Definition**

Term: 083, Priority: 1.0, Population Description: Freshman Assignments, NO Activated

**Student Attributes**

Student type: Undergraduate ...  
Lifestyle: APARTMENT ST...  
Housing type: Freshman  
School: (none selected)  
College: Undergrad Busi...  
Early Decision: ☐  
Match on questionnaire: ☒  
Low Cost: ☐  
Gender: ☐ Female, ☐ Male

**Hall Choices**

Cushing, hall choice 3, hall choice 5, hall choice 2, hall choice 4, hall choice 6

**Room Attributes**

Hall: ☐ Albright, ☐ Baldwin, ☒ Cushing, ☐ Wilder  
Suite type: (none selected)  
Hall type: Traditional Dormitory  
Charge type: (none selected)  
Room program:  Set

**Potential Problem Cases**

Comments section contains: asthma, Set  
☐ There are comments in student's record  
Student type: (none selected)  
Housing type: Freshman

**SQL extraquery text for components**

and asstatus='UN' and asnterm='083' and term\_term='083' and term\_htype='F' and term\_lifestyle="" and term\_college='UB' and term\_rezhal1='Cushing' and (term\_matchq is not null and term\_matchq<>"")

**SQL extraquery text for room search**

where texpireddate is null and troomuse='STU' and troomterm='083' and (troomprogram is null or troomprogram="") and ((tbuilding='Cushing')) and thalltype='TRAD'

☐ Print Population

OK, Cancel

Figure 4.8 - Auto assignment population defined to match and assign freshmen students to a hall named Cushing.

There is a lot of granularity that can be added to each of your population definitions. You may even want to continually modify one population definition until all of your students have been matched and assigned. The way you use these functions is up to you.

Your institution could also handle assignments in a manual fashion, although that could be quite cumbersome and take a long time.

#### 4.4.1 Some issues about assignments:

While it is true that handling assignments in CHARLES is not difficult, you should, however, be aware of some things that occur behind the scenes during the assignment process.

- 1) Assigning students to rooms will adjust any rate that has been assessed. This means that if the student has already been charged, that charge will be adjusted in favor of the new rate. This occurs at the time of the assignment.
- 2) Assigning students to rooms causes the student's assignment status to change to a status of "AS", indicating a current assignment. Prior assignments are flagged as "inactive".
- 3) Students processed through the Auto assign functions are assigned for the entire semester, from start to finish. Therefore, auto assignment cannot be used to process moves or re-assignments.

Name	Gender	Room #	Type
Walto, Jose D.	M	CU-0002A-1	RU
Walton, David J.	M	CU-0002A-2	RU
Moose, Bullwinkle T.	M	CU-0002B-1	RU

Confirm Kickout ☒

Print kickouts      Assignments to be made: 3

Print results      Cancel      Approve

Figure 4.9 - A sample Auto Assign results window, ready for approval.

## 4.5 ROLLING STUDENTS FROM FALL TO SPRING

Near the end of the fall term, it becomes necessary to move your fall students and their assignments into the spring term. Even though the spring term is an entity unto itself, most institutions do not have separate application and assignment functions pertaining to spring. We have, therefore, decided to create a method called "rollover" to handle the population of the spring term.

Creating the actual term in the database term table is the responsibility of your system administrator. Once the term has been created, and rate tables have been set up, it is time to roll your students and assignments into the new term.

It is not the intent to go into all the inner workings of the rollover procedure here. We simply state it as a step in the cycle. In essence, it is up to the institution to decide when this rollover should happen. Since most institutions like to be able to bill students for the upcoming term before they leave for the holiday break, the rollover should occur as close to the holiday break as possible.

There are a few things to be aware of once the rollover has occurred:

- 1) When rollover occurs, it takes a snapshot of the fall term and replicates it in the spring.
- 2) After rollover, both terms are active. The current term is still the fall term. Therefore, changes that are made in the fall term after rollover, do not get replicated in spring. If a change must be made in the fall, it will have to be done in spring as well. Because of this duplication of effort involved in changes after the rollover occurs, it is best to roll over as close to the end of the fall semester as possible and be sure to inform your assignment processing personnel to be sure to make changes in both fall and spring terms.
- 3) The rollover procedure does not generate charges. Charges should not be activated for the spring term until after the rollover from fall has occurred. (We will deal more specifically with term handling, rollover and charge generation in the Systems Administration discussion in the **Working in CHARLES** manual.)
- 4) The rollover will not close the term automatically. Terms must be closed by the system administrator when that person is satisfied that the final state of the term has been reached. Closing a term forces the system to examine student contract usage and closes or expires contracts that have been exhausted. Usually a fall term should be closed before allowing students to reapply for the coming academic year.

## 4.6 PROCESSING FINAL SPRING CHECK OUT

The final stage in the cycle, before it all begins again (actually it has already begun by this time) is to process final spring check out records. When students actually check out from the spring term, we can close out the term and end the academic year.

Once again, closing the term means that the system will examine student contracts for closures and expirations. We will discuss this more in the next section.

We have now seen the entire academic year cycle. *CHARLES* has two different ways to handle summer housing. We will look into these in chapter 6. Before getting to that, we should look some more at some of the *CHARLES* theory, particularly as it pertains to eligibility and tracking.

## 4.7 STUDENT ELIGIBILITY - CONTRACTS AND TRACKING

Another aspect of the theory of *CHARLES* is that of housing eligibility. In order to allocate space, there is a concept built into the *CHARLES* system dealing with a housing contract. *CHARLES* assumes that when a student applies for housing, they are agreeing to a contract (of your own design) which binds the applicant to your housing rules and regulations.

In order to track and control student eligibility, these contracts are attached to the housing type of students and are assigned when the application is received in the system. For instance, a new freshman applicant will not be allowed to submit their application until they agree to the tenants of the contract. Once they agree, they will be assigned to a contract, allowing for some number of semesters your institution believes is appropriate. For purposes of illustration, let's assume freshman will be given 8 semesters of housing to allow them to complete all the course work to finish their degree.

*CHARLES* tracks contract use by attaching contract behavior to changes in the student's assignment status. You are able to assign this contract behavior in your system. Figure 4.10 shows the systems administration window for regulating contract behavior.

To illustrate, let's assume that we are assigning a new freshman. When we accept their application, CHARLES gives them a new contract with 8 semesters on it. There are no semesters used. When that freshman receives their assignment, their assignment status changes from unassigned (status "UN") to assigned (status "AS"). We should have a behavior entry in our behavior table to inform CHARLES about what to do with the contract. Figure 4.10 shows a sample of what to expect that behavior to be.

Figure 4.10 -Assignment Behaviors are set up in every semester to govern contract statuses and track eligibility.

As you can see in figure 4.10, assignment behavior is determined for each term. In the above illustration, whenever a person's assignment status changes from unassigned to assigned in the fall 2008 term, the counter on the contract will be incremented. In fact, since these behaviors are defined by term, the behavior is limited to the term. Contract behavior across terms is handled by looking at only current student contracts.

Contracts in CHARLES have two statuses you should be aware of. At any time, a student can only have one current contract so contracts contain a currency flag. Also, contracts can be either active or expired. An active contract is one that is not only current, but able to accept changes across terms or semesters. An expired contract is one that has been terminated and cannot accept changes across terms or semesters This allows you to determine student eligibility in a multitude of ways. Figure 4.11 shows a sample contract working window.

For instance, suppose your institution has a rule that determines that when a student terminates their stay in housing, the current contract is to expire and that if the student applies again in the current term, they would need a new contract. You would define your contract behavior to handle the



Student Contracts

Life for Education Systems

N12-21-1111

Student ID

Lolabrigida, Gina L.

Name

gll1

NetID

New Graduate Contract

ACTIVE

Contract Status

2

Semesters

-

1

Semesters Used

=

1

Remaining

AUG 24 2008

Start Date

End Date

☒ Current Contract

Contract History

Status	Contract	Start Date	End Date
<input checked="" type="checkbox"/> ACTIVE	New Graduate Contract	AUG 24 2008	

Find

Edit

New

Next

Prior

OK

Cancel

Figure 4.11 -The student contract working window showing a current and active contract.

cancellation (going from assignment status “AS” to assignment status “CA”), setting the current contract to expire. But you would also need to handle the reapplication (going from assignment status “CA” to assignment status “UN”) to prompt for a new contract. Entering these behaviors is very straightforward.

The one other process effecting student contracts is the process which closes the semester. When the time in the cycle arrives where you no longer expect to be making changes in the term, the systems administrator is required to close the semester. This is a process which expires all contracts whose terms have been fully used - basically setting up the contract database for the upcoming semester or academic year. When a student attempts to reapply for housing, their active contract is checked for status (active or expired) and for remaining semesters. If their contract checks out as active with remaining semesters, they are allowed to submit their re-application.

In general, in order to be able to handle all eventualities of changes in assignment status during the course of the semester, you should have behaviors defined for every possible change in assignment status and that these behaviors follow your institution’s business rules and contract requirements.



## 5 Working with Assignments

We have already seen how the system can automatically assign populations of students by using the Auto Assign and matching functions. But there are other, less sophisticated ways in which assignments can be modified in the system. This is a brief overview of the functionality built into *CHARLES* for managing your day-to-day room assignments and modifications.

As we pointed out in chapter 4, the assignment functions are selected from the function list under Academic Year Housing/Assignment Processing area. The high level function Assignment Processing is where we handle all term assignments, moves and cancellations. Figure 5.1 shows you the main Assignments & Moves working window.

The screenshot shows the 'Assignments & Moves' window. At the top left, there's a dropdown menu set to 'Fall 2008'. Below it are several input fields: Student ID, Name, NetID, Gender, Grad Date, Student Type, School, Contract, Contract Status, and # remaining. A 'Preferences' section contains 'Hall Choices', 'Roommates', 'Lifestyle', and 'Early Decision'. A 'Matching' section has three checked checkboxes: 'Match Gender', 'Match Staff Status', and 'Match Grad/Ugrad'. The 'Assignment' section features a table with columns 'NetID', 'Roommate Name', 'BedSpace', and 'Gender'. To the right of the table are dropdowns for 'Start Date', 'Hall', and 'BedSpace', and input fields for 'Suite Type', 'Suite Capacity', 'Program', 'Use', 'Gender', 'GradUse', 'Status', 'Effective Date', and 'Assigned Date'. On the right side of the window, there is a vertical column of buttons: 'Find', 'Assign', 'Move', 'Bed for Bed', 'Print Notice', 'OK', and 'Cancel'.

Figure 5.1 -The primary Assignments & Moves working window and its associated functions.

Taking a quick look at figure 5.1, you can see that assignments are dependent upon term (in this case, the default term of Fall 2008 is automatically selected). The functions along the right side of the window indicate the actions you can undertake. You can Find an assignment, Assign a student, Move a student, process a bed-for-bed switch and print an assignment notice.

Note also that there is a Matching section on the window with a few check boxes. The default in *CHARLES* is to match by Gender, match designated Staff (RA or GA students) with pre-designated staff spaces, and to match graduate and undergraduate students with spaces pre-designated for those populations. Since these are check boxes, you have the ability to override the defaults if you need to make an assignment outside of these parameters.

## 5.1 ASSIGNING A STUDENT

You'll recall that we've mentioned the importance of the assignment status in *CHARLES*. To reiterate, the assignment status is the method by which *CHARLES* keeps track of a student as they travel through the housing cycle. The assignment status has a type of hierarchy in that while a student application passes through the cycle, the assignment status changes along the way.

A student whose information has been imported from the admissions office carries an assignment status of IN (inactive). Once the person fills out their application, the status changes to UN (unassigned). If the student cancels their housing stay, it would change to CA (cancelled). If they are put into the auto assignment function, it would change to AA (awaiting assignment) when the population to which they belong is activated. Ultimately, once a room is assigned, the status moves to AS (assigned).

There are a few more assignment statuses we will mention in due course, but the ultimate goal in assigning a student is to place the student into an available space and flip their assignment status to AS or assigned. *CHARLES*, therefore, will only allow you to assign (with the Assign function) student records that are currently unassigned (status UN).

The screenshot shows the 'Assignments & Moves' window in the CHARLES system. The window is titled 'Assignments & Moves' and has a subtitle 'Life for Education Systems'. It contains several sections for student information and assignment details.

**Student Information:**

- Student ID: A14-22-3313
- Name: Dervish, Whirling F.
- NetID: yabba
- Gender: M
- Grad Date: 0612
- Student Type: UG F
- School: UA
- Contract: Standard Contract
- Contract Status: ACTIVE
- # remaining: 8

**Preferences:**

- Hall Choices: (empty)
- Roommates: Squirrel, Rocket J.
- APARTMENT STYLE: Lifestyle
- Early Decision: (empty)

**Assignment:**

- Start Date: AUG 24 2008
- Hall: Cushing
- BedSpace: CU-0002A-1
- RoomList: CU-0002A-1, CU-0002B-1, CU-0003A-1, CU-0003A-2, CU-0003B-1, CU-0003B-2
- TR: (empty)
- Suite Type: (empty)
- Suite Capacity: 3
- Program: (empty)
- Gender: M
- Status: UN
- Effective Date: JAN 11 2009
- Assigned Date: JAN 9 2009

**Buttons:** Find, Assign, Move, Bed for Bed, Print Notice, OK, Cancel.

Figure 5.2 -Selection of a bed space shows potential roommates.

When you press the Assign button, you are prompted to enter one of the student identifiers (an id or name), which finds the person and redraws the window. Then you must select the start date of the assignment, and select a hall and bed space. Once you choose a hall, *CHARLES* uses the matching parameters in the Matching box to find any available spaces meeting those choices. When you choose a bed space, *CHARLES* will display (in the roommate list) any people who are already assigned to the suite in which the bed space belongs. Figure 5.2 shows an assignment in progress up

to the point of selecting the bed space. (Note that when *CHARLES* found the record, it shows the current assignment status as UN.) Upon completion of the assignment, *CHARLES* informs you of the success or failure of the procedure.

When a record is assigned, it creates a change in assignment status, which effects the student contract in the manner specified by your system in the assignment behaviors we discussed in the last chapter. It will also check to see whether any charges for the space need to be assessed. Typically, one would expect a charge to be generated. However, there are some instances where that might not happen. We will discuss those in the next chapter. For now, let's just concentrate on room assignments.

## 5.2 MOVING STUDENTS

Another type of assignment you can undertake is a move. Move is a function that allows you to move a student from one space to another or from one status to another. Here are a couple of scenarios.

- 1) A student receives notification of their assignment and is unhappy with the location. She requests to be moved to another floor or hall. In this case, you would look up the student using the "Find" button. Once the record appears on the window, press the "Move" button. You are immediately prompted to enter a new start date, residence hall and bed space. Therefore, the move function allows you to move a record from one space to any other available space in the system meeting the matching parameters selected on the window.
- 2) A student has completed a wait list application (assignment status WL) or has been cancelled (assignment status CA) for some reason or another. In order to assign someone from the wait list, or to reactivate a person from a cancelled status, you must use the "Move" function. Find the student's record, press the "Move" button and enter the start date, residence hall and bed space.

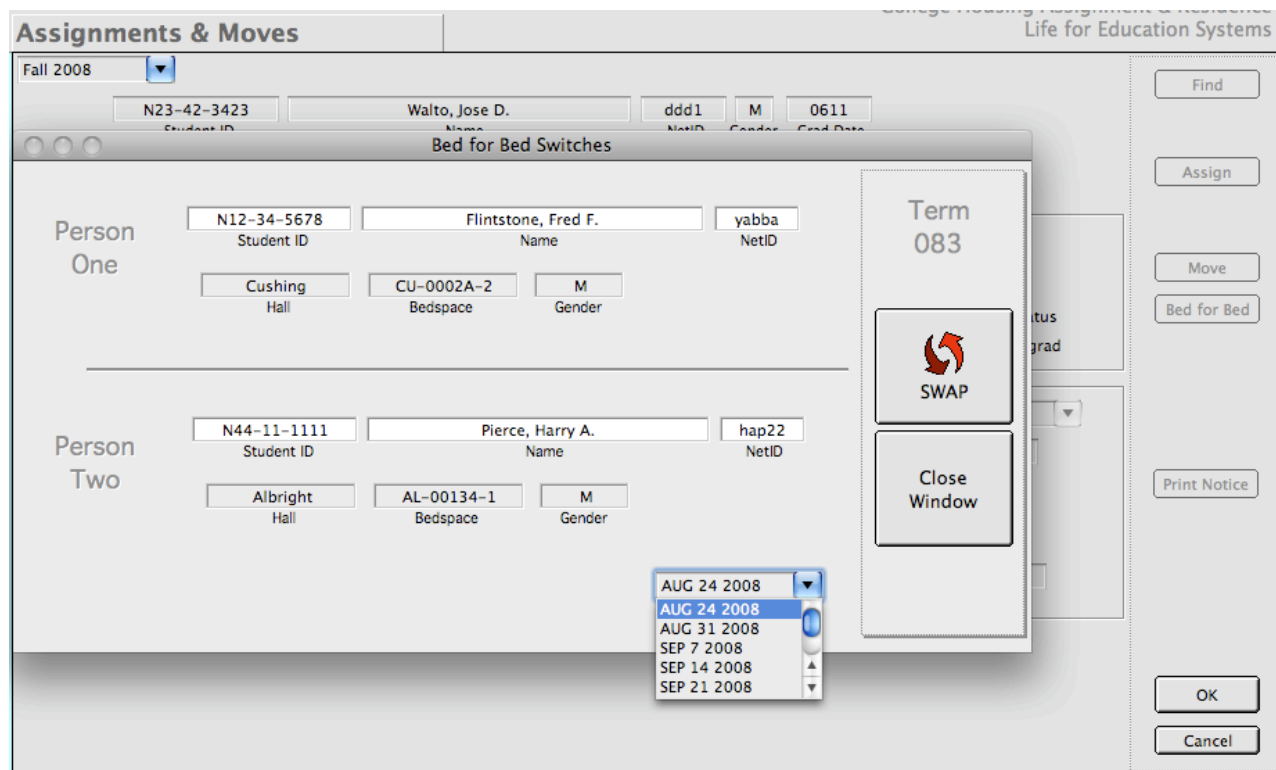


Figure 5.3 -A bed for bed switch is a way to swap room assignments between two willing individuals.

The final type of assignment you can undertake is actually two moves at once and is called “Bed for Bed”. As its name suggests, it allows you to “swap” the bed spaces of two assigned individuals. To do this, press the “Bed for Bed” button and fill out the window that appears. (figure 5.3)

### 5.3 CANCELING STUDENTS

Another function in the high level function “Assignment Processing” is called Cancellations. Cancellations are processed in their own working window because there are several types of cancellations that can be processed.

A cancellation is defined as a termination of the student’s assignment for any number of reasons. These reasons are contained in a system table and can be controlled (to a degree) by your system administrator. Students can cancel because they are going on leave, are studying abroad (or at another institution for a time), are leaving the institution or are taking a medical leave. By no means have we touched on all reasons for cancellations. These are just some examples.

Usually, we refer to a cancellation as a cancellation of an application because when you cancel an application, it results in the entire cycle of the application. A student may not be assigned at the time you cancel him/her. Cancellations end the student’s application activity for the cycle.

To cancel an application, press the “Process Cancellation” button. Enter the student’s identifier and press return. When *CHARLES* finds the record, it prompts you for a reason, and requires that you enter an effective date and the date the cancellation was received by your office. (figure 5.4)

The screenshot shows the 'Cancellations' window in the 'Life for Education Systems' application. The window is titled 'Cancellations' and has a dropdown menu for 'Fall 2008'. The form contains the following fields and buttons:

- Student ID: A14-22-3313
- Name: Dervish, Whirling F.
- Net ID: (empty)
- Student Type: UG-F
- School: UA
- Residence Hall: Cushing
- Room Number: CU-0002A
- Status: AS
- Assign Date: JAN 14 2009
- Effective Date: AUG 24 2008
- Reason for Cancellation: Personal Leave (selected from a dropdown menu)
- Term Dates: AUG 24 2008 (Term Start), DEC 19 2008 (Term End), JUL 1 2008 (Cancel Deadline)
- Weeks Begin On: Sunday
- Weeks in Term: 15
- Cancellation Effective Date: (empty)
- Cancellation Received Date: (empty)
- Override Cancellation Deadline: (checkbox)
- Buttons: Process Cancellation, Study Abroad Programs, Cancel NonPayment, OK, Cancel

Figure 5.4 -Entering a cancellation for a personal leave.

When the cancellation is processed, it results in a change in the assignment status which triggers a change in the student's current contract according to the behavior built into your system. It may also result in a charge and/or credit to the student's account record. If the student was assigned prior to the cancellation, it will free up the space so another person can be moved in.

## 5.4 HANDLING ROOM CHANGE REQUESTS

*CHARLES* is able to process room change requests that come directly from your students. You can either set up a web page to handle the incoming request, or use a paper form with some one of your staff taking care of the data entry.

From time-to-time, students become dissatisfied with their accommodations for any number of reasons. Depending on your internal regulations, you can choose whether the room change functionality exists inside the Assignment Processing higher level function, or in the Residence Life higher level function. It will work the same no matter where you decide to place it. This will allow you to allocate room change handling to the residence halls if you like.

Room change requests are entered into a system table which allows residents to choose a variety of hall types, halls, and room types, as well as explain the reason for their change request. The system categorizes these into statuses. A new room change request is given the status of "Active". Once a room offer is made, its status changes to "Offered". If the student approves the offer, it is changed to "Approved". If the student declines the room change offer, it is changed to "Declined".

The screenshot shows a web-based form titled "Room Change Request". The form is organized into several sections. The top section contains input fields for "Student ID" (N12-34-5678), "Name" (Flintstone, Fred F.), and "Net ID" (yabba). Below these are fields for "Current Hall" (Cushing), "Current Space" (CU-0004B-1), and "Gender" (Male). The next section features three dropdown menus: "Hall Type" (set to Apartment Style Accom...), "Suite Type" (set to S-Single), and "Priority" (set to 2-Very Importa...). Below these are three more dropdown menus for "First Choice" (Albright), "Second Choice" (Halsey), and "Third Choice" (Select). A checkbox for "Low Cost Option" is located to the right of the "Third Choice" dropdown. A large text area for "Reason for Request" contains the text: "Please assign me to a single. I'm having a lot of trouble with the noise associated with having multiple roommates. Thanks." At the bottom right, there is a "Submit Date" field and two buttons: "Ok" and "Cancel".

Figure 5.5 -Entering a room change request.

In figure 5.6, you'll see that requests are handled on a Hall by Hall basis. In order to see newly submitted requests, the Status drop list should be on "Active". In order to make a room change offer, simply click on the room change request in the Requests list and available spaces matching the student's preferences will be shown in the space list on the right side of the window. If you do not find any spaces with the default set of choices, you can click on those choices until you find a space to offer.

Once you click on a space in the space list, you will be able to make an offer. Just click on the Make Offer button. *CHARLES* will put a block on the space (so others can't assign anyone to it) and you are prompted to enter the date you would like to hold the space until. This, in essence, gives the student making the request a deadline upon which to get back to you. Once you make an offer, *CHARLES* will send an email to the student.

The email messages that are sent from this function are governed by another table in the system allowing you to specify the messages attached to the particular room change statuses. Since the

room change statuses are static (meaning you can't change them), we have placed the status table in the systems administrator section of *CHARLES* and changes to email messages will need to be handled by an appropriate person.

**Room Change Requests** Life for Education System

Fall 2008

Requests Current Hall: Cushing Status: Active

ID: N12-34-5678 Gender: Male Halls: Albright ☐ LowCost? Please assign me to a single. I'm having a lot of trouble with the noise associated with having multiple  
 Name: Flintstone, Fred F. Halsey Received: 02/21/2009  
 Type: APARTMENT S  
 Status Active Current Room: CU-0004B-1 Offered:

☒ Any Hall ☒ Match Hall Type  
☐ Hall Choice 1 ☒ Match Room Type  
☐ Hall Choice 2 ☒ Match Low Cost  
☐ Hall Choice 3 ☒ Match Gender

Hall	Space	Room Type	Charge Type	Gender
Albright	AL-00137-1	S	S	

☒ Notify by email ☒ NetID email address  
☐ Term email address  
☐ Alternate email address

Figure 5.6 -Room Change Requests Working Window.

Once an offer has been made, the space is blocked, and the student is emailed with a response date, all you have to do is wait until the student responds. Once he or she does, it's on to the next step, approval or denial.

Approval is very simple. Select the "Offered" status in the Status list to see offers that have been made. Then click on an offer in the Request list. The space list changes, showing only the offered space (figure 5.7). To approve, press the Approve Offer button. You'll be prompted for the effective date for the move. Enter it and the room change is complete. It's status is set to "Approved", and the student is moved to their new space. Behind the scenes, *CHARLES* handles the charge pro-rates and gender setting issues.

You'll notice on figure 5.7 that there is also a Decline Offer button. If a student responds negatively to your offer, their room change request would have to be processed as "Declined". In that instance, you would press the "Decline Offer" button. This is different than Declining the request. Declining the offer is a student initiated action. Declining the request is an administrative decision by your office. While this is a subtle difference, it is important because the email message that gets sent to the student is different. Students who decline the offer are informed with a confirmation of declination while a declined request allows you to attach a text reason for declining the request.

Room Change Requests
Life for Education Systems

Fall 2008

Requests
Current Hall: Cushing
Status: Offered

ID: N12-34-5678 Gender: Male Halls: Albright  
Name: Flintstone, Fred F. Halsey  
Type: APARTMENT S  
Status Offered  
Received: 02/21/2009  
Please assign me to a single. I'm having a lot of trouble with the noise associated with having multiple  
Current Room: CU-0004B-1 Offered: AL-00137-1

Hall	Space	Room Type	Charge Type	Gender
Albright	AL-00137-1	S	S	

☒ Notify by email
☒ NetID email address
☐ Term email address
☐ Alternate email address

Add Request
Remove Request
Decline Request
Accept Offer
Decline Offer

Figure 5.7 -Processing the response to a room change offer.

Finally, you can also remove a request. This is an administrative function that simply deletes the request from the database. No email notification is generated. It is there as as method of cleaning up inappropriate requests that students might submit.



## 6. Halls, Suites, Rooms and Bed Spaces

The title of this chapter is a summation of the different ways in which you can view your inventory of spaces. It is also the hierarchy how *CHARLES* organizes your bed spaces.

The first thing to look at here is your individual residence halls. Each hall has a name, an address and some identifiers to indicate the type of residence hall it is and how it is used. Figure 6.1 shows the Space Management working window.

The screenshot shows the 'Space Management' window with the 'Residence Halls' tab selected. The window contains several input fields for hall information: Name (Halsey), Title (Admiral Halsey Hall), Street (18 Farview Drive), City (Nontown), State (MA), and Postal Code (01006-). There is also a 'Code (room prefix)' field and a 'Codes' list box containing 'H1' and 'H2'. Below these are dropdown menus for 'Hall Style' (TRAD - WITH A/C - Traditional Hall with A/C) and 'Hall Group' (TRAD STYLE - Traditional Dorm Style). A checkbox labeled 'Use suffix in room numbers' is checked. On the right side, there are buttons for 'Find Hall', 'New Hall', 'Edit Hall', 'Delete Hall', 'Add Code', 'Next', and 'Prior'. At the bottom right are 'OK' and 'Cancel' buttons.

Figure 6.1 -The Space Management working window showing a sample residence hall.

This working window has several tabs across the top, allowing you to manage your halls and spaces. The functions on the window effect several tables in the database. The residence halls tab allows you to enter residence halls, their titles, codes and hall types. It also allows you to determine whether the rooms attached to the hall use suffixes in their identifiers. We'll see more about this when we discuss entering rooms.

Residence halls can have multiple codes. These can be used to more completely identify your room numbers and are especially useful for multi-part halls that may have room numbers that are similar for different wings or towers.

The Hall Type drop list allows you to identify the type of hall and reads from a table you can populate. This is a way to associate a hall with a type of accommodation that can be used as a preference for student applicants. It may also help in reporting in that it categorizes a number of halls as "belonging" to a specific type.



To enter a new hall, press the New Hall button and fill out the window. Once a hall has been entered, you can edit your entry, add a code to your entry or delete it. This enters a root level residence hall.

*CHARLES* also has a concept called Term Halls. As you recall, *CHARLES* is a term-based system so the hall must exist not only at the root level of the database, but also in each term. This allows you to take halls out of service for maintenance or renovation if necessary. In order to activate a hall for a term, you just begin to enter spaces into the term. Figure 6.2 shows the message you receive when first activating a hall in the Fall 2008 term. To add the hall to the term, simply press the Yes button.

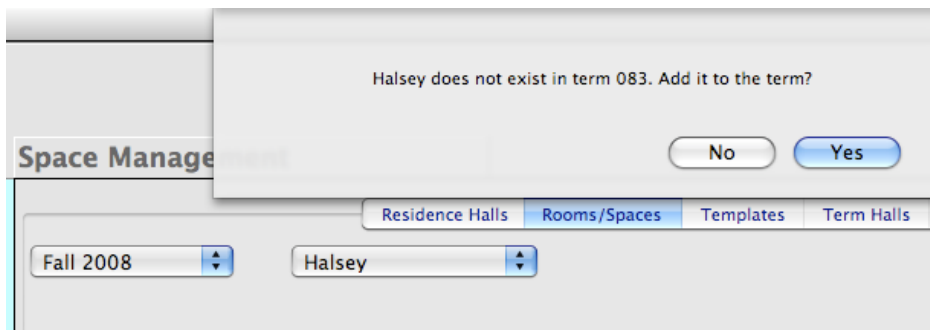


Figure 6.2 -A message and a question that requires an answer.

## 6.1 ADDING SPACES

Once the hall exists in the term, you can add spaces by pressing the “Build Spaces” button. When you begin to build spaces, the working space changes, utilizing information you entered in your hall description, specifically in reference to the hall codes and room suffix usage. Figure 6.3 shows us beginning to enter a room into our new hall named Halsey.

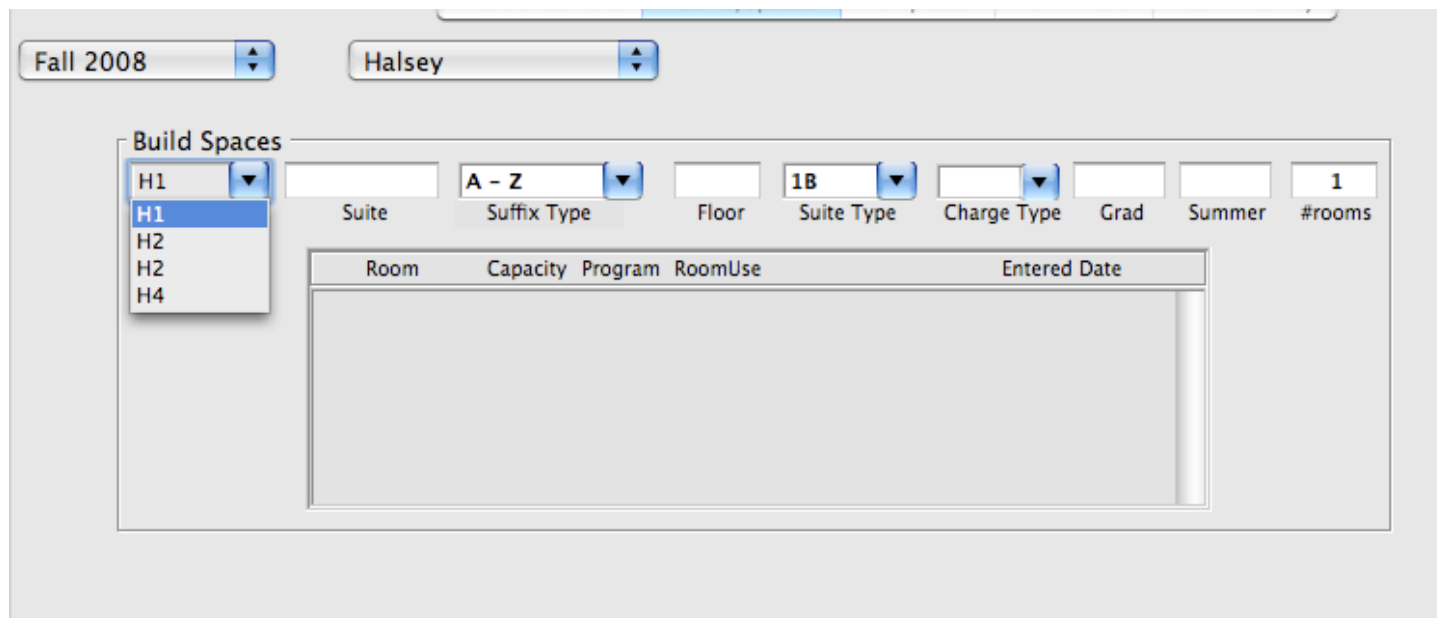


Figure 6.3 -Building spaces in the Halsey residence hall.

It is important to point out that when we add spaces to the system, we are really entering entire suites of rooms at once. The hierarchy is hall, then suite so when we enter rooms, we must be inside a suite. Also when we enter bed spaces, we should be inside the room.

Note that all of the codes we entered for Halsey appear in the prefix drop list. The prefix tags the rooms to indicate which part of the hall the room belongs to. It can also carry a different address than the hall address to allow you to group spaces in a more creative way.

There are also other drop lists that allow you to utilize system tables to fully categorize the spaces you enter. Since we indicated that we would like to use suffixes in our room numbers, a drop list appears to allow you to select the type of suffix you want to use. Suite types and charge types allow you to associate your spaces with types defined in your systems database. The Grad and Summer fields are yes/no values that indicate whether the space should be used for graduate students or for summer housing respectively. (Remember that we had matching parameters in the assignment section which used the graduate matching space indicator.)

As you enter the information for your suite, tabbing from field to field, you arrive at the #rooms data element. Tabbing through this field (it assumes at least one room) populates the room list you see below:

Room	Capacity	Program	RoomUse	Entered Date
H1-0201A	2		STU	
H1-0201B	1		STU	

Figure 6.4 -Adding a suite of rooms to Halsey.

Since we have indicated two rooms exist in this suite, we see the list populated with two rooms. *CHARLES* creates the room identifier (room number) from the prefix, suite number and suffix you specified. Since the suffix type selected here is a-z, the first room in the suite gets tagged with A, the second with B. (a third room would have gone to C, etc.)

Each room in the list must be given some positive number of spaces (capacity) and a room use. Room uses are typically STU for student use (the default) or some other indicator like STA for staff. Your institution could use any other identifiers you wish.

Figure 6.5 -Prompt to continue entry of suites.

Once you have finished, press OK or return to save your entry. You will receive a message asking whether you wish to continue entering suites. Pressing Yes will clear the template on the working window and allow you to enter another suite. No will halt the entry process.

You can continue in this fashion until you have entered all of your spaces, or you can come back later to continue.

When you have a number of spaces in your hall, finding the hall (using the drop lists on the working window) will show you all the suites in your hierarchy.

There is a lot more information you can add to your spaces. The expandable/collapsible list of suites gives you access to every room and every bed space in the room. Note in the list that for each suite we entered, *CHARLES* created a room number and a bed space for each quantity of capacity we entered.

In order to record additional information, at either the room or bed space level, simply click on the entry in the suite list.

The suite list can also be manipulated by using the arrow keys on your keyboard. Pressing enter when a high level entry is selected (one with a yellow folder) will expand or collapse the entry.

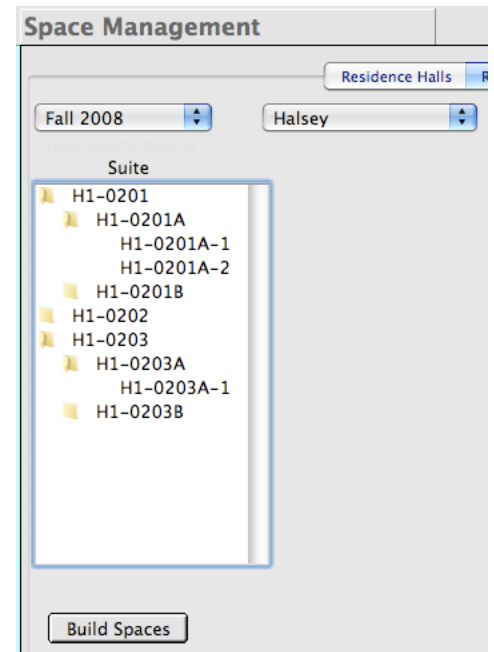


Figure 6.6 -Halsey with three suites entered..

## 6.2 COMPLETING SPACE IDENTIFICATION

Depending on which level in the hierarchy you select, you see different things in the working window. Selection of the suite name shows you all rooms. Selection of a room shows you all bed spaces in the room. Selection of the individual bed space shows you only that bed space.

When you select something in the hierarchy, you are given its contents in the window. Selection of an entire suite shows you all the rooms belonging to the suite and *CHARLES* gives you the opportunity to edit the whole suite - the "Edit Suite" button appears. Selection of a room or space gives you the opportunity to edit spaces - the "Edit Spaces" or "Edit Space" button appears.

Note also that you can always see the suite information in the area above the rooms or spaces. If you need to change the capacity of a room, do it through the "Edit Suites" function. If you want to add information to the spaces, select a room from the list or a particular bed space and use the "Edit Spaces" button.

Information that can be added to the suite itself is just the suite key number. If you edit suites, you have the opportunity to change the capacity of a room, block a room from assignment, flag a room for a particular program, change the use of a room or modify its charge type.

Information that can be added to the bed spaces are the key numbers, phone numbers, data jack numbers or the mailbox assigned to the space (if you handle mail by bed space). Figures 6.7 and 6.8 show expanded suite and room examples and illustrates how the edit buttons change.

**Space Management** Life for Education Systems

Residence Halls **Rooms/Spaces** Templates Term Halls Room Activity

Fall 2008 Halsey

**Suite**

- H1-0201
  - H1-0201A
    - H1-0201A-1
    - H1-0201A-2
  - H1-0201B
- H1-0202
- H1-0203
  - H1-0203A
    - H1-0203A-1
  - H1-0203B

**Suite Attributes**

H1-0201      2      3      2B     

Number of Rooms      Suite Capacity      Suite Type      Suite Key Number

**Room Attributes**

Room Number	Cpcty	Program	Use	Grad	Blocked?	Until	Charge Type	Entered Date
H1-0201A	2		STU	NO	<input type="checkbox"/>		2B	JAN 15 2009
H1-0201B	1		STU	NO	<input type="checkbox"/>		2B	JAN 15 2009

Build Spaces      Edit Suite      Delete Suite      Print Worksheet

OK  
Cancel

Figure 6.7 -Working window showing all rooms in the suite named H1-0201 - ready for edit.

**Space Management** Life for Education Systems

Residence Halls **Rooms/Spaces** Templates Term Halls Room Activity

Fall 2008 Halsey

**Suite**

- H1-0201
  - H1-0201A
    - H1-0201A-1
    - H1-0201A-2
  - H1-0201B
- H1-0202
- H1-0203
  - H1-0203A
    - H1-0203A-1
  - H1-0203B

**Suite Attributes**

H1-0201      2      3      2B     

Number of Rooms      Suite Capacity      Suite Type      Suite Key Number

**Space Attributes**

Space Number	Key Number	Phone Number	Data Jack	Mailbox	Use	Charge Type
H1-0201A-1					STU	2B
H1-0201A-2					STU	2B

Build Spaces      Edit Spaces      Delete Suite      Print Worksheet

OK  
Cancel

Figure 6.8 -Working window showing all bed spaces in the room H1-0201A - ready for edit.

There are a couple of points to make while looking at figures 6.7 and 6.8. Notice that in both the space attributes and the room attributes, we see the drop list to modify the charge type. When you modify the charge type in room attributes, the charge type for all spaces in the room is effected. However, it is possible to have different charge types for each space in the room. This would be handled by modifying the value in the space attributes area.

Also, when modifying a room's capacity, *CHARLES* does not ever delete a space from the system. It does, however, expire the space from the list of "active" spaces. Adding a space to a room's capacity will sequentially add an additional space to the room with a bed space indicator that is one more than the highest bed space number that currently exists. For example, if we were to add a bed space to H1-0201A, the next logical number to add is H1-0201A-3.

### 6.3 BUILDING A TERM'S SPACES FROM A PRIOR TERM

CHARLES contains a function which allows you to, in effect, roll a term's spaces into another term. This function is called "Clone Rooms" and is located on the Templates tab. Cloning rooms is a quick way to grab your list of spaces from a prior term, and roll them into your current term.

Pressing the "Clone Rooms" button gives you the opportunity to clone the entire hall compliment from a prior term, or to pick specific halls - perhaps only the ones you will be using going forward. Figure 6.9 shows you what that process looks like.

In this case, we are attempting to clone the spaces from Fall 2008 and add them to Spring 2009. We have a check list to indicate which halls to clone. Pressing OK will add the selected halls to the target term.

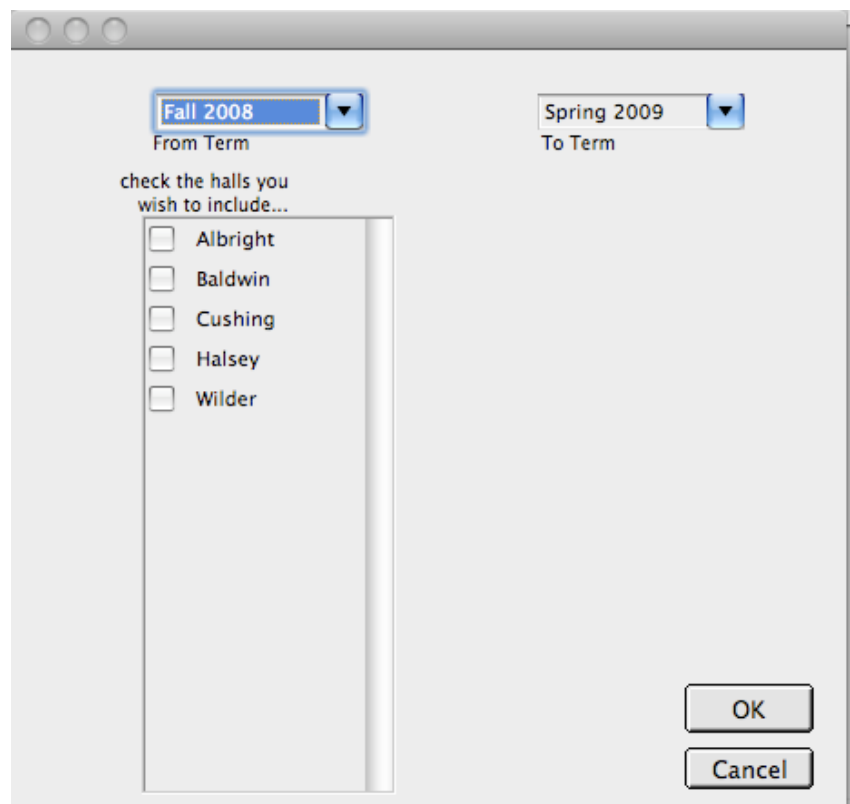


Figure 6.9 -Cloning spaces from another term.

It should be pointed out that if the residence hall already exists in the target term, you will not see it listed in the checklist. This is to keep you from overwriting any work you may have already done in setting up your spaces for the upcoming term.

*(We should also point out that you would probably not want to clone a fall term into spring if you are planning to roll fall assignments into spring. The Rollover function will handle your space requirements in that case. We used these two terms for illustration purposes only.)*

## 6.4 INDICATING LOTTERY PARAMETERS FOR ROOM RE-SELECTION

The Term Halls tab has information relating to the room re-selection for returning students - or lottery - particularly having to do with the scheduling of re-selection pick times. This only makes sense during a new fall term into which no one has been assigned. CHARLES has a more advanced re-selection process in newer versions and these data elements are no longer used. You may use them as you see fit.

We will discuss the *CHARLES* room re-selection (or lottery) in another section.

## 6.5 VIEWING A HALL'S SPACE USAGE OR ACTIVITY

The final tab on the working window is basically a query tool allowing you to select and see other pieces of information relating to your spaces that we have not yet discussed. The Room Activity tab shows you a window with a few filters to view the spaces you are most interested in. Figure 6.10 shows you the window for our Fall 2008 Cushing Residence.

The screenshot shows the 'Space Management' window with the 'Room Activity' tab selected. The window title is 'Life for Education Systems'. The interface includes a tab bar with 'Residence Halls', 'Rooms/Spaces', 'Templates', 'Term Halls', and 'Room Activity'. Below the tabs, there are two dropdown menus: 'Fall 2008' and 'Cushing'. A search bar labeled 'Find all the suites with' has 'no value' entered. A table displays room data with columns: Room ID, Suite, Charge Type, Gender, Program, Grad Use, and an empty column. The table contains 10 rows of data. To the right of the table, there are two more dropdown menus: 'Grad Use' set to 'either' and 'Show' set to 'all spaces'. At the bottom right, there is a 'Print List' button with a printer icon. A 'Total: 10' label is located at the bottom of the table area. At the very bottom of the window are 'OK' and 'Cancel' buttons.

Room ID	Suite	Charge Type	Gender	Program	Grad Use	
CU-0002A-1	CU-0002	TR	M		NO	
CU-0002A-2	CU-0002	TR	M		NO	
CU-0002B-1	CU-0002	TR	M		NO	
CU-0003A-1	CU-0003	1B			NO	
CU-0003A-2	CU-0003	1B			NO	
CU-0003B-1	CU-0003	1B			NO	
CU-0003B-2	CU-0003	1B			NO	
CU-0004A-1	CU-0004	2B			YES	
CU-0004A-2	CU-0004	2B			YES	
CU-0004B-1	CU-0004	2B			YES	

Figure 6.10 -Viewing room activity for Cushing Residence Hall.

This part of the space management interface allows you to quickly find out some interesting information about your hall. For instance, a quick look at figure 6.10 shows us that there are 10 bed spaces in Cushing for Fall 2008. Some suites have been assigned since there are some gender fields with values.

We have a select list at the top of the space list where we can select some values to filter out what displays in the space list. We can also filter our spaces by grad use or by showing all spaces, assigned spaces or empty spaces.

There is also the ability to print the list to the screen or to a printer.

A quick note about gender assignments in *CHARLES*: Gender assignments are set in the term room table. They all start out as blank. As room assignments are made, the first person assigned to a suite (for the term) flags the gender assignment for every bed space in the suite. Gender assignment flags in bed spaces are never cleared unless every person assigned to the suite is moved to another space or cancelled.

Figure 6.11 shows the same tab with some filters applied.

Figure 6.11 -Viewing empty bed spaces in Cushing with a capacity of 2.

The screenshot shows the 'Space Management' window with the 'Room Activity' tab selected. The interface includes filters for 'Fall 2008' and 'Cushing'. A search filter is set to 'Find all the suites with number of bedspaces equal to 2'. The table displays four rows of data for room IDs CU-0003A-1, CU-0003A-2, CU-0003B-1, and CU-0003B-2, all with a charge type of 1B and no gender or program assignments. The 'Grad Use' column shows 'NO' for all entries. A 'Total: 4' is displayed at the bottom right of the table. On the right side, there are dropdowns for 'Grad Use: no' and 'Show: empty spaces'. A 'Print List' button with a printer icon is located at the bottom right, along with 'OK' and 'Cancel' buttons.

Room ID	Suite	Charge Type	Gender	Program	Grad Use
CU-0003A-1	CU-0003	1B			NO
CU-0003A-2	CU-0003	1B			NO
CU-0003B-1	CU-0003	1B			NO
CU-0003B-2	CU-0003	1B			NO

Total: 4

Grad Use: no

Show: empty spaces

Print List

OK

Cancel

## 7. Viewing Student Records: Correspondence & Inquiry

For the casual desk worker, office clerk or assistant, telephone technician or any other number of staff members requiring quick access to student records and tracking, *CHARLES* becomes a very friendly and convenient source of information.

### 7.1 STUDENT INQUIRY

One of the most important parts of the entire system for information retrieval is the Student Inquiry function. This is a working window including nearly every piece of information relating to a student's stay in your institution, giving your staff instant access to see and explain what is happening to an individual throughout their entire stay in housing. Figure 7.1 shows a sample inquiry working window.

The screenshot displays the 'Student Inquiry' window from 'Life for Education Systems'. The window is divided into several sections for data entry and viewing.

**Top Section:** Contains a dropdown for 'Fall 2008' and input fields for 'Student ID' (N44-11-1111), 'Name' (Pierce, Harry A.), 'Gender' (Male), 'Birthdate' (JAN 5 1986), 'Grad Date' (0611), and 'NetID' (hap22). Buttons for 'Find Student' and 'Find Suite' are on the right.

**Addresses Section:** Includes a 'Cell Phone' field (999) 122-3333 and a table of addresses.

Type	Street	City	State	Zip	Country	Phone
P	123 Main Street Apartment 77	London	XX	W3-12-347P	US	(999) 122-3333
C	123 Main Street Apartment 77	London	XX	W3-12-34	UD	(999) 122-3333

**Current Status Section:** Features tabs for 'Current Status', 'History', 'Charges/Payments', 'Correspondence', 'Room History', and 'Miscellaneous'. Below the tabs are input fields for 'Student Type' (UG-T), 'School' (UG), 'College' (UA), 'Year' (2), 'Group/Rate' (1), and 'Confirmation Number'.

**Emergency Contact Section:** Includes fields for 'Person to Contact' (Pierce, Ma), 'Relation' (MO), 'Phone Number' (999) 122-3333, and 'Alternate Phone' (999) 122-3333.

**Contract Section:** Includes fields for 'Contract Type' (Transfer Contract), 'Contract Status' (ACTIVE), 'Contract Date' (AUG 24 2008), 'Semesters' (4), 'Used' (2), and 'Remaining' (2).

**Assignment Section:** Includes fields for 'Cushing' (CU-0002A-2), 'Bedspace' (TR), 'Suite Type' (TR), 'Phone Number', 'Assignment Status' (AS), 'Assignment Date' (JAN 14 2009), and 'Effective Date' (AUG 24 2008).

**Roommates Section:** A table showing roommates for 'Cushing - suite: CU-0002'.

Roommate Name	Roommate ID	Roommate Suite	Roommate Effective Date
Dervish, Whirling F.	A14-22-3313	CU-0002A-1	AUG 24 2008
Pierce, Harry A.	N44-11-1111	CU-0002A-2	AUG 24 2008

Buttons for 'OK' and 'Cancel' are at the bottom right.

Figure 7.1 -A sample student inquiry working window showing a fictitious student's record for Fall 2008.

If you have desk staff or student workers tasked to answer phone inquiries, the Inquiry functions should be their main source of information. Even though there are only two functions on the working window, "Find Student" and "Find Suite", these two simple functions allow you to retrieve enough information about a student to answer nearly every question they (or their parents) may have about their application and assignment.

The sections of the inquiry window are clearly marked to indicate where the information you need is located. Across the top of the window, we can see certain demographic information. Below that, any addresses the student or applicant has in a scrollable list. Below that, we have a tab area with several choices, the default being the "Current Status" tab.



In order to find a student's information, press the "Find Student" button and enter one of the student's identifiers (an ID or their name), then press return.

As with many functions in CHARLES, this one is term specific and will choose your current default term when it is selected. The term list in the upper left corner of the working window is always active so you can scroll up and down through it to find information pertaining to other terms.

The addresses section gives you quick access to all of a student's addresses. There are three buttons in this section, giving you additional functions to allow you to add an address, edit the selected address or delete the selected address. It should be noted that a student's permanent address cannot be deleted (although it can be modified). Figure 7.2 shows an address insert in progress.

The screenshot shows the 'Student Inquiry' window for a student named Dervish, Whirling F. The 'Addresses' section is active, and a dialog box is open for adding a new address. The dialog box contains the following fields:

- Type: M
- Street 1: P.O. Box 332
- Street 2: Friendly Station
- City: Sommerville
- State: MA
- Postcode: 11200
- Country: US United States
- Phone: 413

The dialog box has 'OK' and 'Cancel' buttons. The background window shows the student's information, including their birthdate (OCT 15 1989), gender (Male), and emergency contact (Dervish, Mother).

Figure 7.2 -Adding a mailing address to a student on the Student Inquiry working window.

In the Current Status tab, we can see the type of student we have selected, where they are assigned, who their roommates are, what school they are in, who their emergency contact is, their current room assignment, when it was assigned and their current contract status. Notice that the student we've looked up also appears in the roommates list. If we double-click on the roommate in the roommates list, that person's information is pulled from the database and the window is redrawn. This way, we can quickly see all the information about all the people assigned to that suite.

The History Tab shows any database activity that has occurred during the course of activity for a student record. CHARLES stores all inserts and edits to its database records and tags them with who made the change and when. This tab shows you a scrollable list of those changes. (figure 7.3) To see more information about an entry there, double-click that entry's line in the list.

The screenshot shows the 'Student Inquiry' window with the 'History' tab selected. A list of database activities is displayed, including inserts and updates. A dialog box is open showing the details of the selected entry (Sequence 3775).

Sequence	User	NetID	Time
3773	harry		07:13
3774	harry		07:13
3775	harry		07:13
3776	harry		07:13
3777	harry		07:13
3778	harry		07:13
3779	harry		07:13
3825	Admin		17:03

The dialog box shows the following details for the selected entry (Sequence 3775):

- INSERT
- sa\_type=M
- sa\_street1=P.O. Box 332
- sa\_street2=Friendly Station
- sa\_city=Sommerville
- sa\_state=MA
- sa\_postcode=11200
- sa\_sid=A14223313
- sa\_country=US
- sa\_phone=

Figure 7.3 -Expanding a line of database history on the History Tab to see what data was inserted.

The Charges/Payments tab show you exactly that. It is a history list of all items charged and payments (insofar as you track them) made for the student during the course of the term. Figure 7.4 shows the charge history for our fictitious student.

There are two functions embedded in this tab. One is to allow for the reassessment of charges for the term selected. Another is to allow you to enter new payments. Care should be exercised in assigning these tasks to individuals. As with other functions, these are controlled according to the user's group access level.

**Student Inquiry** Life for Education Systems

Fall 2008 A14-22-3313 Student ID Dervish, Whirlring F. Name Male Gender OCT 15 1989 Birthdate 0612 Grad Date wfd1 NetID

Cell Phone: (413) 555-3224

**Addresses**

Type	Street	City	State	Zip	Country	Phone
P	523 Park Place Apartment #34	Spin City	MA	01100-2323	US	(413) 555-3223
M	P.O. Box 332 Friendly Station	Spin City	MA	11200-	US	

Current Status History **Charges/Payments** Correspondence Room History Miscellaneous

**Charges** [ChargeGeneration](#)

Code	Description	Account	Effective	Processed On	End Date	Amount
Est-UGF	Freshman Estimate	664-111	D AUG 24 2008	JAN 9 2009	DEC 19 2008	4000.00
Est-UGF	Freshman Estimate	664-111	C AUG 24 2008	JAN 14 2009	DEC 19 2008	-4000.00
Cus-TR	Cushing Triple Charge	6643	D AUG 24 2008	JAN 16 2009	DEC 19 2008	3895.00

**Payments** [Enter Payment](#)

Code	Description	Paid On	Processed On	Processed By	Amount
083-RSV1	Housing Reservation Payment	MAY 19 2008	JAN 16 2009	Admin	1000.00

OK Cancel

Figure 7.4 -The Charges/Payments tab shows the history of account activity for each term.

The Correspondence tab (figure 7.5) is where you might see any activity that one of your co-workers thought should be mentioned in the student record. It is a way to log interactions between your office and the student (or any other individual for that matter) pertaining to your student's file. This information becomes immediately accessible for anyone dealing with a student inquiry or wishing to see any interactions to date in a student record. *CHARLES* calls these interactions "Notes".

It is also where you might like to add a new record to the student's correspondence file. To do this, simply press the "Add New Note" A window appears allowing you to make the entry. Press OK when finished.

Since the list of student notes is not large enough to see some lengthy correspondences, you may double-click the record line in the list to expand the message. If your user group has the privilege, you may also edit or delete the note using the expanded view. (figure 7.6)

**Student Inquiry** Life for Education Systems

Fall 2008 A14-22-3313 Student ID
Dervish, Whirling F. Name
Male Gender
OCT 15 1989 Birthdate
0612 Grad Date
wfd1 NetID
Find Student

Cell Phone: (413) 555-3224
 Find Suite

**Addresses**

Type	Street	City	State	Zip	Country	Phone
P	523 Park Place Apartment #34	Spin City	MA	01100-2323	US	(413) 555-3223
M	P.O. Box 332 Friendly Station	Spin City	MA	11200-	US	

Current Status | History | Charges/Payments | **Correspondence** | Room History | Miscellaneous

Date	UserID	Text
JAN 16 2009	Admin	Student's mother called on August 1 to see whether Whirling could bring his cat to school. Was told that the ...

Add New Note (double-click line to see full message)
OK
Cancel

Figure 7.5 -The Correspondence tab shows the notes attached to the student's record.

Another thing to be aware of is that student notes are not term specific. These notes will appear no matter which term is selected in the term list. Therefore, you should pay attention to the date of the note when looking at these records if you need to determine their context.

Similarly, the Room History tab is not term specific in that it shows all assignments that your student has had since they have been at your institution. Here you can see all of the rooms the student has been placed into, by term and ordered by date. (figure 7.7)

As you can see in the example in figure 7.7, this student has been assigned several times and across several terms. The list shows the term, the bed space, assignment status, effective dates of the assignments, check in and check out information and any comments that the system has placed in the assignment record

You should note that *CHARLES* does not delete assignments as it moves people from space to space. It does, however, mark the assignment as

**Correspondence**

Admin User Login JAN 16 2009 Date

Student's mother called on August 1 to see whether Whirling could bring his cat to school. Was told that the office was allergic to cats and that it would be okay as long as the cat was kept out of the office.

She was also told there would be an additional charge and that we did not have a meal plan for cats.

Mother said thanks.

Edit Note Print Note
Delete Note

Figure 7.6 -Double-clicking the list line displays the entire note.

inactive, thereby preserving the history of activity undertaken on the student's behalf.

Student Inquiry

Life for Education Systems

Fall 2008

N44-11-1111

Pierce, Harry A.

Male

JAN 5 1986

0611

hap22

Student ID

Name

Gender

Birthdate

Grad Date

NetID

Addresses

Cell Phone: (999) 122-3333

Type	Street	City	State	Zip	Country	Phone
P	123 Main Street Apartment 77	London	XX	W3-12-347P	US	(999) 122-3333
C	123 Main Street Apartment 77	London	XX	W3-12-34	UD	(999) 122-3333

Current Status

History

Charges/Payments

Correspondence

Room History

Miscellaneous

Term	Status	BedSpace	Effective	Checked In	Checked Out	Description
083	AS	CU-0002A-2	AUG 24 2008	AUG 28 2008		
083	AS*	AL-00133-2	AUG 24 2008			
083	AS*	CU-0002A-2	AUG 24 2008			
083	AS*	AL-00134-1	AUG 24 2008			
082	AS					Room swap with N12345678 - Flintstone, Fred F.
081	AS	AL-00133-2	JAN 6 2008	JAN 9 2008	MAY 5 2008	Reactivate from cancelled on JUN 5 2008.

Note: An asterisk (\*) in the Status field indicates a currently inactive assignment.

OK

Cancel

Figure 7.7 -The Room History tab displays every assignment the student has had during their stay in housing.

The final tab on the Student Inquiry working window is to show any information pertaining to student records that may not fit anywhere else. This is where you can find additional data elements from the student's term or demographic record, comments or other data you'd like to have at your fingertips.

CHARLES can be configured to show any other data you'd like to place here for your convenience.

If your institution would like to add additional information placed in this tab, please contact customer support.

There is another method in which you can communicate with your students. It is through the eMail Generation function.

Student Inquiry

Life for Education Systems

Fall 2008

N44-11-1111

Pierce, Harry A.

Male

JAN 5 1986

0611

hap22

Student ID

Name

Gender

Birthdate

Grad Date

NetID

Addresses

Cell Phone: (999) 122-3333

Type	Street	City	State	Zip	Country	Phone
P	123 Main Street Apartment 77	London	XX	W3-12-347P	US	(999) 122-3333
C	123 Main Street Apartment 77	London	XX	W3-12-34	UD	(999) 122-3333

Current Status

History

Charges/Payments

Correspondence

Room History

Miscellaneous

Student SSN

Comments

Student speaks with a foreign accent. May be difficult to understand on the phone.

OK

Cancel

Figure 7.8 -The Miscellaneous tab - for access to other needed information.

## 7.2 eMAIL GENERATION

eMail Generation allows you to define populations and correspondences to send to your applicants, informing them of deadlines and other important information during their journey through the housing cycle.

The screenshot shows the 'eMail Generation' window with the 'Population' tab selected. The window has a title bar with 'eMail Generation' and 'Life for Education Systems'. Below the title bar is a dropdown menu set to 'Fall 2008'. The main area is divided into three columns: 'Assignment Status', 'Student Types', and 'Buildings'. The 'Assignment Status' column has a list of checkboxes: 'Assigned' (checked), 'Cancelled', 'Inactive', 'Medical Leave Cancellation', 'Study Abroad Cancellation', 'Unassigned', and 'Wait List'. The 'Student Types' column has a list of checkboxes: 'Graduate Student', 'Undergraduate Student', 'School Codes' (with sub-items: 'Grad Arts & Sciences', 'Undergrad Arts & Sciences', 'Undergrad Business', 'University without Walls'), and 'Housing Types' (with sub-items: 'Freshman' (checked), 'New Grad Student', 'Returning Undergraduate', 'Transfer Student'). The 'Buildings' column has a list of checkboxes: 'Cushing-CU', 'Baldwin-BA', 'Albright-AL', 'Wilder-WI', 'Halsey-H1' (checked), 'Halsey-H2', 'Halsey-H2', and 'Halsey-H4'. To the right of these columns are radio buttons for 'Send to': 'NetID address' (selected), 'Term eMail Address', and 'Alternate eMail Address'. Below these are radio buttons for 'Gender': 'Either' (selected), 'Female', and 'Male'. At the bottom left is a checkbox for 'Override SQL'. To its right is a text box containing an SQL query: `(select s_id,s_netid,s_firstname,s_lastname,s_name,s_gender from f_studentdemo,f_assignment,f_termstudent where (asnstatus='AS') and (term_htype='F') and (asnroom like 'H1%') and asnterm='083' and asnsid=s_id and asnactive=1 and term_term='083' and term_id=s_id)`. At the bottom right are two buttons: 'Save Population' and 'View Population'.

Figure 7.9 -The eMail Generation function has a working window with three tabs.

Your messages and populations are defined by term. The first step in generating an email is to decide who to send it to. The Population tab is where you define the group of students to send your email to. You may choose to send it to student NetID addresses, their term eMail Addresses or the alternate eMail addresses the students have given us on their applications.

Defining your population is fairly easy. Simply select the categories of data elements from the lists to put together your definition. As you select items from the various check lists, an SQL query is generated in the lower text box. In the sample in figure 7.9, we've selected assigned freshmen in Halsey, wing H1.

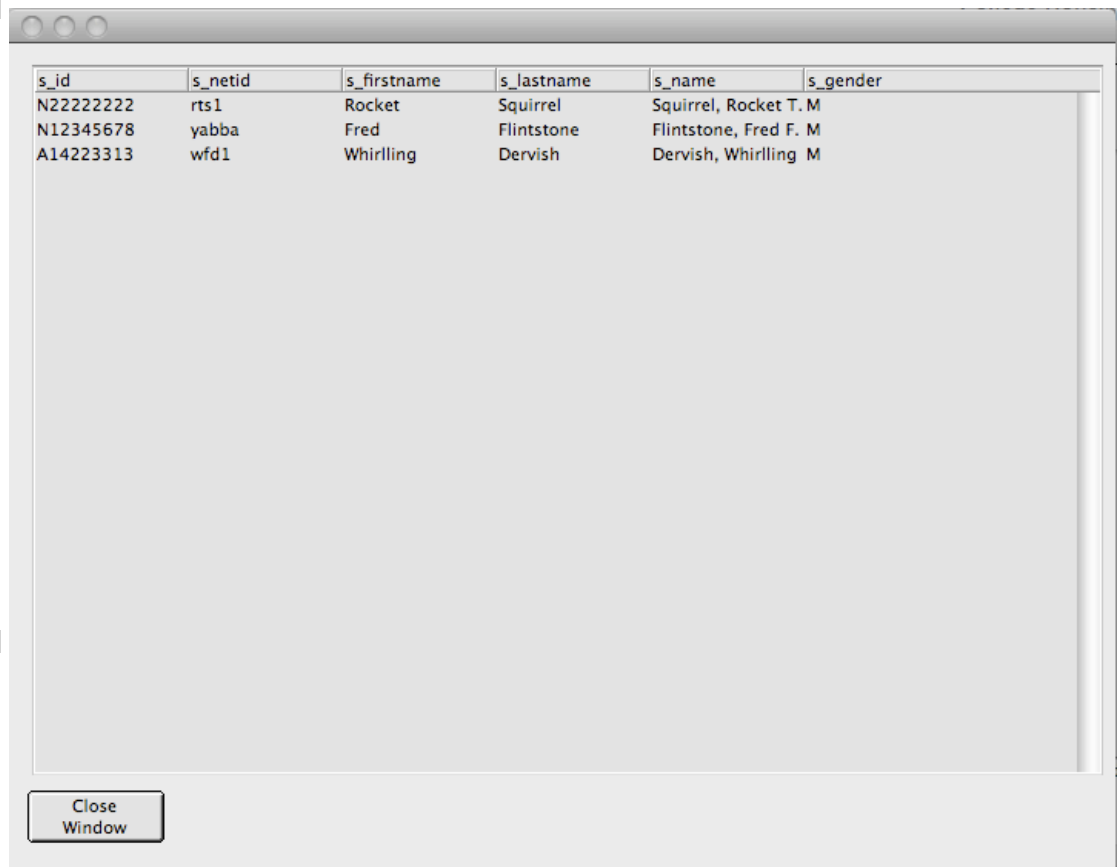
Notice that to the left of the sql text box, there is an override sql check box. If you check this, you will be able to formulate your own sql query (if you know about such things) upon which you may define populations that may fall outside of the list of selected options.

Also note that once your population has an sql definition, you will see the Save Population and View Population buttons. You are allowed to save a population definition so you can use it again. Viewing the population is so you can be sure the population you've devised is fetching the right people. Figure 7.10 is the window that appears when you view your population.

Once you are satisfied that the proper group of people is being fetched from your population description, you are ready to compose your email message.

Click on the Compose tab. Figure 7.11 shows you a sample composition.

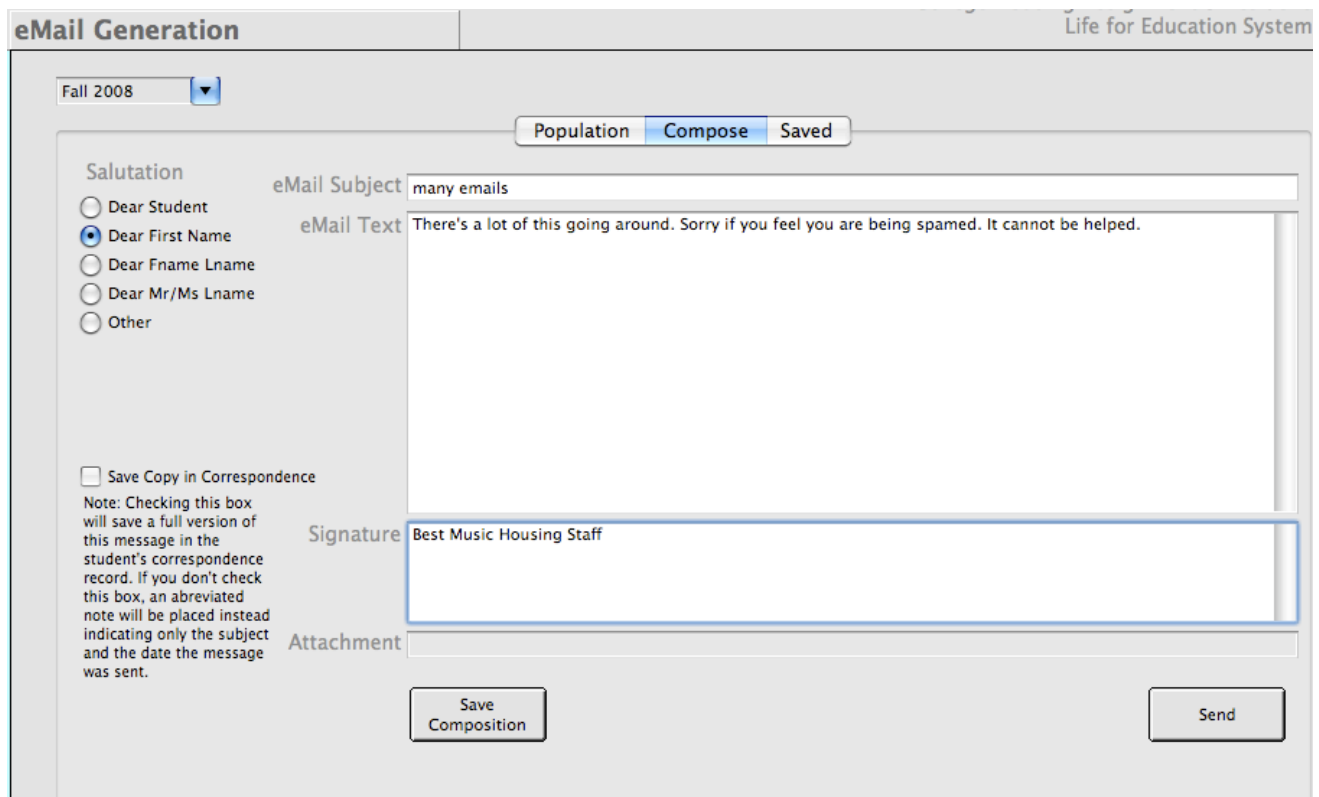
As you can see, on the compose tab, you are able to enter an email subject, text and signature. You are also able to select how the email is addressed to your population by using the salutation.



s_id	s_netid	s_firstname	s_lastname	s_name	s_gender
N22222222	rts1	Rocket	Squirrel	Squirrel, Rocket T. M	
N12345678	yabba	Fred	Flintstone	Flintstone, Fred F. M	
A14223313	wfd1	Whirlling	Dervish	Dervish, Whirlling M	

Close Window

Figure 7.10 -Checking your population to see if it is collecting the expected result.



**eMail Generation** Life for Education System

Fall 2008

Population **Compose** Saved

Salutation

☐ Dear Student

☒ Dear First Name

☐ Dear Fname Lname

☐ Dear Mr/Ms Lname

☐ Other

☐ Save Copy in Correspondence

Note: Checking this box will save a full version of this message in the student's correspondence record. If you don't check this box, an abbreviated note will be placed instead indicating only the subject and the date the message was sent.

eMail Subject many emails

eMail Text There's a lot of this going around. Sorry if you feel you are being spamed. It cannot be helped.

Signature Best Music Housing Staff

Attachment

Save Composition Send

Figure 7.11 -Composing an email to send or to save.



The Compose tab is also where you send your your email when you are ready. You also have the opportunity to save your email composition, which can be used again later, either as a complete composition or as a template that you can use to edit with.

Note in figure 7.11 the check box to Save copy in Correspondence. Recall the correspondence tab on the inquiry where we can see notes attached to student records. This check box allows us to save an entire copy of the email we generate into the notes in the Correspondence tab. If we do not check this box, CHARLES will assume you don't want to save the entire email message. It will, instead, place a note in the Correspondence tab anyway, more simply stating that an email correspondence was sent, when it was sent and the subject of that email.

The final tab in the eMail Generation working window (figure 7.12) is the one in which you get to select and reuse your saved populations and saved email compositions.

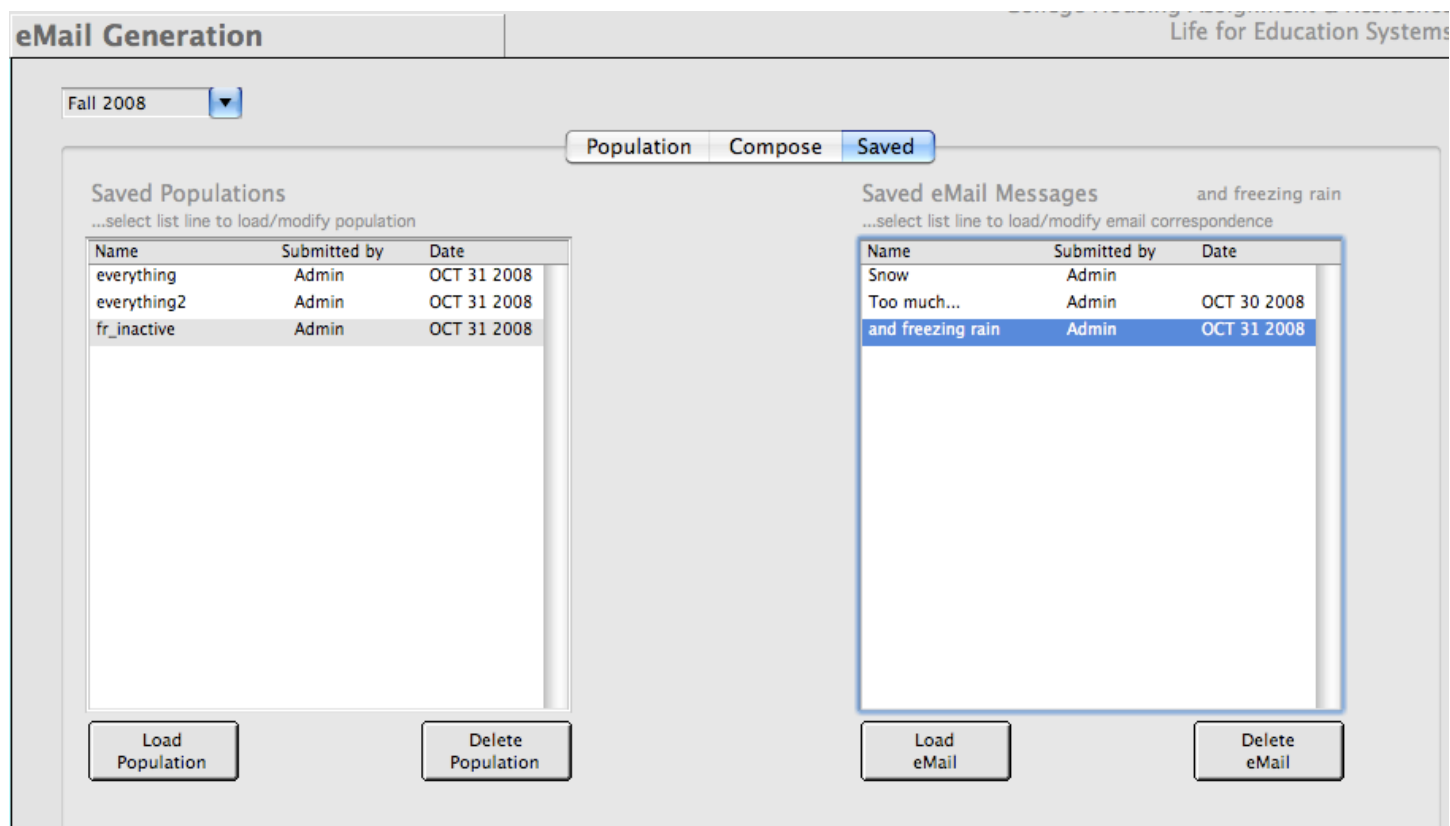


Figure 7.12 -Selecting from saved populations and email messages.

When you select from the list of saved populations, two buttons appear: “Load” and “Delete” Population. Loading will select your population. If you want to use it as it is, that is fine. If you would like to edit a loaded population, simply select the Population tab and your population will be there awaiting your modifications.

Similarly, selecting a saved email message causes the “Load” and “Delete” eMail buttons to become visible. Loading the selected eMail puts its name in the upper right just above the list of messages so you can tell it is loaded.

When you have loaded both a population and an email message, you will be presented with a “Send” button. (figure 7.13) Pressing this will compile your population (based on your description) and send the selected email message to each person.

Population   Compose   **Saved**

**Saved Populations**   fr\_inactive

...select list line to load/modify population

Name	Submitted by	Date
everything	Admin	OCT 31 2008
everything2	Admin	OCT 31 2008
fr_inactive	Admin	OCT 31 2008

Send

Load Population

Delete Population

**Saved eMail Messages**   and freezing rain

...select list line to load/modify email correspondence

Name	Submitted by	Date
Snow	Admin	
Too much...	Admin	OCT 30 2008
and freezing rain	Admin	OCT 31 2008

Load eMail

Delete eMail

Figure 7.13 -Saved tab with both a loaded population and a loaded email message, ready to send.

You may ask how *CHARLES* is able to do this? The answer is, basically, that there are some data elements in your institutions set-up table which allow your systems administrator to store your institution's SMTP server and a user name with password for authentication. So it is actually that you inform *CHARLES* how to send these messages when you enter these parameters.

Finally, one last picture (figure 7.14) to show what happens when you elect to save a copy of your entire email to a student's notes. Any notes sent by the system will be prefixed with the SYSEMAIL tag.

**Student Inquiry**   Life for Education Systems

Fall 2008   N12-34-5678   Flintstone, Fred F.   Male   JAN 5 1965   0112   yabba

Student ID   Name   Gender   Birthdate   Grad Date   NetID

Cell Phone: (212) 333-3333

**Addresses**

Type	Street	City	State	Zip	Country	Phone
P	123 Dino Way	Bedrock	NY	11000-	US	(212) 333-3333
C	123 Dino Way	Bedrock	NY	11000-	US	(212) 333-3333

Current Status   History   Charges/Payments   **Correspondence**   Room History   Miscellaneous

Date	UserID	Text
OCT 31 2008	Admin	SYSEMAIL To: yabba@bestmusic.edu Subject: Finally almost ready Dear Mr. Flintstone, This new system is al...
JUL 23 2008	Admin	dabba do

(double-click line to see full message)

Add New Note

OK

Cancel

Figure 7.14 -Sample of note generated from the eMail Generation function..



## 8. Other Convenient Functions

Since we've dealt primarily with the Academic Year functions thus far, this may be a good point to introduce some of the other convenient functions (OCFs) that are built into *CHARLES*.

Since these OCFs are not easily categorized into some other heading, we will deal with them in their own sub sections. Let's begin...

### 8.1 HANDLING GROUPS

*CHARLES* has a function called Group Entry, which allows you to define groups. These can be internal to your institution or external. They could be fraternities or sororities. They could be members of the basketball team, football team, golf aficionados, or people who can't roll their tongue. Basically, you can have groups for any reason that makes sense on your campus. The group entry function is where you define these and assign specific attributes to them.

Figure 8.1 shows you the basic group entry function's working window.

**Group Entry** Life for Education Systems

Fall 2008 CU Cushing Scholars  
Group Code Group Name

Contact myself and I  
Street Address 123 Main  
City, State, Postcode Anytown NY 11000  
Phone ( ) 8-8888  
Group Email me@myselfandI.com  
Entered on: April 11, 2008

☐ Institution affiliated

Term Group Data Correspondence

1 Rate AUG 24 2008 Arrival Date DEC 22 2008 Departure Date

**Group Members**

Name	ID	Gender	Assigned Room	Assignment Status
Dervish, Whirling F.	A14-22-3313	M	CU-0002A-1	AS

Print

Find Insert Edit Next Prior Delete Add Member OK Cancel

Figure 8.1 -Group Entry working window showing a group with one member.

Each group has a term associated with it, a code, name, contact and address. It also has a rate and arrival and departure date. Groups also have their own affiliated notes which can be found in the Correspondence tab.

The functions along the right side of the working window behave as you'd expect from your familiarity with the system so far.

The convenience features of this function are built into its grouping functionality, its ability to group people according to shared attributes or similarities. Another even greater convenience, though, is its ability to charge all of the people in the group a set rate.

If, for example, your institution (not that anyone would do this) gives a 50% discount for housing to all varsity football players. To handle that, all you have to do is choose a rate from among the 15 possible rates per charge type, calculate it to be half off, and create a group that charges the rate you chose. As you enter the varsity football members, their rates will automatically pick up the new (and in this case, cheaper) rate.

You could also add members from the already entered population. If you add someone who is already entered and assigned in the system, *CHARLES* will recalculate their charges to match the group according to the arrival and departure dates that have been entered and the date you wish to initiate the addition.

Likewise, you can remove members from the group with the same (albeit opposite) effect, crediting the group charge and calculating the student's charge based on the rate definitions set up by your system administrator. To do this, double-click the list line in the Group Members list corresponding to the person you wish to remove.

Remember that the group and rate are displayed on the Student Inquiry working window so you can easily see when a person is affiliated with a group.

## 8.2 TROUBLE TICKETS

*CHARLES* has a built-in method of communication that allows you to, in effect, speak to your system administrator, informing them of errors in processing or any other message you would like to pass on for explanation. This method is called Trouble Tickets.

Trouble Tickets should appear in your list of available functions. Selecting it presents you with a working window in which you can compose your message and deliver it to your system administrator. (figure 8.2)

The upper list in the Trouble Tickets working window shows a list of trouble tickets you have submitted. Clicking on a ticket fills in the lower list containing any correspondences attached to the ticket. Note that on the upper list, there are three tabs across the bottom left of the list.

The way the Trouble Ticket system works is that you submit a ticket by using the "New Ticket" button. This opens another window (figure 8.3) and allows you to compose your correspondence, attach the ticket to a student (if the issue pertains to one), give the ticket some priority and finally submit the ticket for review. The status of the ticket at the time you submit it is "new".

Your system administrator will see your ticket and respond to it. When the system administrator responds, they are able to attach correspondence for clarification or explanation and reset the ticket's status. The statuses for tickets are "new", "pending" or "fixed". (figure 8.4).

Let's look at figures 8.2 - 8.4 to see how this might work. Our user, Harry, believes the charges generated for the assignment of a resident are incorrect. He has asked the system administrator to check the tables and make sure the system is behaving appropriately.



Trouble Tickets Admin

Life for Education Systems

submitter	title	type	term	status	urgency	student id	student's name	checked out
harry	Incorrect fall charges?	charges/credits	083	0	0	A14223313	Dervish, Whirling F.	Admin

New Pending Fixed

Status: New Set Check Out Check In Refresh

submitter	type	date	
harry	0		<p>I'm not sure but I think this student was charged incorrectly. Could you check? I may have entered an incorrect date. His assignment is supposed to be effective on 8/24.</p> <p>Thanks</p>

New Ticket
Delete Ticket
Attach Correspondence
Print Just This Ticket

Figure 8.4 -Harry's ticket appears in the System Administrator's working window.

Once Harry responds to the administrator's request, the administrator can close out the ticket by changing the status of the ticket to fixed.

Anytime you, as a user or as an administrator would like to see the list of

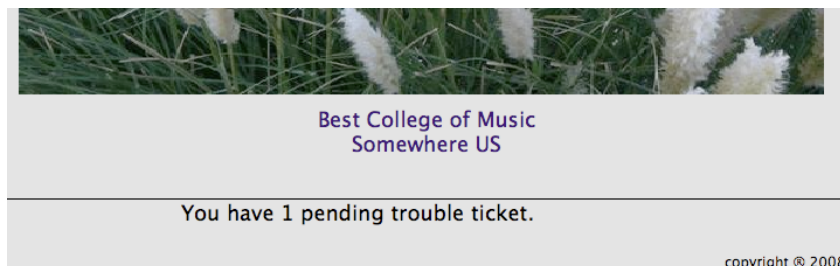


Figure 8.5 -CHARLES tells you when you need to respond to a ticket.

tickets submitted, simply press the appropriate tabs on the upper list of your trouble ticket working window. Also note that if you have a pending ticket, the pending status message will persistently appear on your window until it is resolved.

A couple of other things to point out on this series of windows are the Refresh button (allowing you to go and look for new tickets or responses) and the Attach Correspondence button (where you can respond to the ticket, adding correspondence to the selected ticket).

It also bears pointing out that in order to respond to the pending status of a ticket, you must use the pending tab in the upper window. The default state for the upper list in your working window is to select the "new" tab. Of course, a pending ticket would not exist in the "new" tab. Figure 8.6 shows Harry receiving his response from the administrator.

Trouble Tickets
Life for Education Systems

submitter	title	type	term	status	urgency	student id	student's name	checked out
harry	Incorrect fall charges?	charges/credits	083	1	0	A14223313	Dervish, Whirling F.	Admin

New Pending Fixed

submitter	type	date	talk
harry	0		Hi,  I'm not sure but I think this student was charged incorrectly. Could you check? I may have entered an incorrect date. His assignment is supposed to be effective on 8/24.  Thanks
Admin	1		I believe it is fine. The charges for Mr. Dervish were calculated properly. Please respond that you received this so I can close the ticket.

New Ticket Attach Correspondence Refresh

You have 1 pending trouble ticket.

Figure 8.6 -Harry receives a response to his ticket.

8.3 DATA RETRIEVAL

One of the most powerful features of CHARLES is its ability to pull back data for your edification and use. It contains a query generator called the Data Retrieval function.

We have already seen a version of this when we looked at population generation in the email generation function. Data retrieval uses a similar paradigm in that it allows you to generate a population for a report, but it goes a couple of steps further, allowing you to select which fields from the tables you wish to display, lets you see the results, and to save queries for your (or anyone's) later use. Even more importantly, it allows you to port your results out to a spreadsheet for formatting and presentation.

Figure 8.7 shows you the basic Data Retrieval working window. As with many other working windows, these queries are term specific. There are four tabs across the top of the window, allowing you to formulate your entire query, view it, export it and (if you'd like) save it. Let's look at each of these tabs in turn.

Tab one is called Attributes and allows you to define the population you wish to report upon. You can select from any number of the attributes check lists to fully qualify your population. In the example in figure 8.7, we are interested in some information pertaining to undergraduate freshmen who have not yet completed their applications (meaning their assignment status is "IN").

Data Retrieval

Life for Education Systems

Fall 2008

Attributes

Fields

Preview

Saved Queries

Assignment Status

☐ Assigned  
☐ Cancelled  
☒ Inactive  
☐ Medical Leave Cancellation  
☐ NonResident  
☐ Study Abroad Cancellation

Housing Types

☒ Freshman  
☐ New Grad Student  
☐ Returning Undergraduate  
☐ Transfer Student

☒ Either
☐ Include Dorm Data

Student Types

☐ Graduate Student  
☒ Undergraduate Student

School

☐ Grad Arts & Sciences  
☐ Undergrad Arts & Sciences  
☐ Undergrad Business  
☐ University without Walls

Buildings

☐ Albright-AL  
☐ Baldwin-BA  
☐ Cushing-CU  
☐ Halsey-H1  
☐ Halsey-H2  
☐ Halsey-H2  
☐ Halsey-H4  
☐ Wilder-WI

☐ Permanent Address  
☐ Current Address  
☐ Local Address  
☐ Mailing Address

Note: If you choose to include an address type, you must include the s\_id data element and must not select any data elements from the address file as they will all be included by default.

SQL Query

```
(select * from f_studentdemo,f_assignment,f_termstudent where (asnstatus='IN') and (term_stutype='UG') and (term_htype='F') and asnterm='083' and asnsid=s_id and asnactive=1 and term_term='083' and term_id=s_id)
```

Figure 8.7 -The Attributes tab allows you to qualify your population.

Data Retrieval

Life for Education Systems

Fall 2008

Attributes

Fields

Preview

Saved Queries

Available Fields

s\_id  
s\_seq  
s\_ssn  
s\_name  
s\_lastname  
s\_firstname  
s\_middlename  
s\_birthdate  
s\_gender  
s\_citizen  
s\_entered  
s\_gradate  
s\_netid  
s\_econdemail  
asnseq  
asnsid  
asnterm  
asnassigneddate  
asneffectivevdate  
asnexpiredvdate  
asnstatus  
asnwlcode  
asnwldistance  
asnsaprogram  
asnsareturn  
asncomments  
asnactive  
asnroom  
asncheckin

Fields to Include (In Order)

SQL Query

```
(select * from f_studentdemo,f_assignment,f_termstudent where (asnstatus='IN') and (term_stutype='UG') and (term_htype='F') and asnterm='083' and asnsid=s_id and asnactive=1 and term_term='083' and term_id=s_id)
```

Sort By

Clear

Drag "Available Fields" to "Fields to Include" to have them appear in your export.

Drag the field you want to sort the list by from "Fields to Include" to "Sort By."

Note that you can shift- or command-select multiple fields to drag from "Available Fields," drag fields in the "Include" list into order, drop fields on the trash to remove them, and click the trash to remove all fields.

Figure 8.8 -The Fields tab allows you to select specific data elements to include.

Tab two is called Fields and allows you to pick the specific pieces of data you'd like to see in your result. Figure 8.8 shows the fields tab for the population we defined in figure 8.7.

You'll notice that according to the population you define, *CHARLES* is able to build a list of all the data elements contained in the tables of your population. *CHARLES* will select all the data elements as a default output set. So, in order to only view the pieces of data you are interested in, you should select those fields from the list and drag them from the Available Fields list to the Fields to include list. Instructions are printed on the window for how to use this. You'll also notice that your query appears on this tab for you to view (and recall) as you choose fields. Figure 8.8 shows the tab with some fields selected and a sort field chosen.

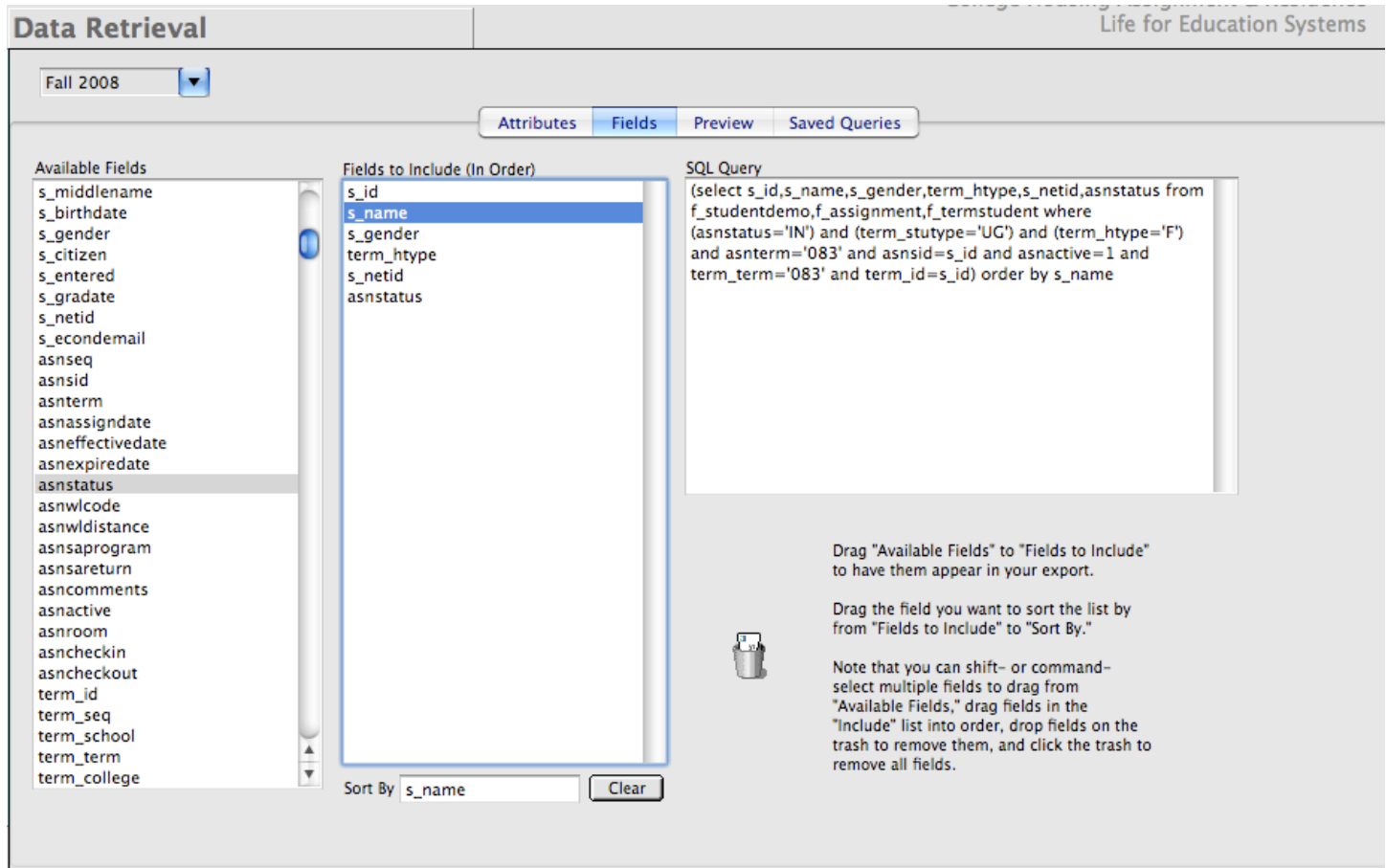


Figure 8.9 -The same query as figure 8.8 with some specific data elements chosen.

You may notice that the sql query changes as you drag data elements from the Available Fields list to the Fields to include list. This should cause the data retrieval process to run faster as it won't have to select all fields from all the relevant tables.

Once you are satisfied that you have all of the data elements you require, you are ready to select the preview tab. This is where you are able to view, save and export your results (figure 8.10). Once you select the preview tab, you are presented with the same sql query and a few buttons. Press the "Preview" button to fetch your results.

The results appear in the lower list on the working window. Notice that all the field names appear across the top as column headings and the data for those elements appears below. Our example has only found one record, as indicated in the Total Lines display at the bottom of the results list.

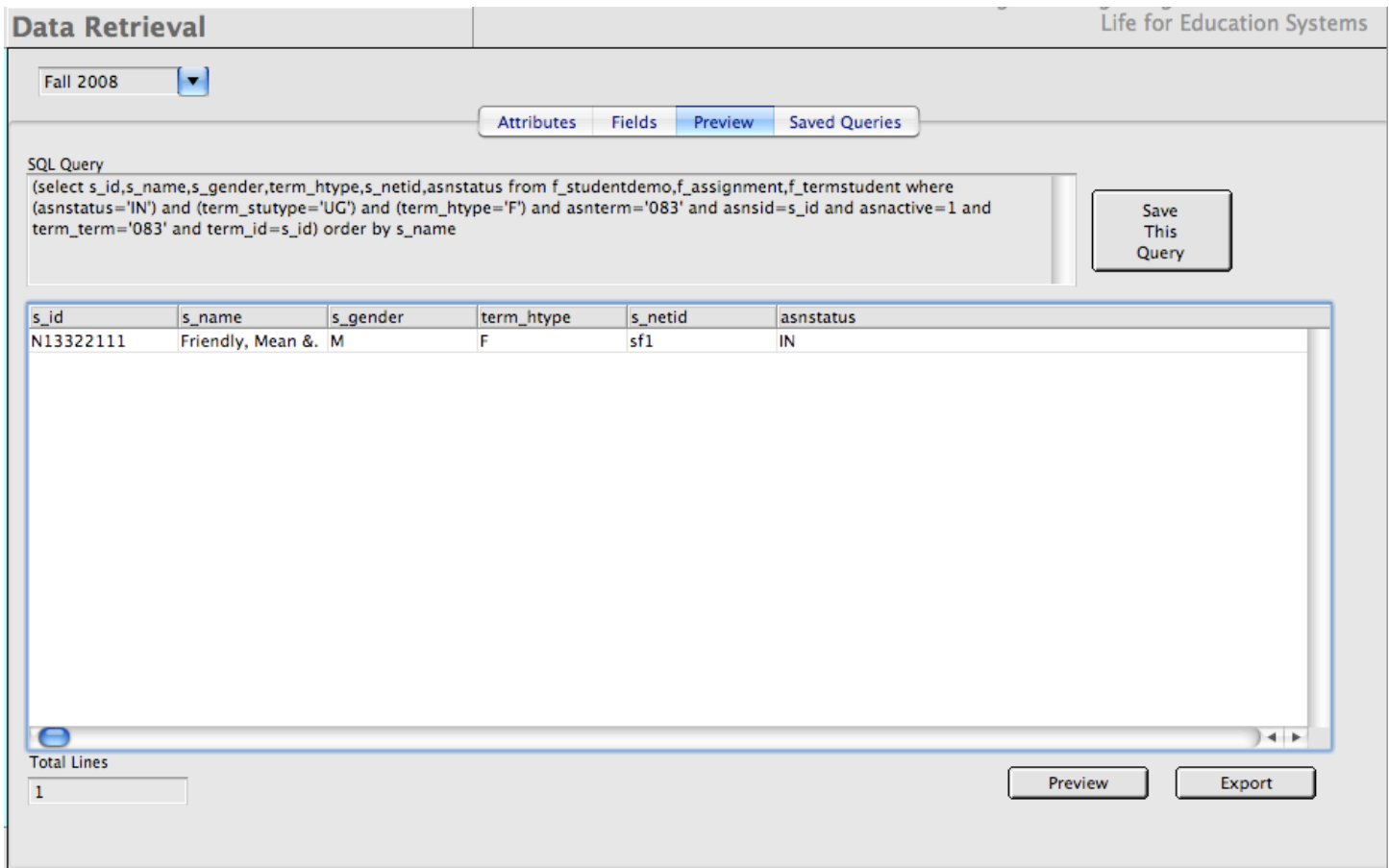


Figure 8.10 -The preview tab is where you can look at results, export them or save your query.

If you are satisfied with the results, press the Export button to send these results to a file. You are prompted for a name and location for your export. (figure 8.11)

If you think this is a query you'd like to run often and would like to save it, press the "Save Query" button.

When you save a query, you have the option to save it for yourself or as a shared query. Saving it for yourself insures that no other user will be able to access this query (unless they build it themselves). Saving it as shared gives all system users (who have data retrieval access) the ability to run the

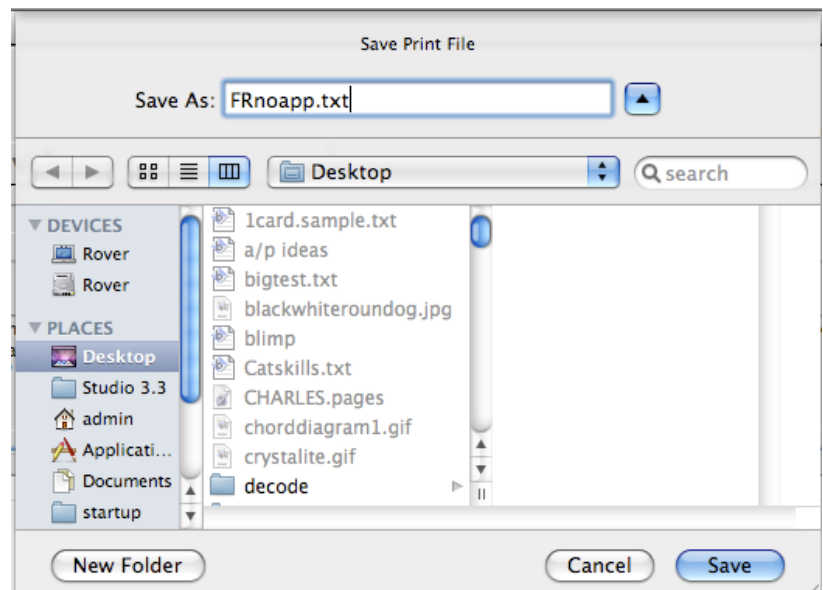


Figure 8.11 -A Macintosh prompt to save your query results.

query as well. Figure 8.12 shows the save query dialogue and its options. Note that we have given the query a name to help describe what it does.

One convenient thing about saving queries is that, due to the fluid nature of the database and your housing cycle, results from saved queries will change over time. You could run this query throughout



your cycle to determine how many potential freshmen applications you can expect to receive - or even as a device to determine when you no longer expect to receive an application and can, therefore, cancel any inactive freshmen records. Of course, the way you deal with your data handling is up to your institution. We simply want you to be aware of the possibilities that *CHARLES* presents.

The final tab on the Data Retrieval working window is (you may have guessed) where you go to utilize the saved queries. Figure 8.12 shows the tab.

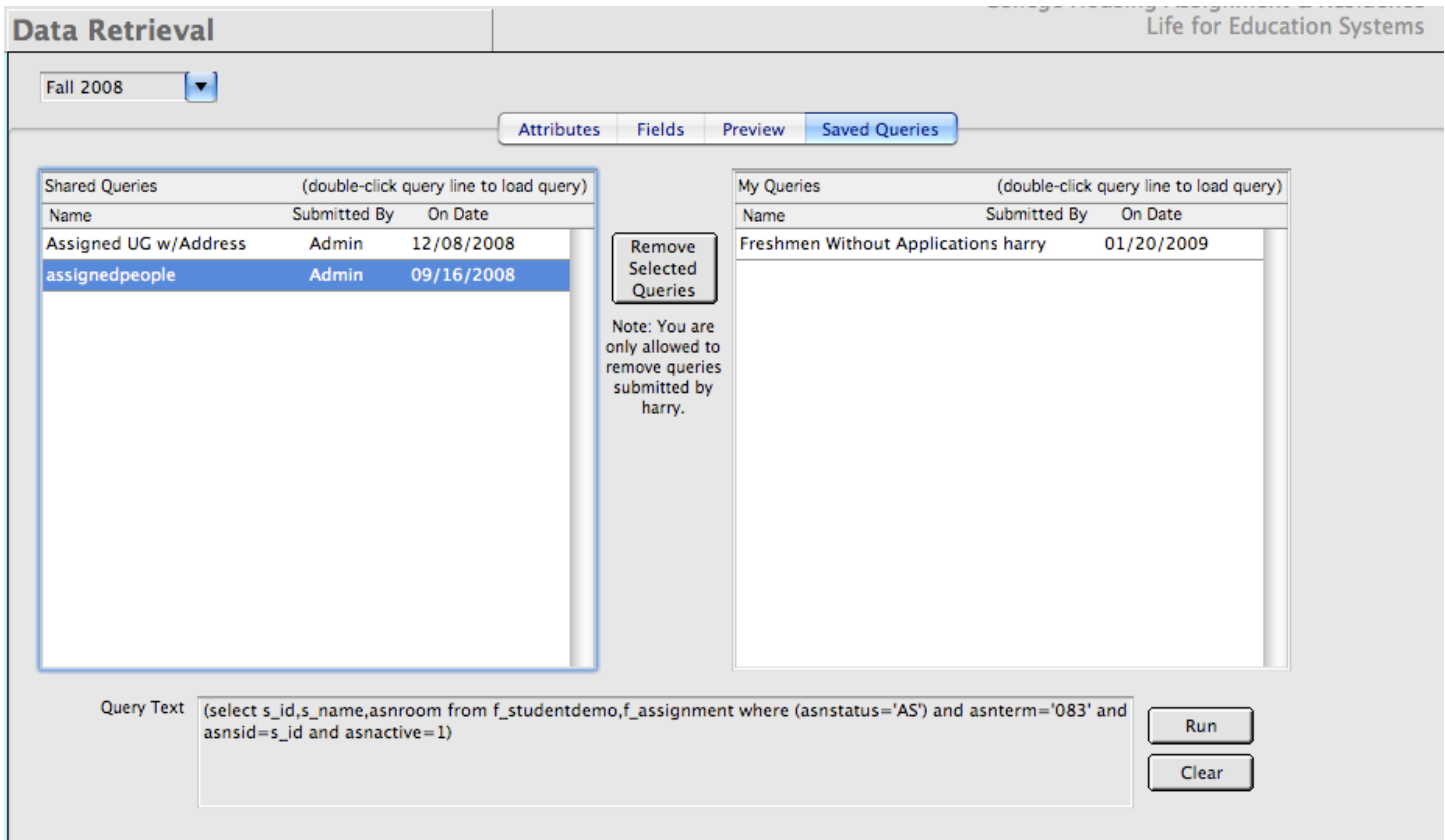


Figure 8.12 -The Saved Queries tab with a shared query loaded.

On the saved queries tab, shared queries appear in the left list while your own (not shared or private) queries appear on the right. To load a query, double-click the query line in either of the lists. You'll see the sql for the query appear in the lower query text box.

Once you have loaded a query, you press the "Run" button to retrieve the results. When you do this, the working window automatically selects the preview tab, fetches the results and redraws the working window. You will then be able to export the results to a file as though you were the one who created the query. (figure 8.13)

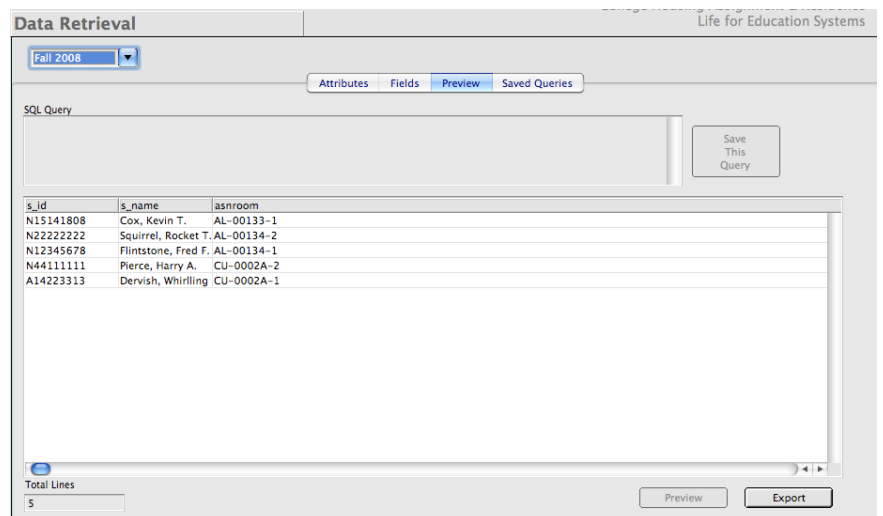


Figure 8.13 -Pressing "Run" from the saved query tab runs the query.

Once your query is exported or you've finished using the data results, you are free to go to the saved queries tab and run another, or to begin building another query. You may actually find retrieving data to be a fun, useful and interesting exercise as you traverse your housing cycle.

## 9. Summer Housing

Summer housing is one of those pieces of the cycle that lives in the netherworld. As the developer of a housing system, we've always been a bit confused about how different institutions deal with the space of time between the end of spring term and the beginning of a new academic year. Since we've seen so many different methods of space use during this period, we've decide to include two distinct methods for handling summer housing and one sub-method (which will be discussed in turn).

Since *CHARLES* was developed based on experience gleaned from a major urban university, whose standard practice was to handle summer as a succession of weekly assignments, our main summer housing module is based on this paradigm. Other institutions handle summer as just another academic term, although perhaps with fewer days and more intensive classes. In short, *CHARLES* allows you to pick the best match for your institution and handle things as you see fit.

Your system administrator is given the option of handling summer one way or the other. In the Systems Administration functions of your system, there is a function called Institution Specifics. This is where your system administrator sets up the customizable look and feel of your version of *CHARLES*. (figure 9.1)

**Institution Specifics** Life for Education Systems

Acronym: BEST Full Name: Best College of Music Somewhere US Abbreviated Name: Best Music College [Insert]

System Preferences Pictures Messages

**System Preferences**

Fall 2008 (Default Term) 1.6 (Current Version)

**Billing System**

- ☒ Use Internal Billing System
- ☐ Export to IA/SIS Billing
- ☐ Export to SCT/Banner Billing
- ☐ Export to Other Billing

**Meal Plans**

- ☒ Do not use Meal Plan Functions
- ☐ Export to Diebold/Griffin System
- ☐ Export to Blackboard System
- ☐ Use Internal Punch Cards

**Registration System**

- ☒ Do not export addresses
- ☐ Export to IA/SIS System
- ☐ Export to SCT/Banner System
- ☐ Export to Other System

**Facilities System**

- ☒ DO NOT Include Facilities Tracking
- ☐ Include Facilities Tracking

SMTP server: smtp.bestcollege.edu

eMail Return Address: housing@bestcollege.edu

eMail Suffix: @bestcollege.edu

SMTP User Name: admin (for authentication)

SMTP User Password: (for authentication)

Web Home: http://www.bestcollege.edu

☐ Use SSN as Student ID ☐ System allows smoking ☒ System generates estimates ☐ Summer as Academic Term

[Find] [Edit] [Delete] [OK] [Cancel]

Figure 9.1 - The first tab of the Institution Specifics set-up working window.

As you can see from this working window, our test college's version of *CHARLES* is set up to use this group of parameters when it handles data. Your system administrator sets this up as one of the first processes upon initial database setup. This way, as users are added to your system, their installations of *CHARLES* will be sure to follow the rules governed by these parameters.

Note the series of check boxes at the bottom of the working window. It is called Summer as Academic Term. If this is checked, you would be able to handle summer housing as though it was just another academic term, meaning all processes for summer housing would be handled under the higher level functions called “Academic Year Housing”. Summer Housing, as a high level group of functions would never appear in your functions list.

There, that was easy! We’ve already taken care of one of the two aspects of how institutions handle summer housing. Simply refer to all of the previous chapters to refresh yourself of how to handle applications, assignments, correspondence, etc.

But notice from figure 9.1 that the “Summer as Academic Term” box is not checked. In fact, the default handling for summer housing in *CHARLES* is to **not** handle summer housing this way, but to handle it the other way - as a series of weekly assignments over a period of weeks. This is what the rest of this chapter will deal with. So if your institution is happy to treat summer as just another academic term, you may move on to chapter 10. (But you may, through continued reading of this chapter, find that the weekly method is more intuitive or better fits what you’d like to see your institution do with summer periods. It’s up to you.)

## 9.1 THE WEEKLY THEORY

The summer term setup is basically the same as for academic year terms. Your system administrator enters a start date, an ending date, cancellation deadlines and other pieces of information to fully qualify the term. More on this in the System Administration chapter.

Once the term is set up, *CHARLES* calculates the number of days and weeks to include in the term. *CHARLES* now knows how many weeks can be used for assignments into all of its spaces.

One advantage of this weekly methodology is that it offers a lot of flexibility in the way spaces can be allocated. Due to the transient nature of many summer housing schemes where people are allowed to stay for varying lengths of time, *CHARLES* is able to assign residents to their desired hall, regardless of the bed space they will ultimately end up in - a sort of reservation to a particular hall in lieu of an actual room assignment. This allows you to calculate charges and bill residents well in advance of the beginning of the summer. Hall managers can then (when they have a moment) disperse these reservations into actual bed spaces.

Therefore, a side effect of the weekly methodology is to share the workload (a bit) with your residence hall managers. You may think this is a good idea or you may not. We recognize that you may not want the residence hall managers to handle your room assignments. Of course it is ultimately up to you how you’d like to task your staff so this is merely a suggestion. We bring it up here because it feeds into the way the function list is organized. Summer room assignment functions reside in the Residence Life high level functions of your functions list. So whether you decide to handle room assignments centrally or allow your halls to handle that part of the process, you’ll know where to find these functions and you’ll have to assign proper privileges to whichever staff members you assign these tasks.

In figure 9.2, we see our user’s (Harry) function list. He has access to both Summer Housing and Residence Life functions so he can do everything needed for summer housing. Note that the

functions in the list are similar to what we saw in Academic Year Housing functions, although the fact that they exist here means that they behave a bit differently.

We have access to reports, charges, a specially designed inquiry window, a data retrieval function and access to spaces for the summer. There is also an Applications area that contains our manual methods for entering applications, as well as tracking payments, groups and a new concept called Short Term Summer Housing.

## 9.2 MORE ON THE WEEKLY THEORY

The weekly theory of summer housing came from a need to assign students based on a two session academic term where students were allowed to take courses during the summer in one of two (or both) six-week sessions. Since students were enrolled for at least six weeks, *CHARLES* was designed to handle weekly assignments: in fact, residents could be assigned for only six

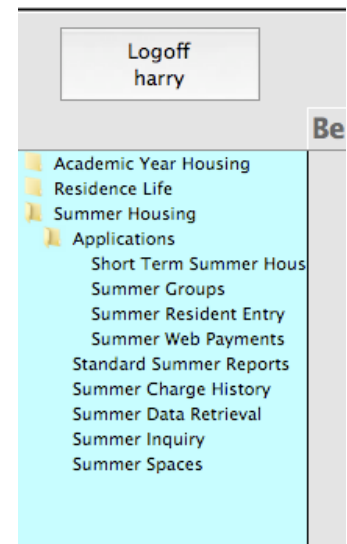


Figure 9.2 - Harry's function list.

weeks (either the first six or the last six) or for the entire twelve weeks. We called this “Long-Term” housing. Then someone said we should allow shorter term stays in summer housing - in case someone wanted to spend a few weeks to do an internship or attend a conference. So we instituted a three-week minimum stay. These could be any of the three weeks encompassed by the term starting and ending dates. Hence, *CHARLES* is set up to handle all of these scenarios.

But then we decided to extend the functionality to allow people to stay for even shorter stays, encompassing maybe a part of a week so they could attend a conference or symposium. So we allowed spaces to be tagged specifically for that use and installed Short Term Summer Housing functions to handle these shorter term (usually less than one week) stays.

It may seem that things are getting too complicated with all of these differing time lines. It really isn't so complicated as long as you keep in mind that there are two distinct types of summer housing stays: Short Term and Long Term.

## 9.3 LONG TERM SUMMER HOUSING

There are several types of potential applicant for long term summer housing. A potential resident could be a student (either from your institution or not) wanting to take some courses. Or they could be someone who is in your area for a vacation or visiting friends (or any number of reasons) and has no interest in taking classes. Or, it could be one of your regular academic year residents who, for whatever reason, does not want to go home for the summer (maybe they have a job they'd like to keep or are working for a professor). The long term summer housing system allows you to distribute these people into the following categories:

- 1) Enrolled - Affiliated: A person who enrolls at the institution during the summer term and is also a matriculated student at your institution pursuing a degree.
- 2) Enrolled - Unaffiliated: A person who enrolls at the institution during the summer term and is not a matriculated student at your institution, but is taking courses for credits to be transferred.

- 3) Not Enrolled - Affiliated: A person who is not enrolled for summer, but is a matriculated student at your institution and wants to be housed on-campus during the summer.
- 4) Not Enrolled - Not Affiliated: A person with no affiliation with your institution and is not taking classes but would like to stay in your housing system for a period of time during the summer.

Of course you may not allow all of these types of people to have housing at your institution during the summer. That is up to you. *CHARLES* simply gives you the option.

Figure 9.3 shows you the Summer Resident Entry working window.

Figure 9.3 - The summer resident entry working window.

Because *CHARLES* is set up to allow you to accept applicants from outside of your institution, we thought it would be important to get as much information about them as we could so they could be billed properly, in advance, and that we would make them pay something up front. For example, looking at the working window, we see that we are asking for two addresses - permanent and current. We also ask whether they are enrolled (the Points data element) and if they are an academic year student or have ever lived in housing at the institution. The combination of these items is how we are able to associate the applicants into the afore mentioned categories.

Note that the assignment section has a category and a hall drop list. *CHARLES* allows you to further categorize your residence halls into groupings of hall types - which can be assigned varying rates. For instance, a traditional dormitory style hall (no kitchen) with air conditioning (an important distinction during summer months in the US) could carry a different rate than a traditional dormitory style hall without air conditioning. These categories are set up by your system administrator for summer use. Figure 9.4 shows the layout of this setup window.

Figure 9.4 - Systems Administrators set up categories for summer residence hall classification.

In the above example, we have specified a category code of “D” for traditional style halls with air conditioning. We have also pointed out that this code is a single occupancy code and we’ve selected some number of charge types to associate with this code (in this case “S-Single Charge” is selected).

Looking back at figure 9.3, these categories are what will appear in the categories drop list. Halls are affiliated with these categories through the rates that are set up for these categories. This allows you to offer applicants a choice of residence hall types on their applications - you don’t have to ask them for a specific hall (although you can if you’d like). Also, it allows you the ability to formulate rates by hall type - not just by room type (or charge type) as in the Academic Year functions.

When summer housing rates are entered into the rate table, it is these categories you assign the rates to. For illustration purposes, figure 9.5 is a representation of the rate entry working window from the systems administrator functions. You can see that when we construct summer rates, we choose the term, hall and category (rather than a charge type) for the rate. We also utilize Rate 1 and Rate 2. For Long Term summer housing, *CHARLES* maps rate 1 to our “enrolled” population and rate 2 to our “not-enrolled” group. This gives additional flexibility in your rate choices. For instance, you could offer discounted housing rates to residents who take classes - perhaps providing an incentive for them to enroll.

When a category is selected at the time of assignment, the hall drop list is built containing only those halls who have rates matching the category chosen. *CHARLES* uses the category selected, the hall chosen and the number of points reported to figure the calculated charge you see in figure 9.3. Another way to state this is that *CHARLES* bases its rates on a combination of hall type, enrollment

status and occupancy type. This will most likely handle any combination of long-term residents you might attract.

**Rates** Life for Education Systems

Summer 2008 Rate Type Room Cushing

D - Trad Hall-A/C-Single Occ

Rate Code Cus-D Account CU-08-11

Alt Account Number

Description Cushing Single

Your system provides 2 types of rates to enter: Room Rates and Estimates.

Room Rates must be entered for each hall and key off of charge types from the charge type table.

Estimates key off of student housing types.

Rate	Rate
Rate 1	100.00
Rate 2	150.00
Rate 3	
Rate 4	
Rate 5	
Rate 6	
Rate 7	
Rate 8	
Rate 9	
Rate 10	
Rate 11	
Rate 12	
Rate 13	
Rate 14	
Rate 15	

Insert Find Edit Next Prior Delete Roll OK Cancel

Figure 9.5 - Systems Administrators set up rates and associate them with the categories.

Again, looking back at figure 9.3, the application is entered in two parts. The entire working window above the assignment portion is the resident's application. As with other application entries in *CHARLES*, press the Insert button and fill out the fields, tabbing between them until you reach the end. Then press return. The application is added to the database after testing for completeness.

In this case, however, you are prompted as to whether you'd like to process the assignment and/or payment. Figure 9.6 shows this prompt.

CHA College Housing Assign Life for Education Systems

Summer Resident Entry

Term Summer 2008

SSN A14-22-3313 University ID Dervish Last Name Whirling First Name F M Gender OCT 15 1989 Birth Date

wfd1@bestmusic.edu Email Address (646) 555-1122 Cell Phone

Permanent Address 523 Park Place Apartment #34 Spin City MA 01100-2323 US - United States (413) 555-3223 Phone

Current Address 8822 Main Loop Baltimore MD 21222- US - United States (646) 555-1122 Phone

Valid Until MAY 18 2008

Dervish, Mervish Emergency Contact Person FA - Father Relation (413) 555-3223 Home Phone (413) 555-3223 Work Phone

YES Academic Year Student YES Ever Live In Housing No Group - Long-Term Group

3 Points 1 Rate MAY 18 2008 Arrival Date JUN 28 2008 Departure Date

Roommates Arriving

Assignment - Select Category Category Hall Assignment Date

Payment Selected Phone Deferral Select Pay Type Payment Type Payment Amount

The application has been entered. Are you ready to process the assignment? No Yes

Figure 9.6 - Once the application is entered, you are prompted to assign.



Selecting no on the prompt leaves the resident with an Unassigned (“UN”) assignment status. If you answer Yes to the prompt, you continue by selecting the hall category and the actual hall name. When you are doing this, *CHARLES* is looking through the rate tables and calculating the charge based on your selections. Once you are sure of your selections, *CHARLES* checks the system for available space for all the weeks for which your resident needs an assignment. If it checks out you are asked to confirm the assignment. If you do confirm it (answering yes to the prompt), the resident is assigned and you are presented with the opportunity to process the payment (figure 9.7). Otherwise, you’ll be prompted for another category and hall choice.

*CHARLES* essentially walks you through the steps of entering the application, processing the assignment and logging the payment by prompting you with questions along the way. Notice that as you are entering data into the working window fields, the OK and Cancel buttons become active. If you need to stop your entry, you may press Cancel at any time. Once you are finished with your application entry, assignment and payment

Figure 9.7 - Finished with the assignment on the way to processing payment.

Figure 9.8 - Our completed application - with assignment - viewed on the Summer Inquiry working window.

processing, you may view the results using the Summer Inquiry function. Figure 9.8 shows the record we just entered.

The Summer Inquiry working window looks a lot like Academic Year Student Inquiry. There are some notable differences however. First, we do not display roommates. We are able, instead, to see the resident's hall assignments - one for each week in residence. We can also quickly see what the student type is (the "E" is for enrolled), their rate (1 for enrolled), and a quick view set of checked boxes indicating the weeks in residence.

But also notice the tabs in the lower tab pane of the working window. The selected tab in figure 9.8 is "Current Status". We are also able to make adjustments, view charges and payments, attach notes through the Correspondence tab, view room history and other Miscellaneous items. In fact, we can do a lot more in this inquiry area than we could in the Academic Year one.

We'll discuss this a bit more - but it's also important to point out that nowhere in figure 9.8 do we see a room or bed space assignment. That's because we haven't got to that step yet. Even though we have been able to reserve space for our resident in a particular hall with a particular room type, generate charges that can be billed, and collected a payment for the person's summer stay, we have not had to assign a bed space. In fact, we are able to wait until the last minute to assign our student to a particular bed space. We could even wait until he/she checks in (although that's probably pushing it). Figure 9.9 shows the charges and payments tab for our resident.

The screenshot shows the 'Summer Inquiry' window for 'Life for Education Systems'. The top section contains student information: Summer 2008, Student ID A14-22-3313, Name Dervish, Whirling F., Gender Male, Birthdate OCT 15 1989, and NetID wfd1. Below this is a table of addresses with columns for Type, Street, City, State, Zip, Country, and Phone. The 'Charges/Payments' tab is selected, showing a 'Charges' table with columns for Code, Description, Sub-Code, Effective, Processed On, End Date, and Amount. A single charge is listed: Cus-D, Cushing Single, U-08-1 D, MAY 18 2008, JAN 21 2009, AUG 9 2008, 600.00. Below the charges is a 'Payments' table with columns for Type, Description, Paid, Processed, Transmitted, Batch, UserID, and Amount. A single payment is listed: Check, Initial Payment, APR 30 2008, Never, harry, 350.00. At the bottom right, a total of 250.00 is displayed. Navigation buttons like 'Find Student', 'Next', 'Prior', 'Print Statement', 'OK', and 'Cancel' are visible.

Type	Street	City	State	Zip	Country	Phone
P	523 Park Place Apartment #34	Spin City	MA	01100-2323	US	(413) 555-3223
M	P.O. Box 332 Friendly Station	Spin City	MA	11200-	US	
-	8822 Main Loop	Baltimore	MD	21222-	US	(646) 555-1122

Code	Description	Sub-Code	Effective	Processed On	End Date	Amount
Cus-D	Cushing Single	U-08-1 D	MAY 18 2008	JAN 21 2009	AUG 9 2008	600.00

Type	Description	Paid	Processed	Transmitted	Batch	UserID	Amount
Check	Initial Payment	APR 30 2008		Never		harry	350.00

250.00

Figure 9.9 - Even though our resident has no bed space assigned, we have a complete billing record.

We can bill and track our residents without actually having to assign them. Lets look at how to find them a bed space for when they check in.

As we stated earlier, the room (or bed space) assignment functions reside in the Residence Hall high level functions. Figure 9.10 shows Harry's function list with the Residence Life functions exposed. For now, we will only deal with the functions that pertain to summer housing.

Summer Room Assignments loads the working window you see in figure 9.11. There are two tabs containing the functions you might want to utilize, Initial Assignment and Move Assignment. To assign a resident into a bed space, choose the hall from the drop list and choose a week from the drop list. As you do this, *CHARLES* looks to see if there are any people without bed spaces in the reservations for the hall and draws them in the list. In our example, we have selected the hall named "Cushing" and the first week to see all people needing assignments for week one having a reservation for "Cushing". Once we have a list of people who need bed spaces assigned, we only have to click on the person we want to assign. The available spaces

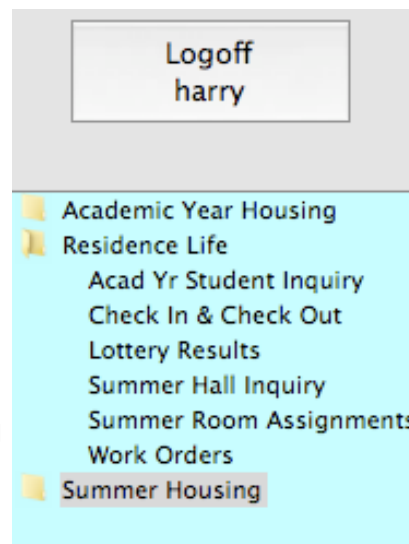


Figure 9.10 - The Residence Life functions.

list on the right finds all spaces that match the resident's reservation type (in this case, single occupancy) and gender. Our example in figure 9.11 shows that *CHARLES* found three possible spaces that it could utilize.

Once we have a list of people who need bed spaces assigned, we only have to click on the person we want to assign. The available spaces list on the right finds all spaces that match the resident's

Assignments Needed: 1												
	1	2	3	4	5	6	7	8	9	10	11	12
Dervish, Whirling F.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cushing	Single	Male										

Available Spaces												
	1	2	3	4	5	6	7	8	9	10	11	12
CU-0004A-1	Used:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Single	Gender:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CU-0005A-1	Used:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Single	Gender:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CU-0005B-1	Used:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Single	Gender:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 9.11 - Finding bed spaces for residents who don't have them.



Again, we see that we must select the hall we are interested in dealing with. Once the hall is selected, *CHARLES* looks for all bed space assignments for that hall and lists them in alphabetical order in the “Participants Assigned” list. In our case, we have only one assignment in “Cushing” for the summer and it is our complaining resident. If we click on the resident in the list, another list of bed space assignments appears the right, as well as a list of spaces below. We can choose any of the spaces that appear in the “Available Spaces” list to move our resident into. When we do, the check boxes in the upper right list are checked, if those particular weeks work for our resident. *CHARLES* also builds an action text box indicating what it expects to happen if you approve this move.

The screenshot shows the "Summer Room Assignments" window for "Summer 2008". It has tabs for "Initial Assignment" and "Move Assignment". A dropdown menu shows "Cushing" selected. A "Match Gender" checkbox is checked. On the left, the "Participants Assigned" list shows "Dervish, Whirling F." with a "Single" gender tag and a row of checkboxes. On the right, a table lists assignments for "Dervish, Whirling F." across weeks 1-6 in the "Cushing" hall, all in "CU-0005A-1" bedspace. A text box indicates the move to "CU-0004A-1" for weeks 1-6. At the bottom, the "Available Spaces" list shows "CU-0004A-1" (Single, Gender: ) and "CU-0005A-1" (Single, Gender: M M M M M M). A "Move" button is at the bottom right.

Figure 9.13 - Moving our unhappy resident to another similar bed space.

The final step is just to press the “Move” button. The resident is moved to the space for the appropriate weeks, the old space’s gender tags are cleared (providing there is no one else in the suite) and the old space is made available for another assignment for all weeks involved.

You may have also noticed that there is a Summer Hall Inquiry function in the function list under the Residence Life high level functions. This is similar to the Summer Inquiry function in the Summer Housing high level functions, but it is somewhat more limited in what can be done and seen about summer residents. It presumes you are probably only interested in one residence hall at a time. It does not allow you to view resident charges and histories. It does, however, allow you to view an entire assignment sheet (or grid) for an entire hall at once. Figure 9.14 is the Summer Hall Inquiry working window. Figure 9.15 shows the room grid for our tiny hall named “Cushing”.

Note in figure 9.15 that we can see the entire hall with all of its spaces and all the people assigned to them in the boxes. We can also see the occupancy types of the bed spaces. It is important to point

Summer Hall Inquiry

Life for Education Systems

Summer 2008

Participants Room Grid

A14-22-3313

Dervish, Whirling F.

Male

OCT 15 1989

wfd1

Student ID

Name

Gender

Birthdate

NetID

Type	Street	City	State	Zip	Country	Phone
P	523 Park Place Apartment #34	Spin City	MA	01100-2323	US	(413) 555-3223
M	P.O. Box 332 Friendly Station	Spin City	MA	11200-	US	
-	8822 Main Loop	Baltimore	MD	21222-	US	(646) 555-1122

SU-E

1

MAY 18 2008

JUN 28 2008

X X X X X X

Student Type

Rate

Arrival

Departure

1

2

3

4

5

6

7

8

9

10

11

12

Group Code

Email Address:

wfd1@bestmusic.edu

Cell Phone:

(646) 555-1122

Emergency Contact

Dervish, Mervish

FA

(413) 555-3223

(413) 555-3223

Person to Contact

Relation

Phone Number

Alternate Phone

Assignment

AS

JAN 21 2009

APR 30 2008

Status

Assignment Date

Effective Date

Confirmation

Application Date

Hall Assignments

Week	Hall	Room	Plan	Asgn Date
1	Cushing	CU-0004A-1	Single	JAN 21 2009
2	Cushing	CU-0004A-1	Single	JAN 21 2009
3	Cushing	CU-0004A-1	Single	JAN 21 2009
4	Cushing	CU-0004A-1	Single	JAN 21 2009
5	Cushing	CU-0004A-1	Single	JAN 21 2009
6	Cushing	CU-0004A-1	Single	JAN 21 2009

Find

OK

Cancel

Figure 9.14 - The Summer Hall Inquiry working window.

Summer Hall Inquiry

Life for Education System

Summer 2008

Participants Room Grid

Cushing

Bedspace	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
CU-0002A-1 Multiple												
CU-0002A-2 Multiple												
CU-0002B-1 Single												
CU-0003A-1 Single												
CU-0003B-1 Multiple												
CU-0003B-2 Multiple												
CU-0004A-1 Single	Dervish, Whirling F.	Dervish, Whirling F.	Dervish, Whirling F.	Dervish, Whirling F.	Dervish, Whirling F.	Dervish, Whirling F.						

Figure 9.15 - The room grid of the Summer Hall Inquiry working window.



out that you can have different occupancy types within a single suite. You can best utilize your “natural” singles this way, as well as maximize your income. It is also important to point out that summer housing spaces are all tagged with charge types which indicate their occupancy use. The codes that can be entered in summer housing bed spaces are limited to “M” for multiple or “S” for single occupancy.

There are only a few more things we should cover in this chapter. One is making adjustments to a resident’s summer housing stay. The others are Summer Housing Groups and Short Term summer housing.

Summer housing adjustments are handled in the Summer Inquiry functions, utilizing the tab called Adjustments. (figure 9.16)

**Summer Inquiry** Life for Education Systems

Summer 2008 Student ID: A14-22-3313 Name: Dervish, Whirling F. Gender: Male Birthdate: OCT 15 1989 NetID: wfd1

Type	Street	City	State	Zip	Country	Phone
P	523 Park Place Apartment #34	Spin City	MA	01100-2323	US	(413) 555-3223
M	P.O. Box 332 Friendly Station	Spin City	MA	11200-	US	
-	8822 Main Loop	Baltimore	MD	21222-	US	(646) 555-1122

Current Status **Adjustments** Charges/Payments Correspondence Room History Miscellaneous

**Current**

	Building	Plan	Charges
1	<input checked="" type="checkbox"/> Cushing	D	100.00
2	<input checked="" type="checkbox"/> Cushing	D	100.00
3	<input checked="" type="checkbox"/> Cushing	D	100.00
4	<input checked="" type="checkbox"/> Cushing	D	100.00
5	<input checked="" type="checkbox"/> Cushing	D	100.00
6	<input checked="" type="checkbox"/> Cushing	D	100.00
7	<input type="checkbox"/>		
8	<input type="checkbox"/>		
9	<input type="checkbox"/>		
10	<input type="checkbox"/>		
11	<input type="checkbox"/>		
12	<input type="checkbox"/>		

600.00

**Changes**

	Building	Plan	Charges
<input checked="" type="checkbox"/> 1	Cushing	D	100.00
<input checked="" type="checkbox"/> 2	Cushing	D	100.00
<input checked="" type="checkbox"/> 3	Cushing	D	100.00
<input checked="" type="checkbox"/> 4	Cushing	D	100.00
<input checked="" type="checkbox"/> 5	Cushing	D	100.00
<input checked="" type="checkbox"/> 6	Cushing	D	100.00
<input type="checkbox"/> 7			
<input type="checkbox"/> 8			
<input type="checkbox"/> 9			
<input type="checkbox"/> 10			
<input type="checkbox"/> 11			
<input type="checkbox"/> 12			

600.00

Adjust Cancel

Rate Adjustment 1 Current Rate Extension Fee Cancel Fee Received Date

Find Student Next Prior OK Cancel

Figure 9.16 - Making an adjustment to a resident’s housing stay.

On the Adjustments tab, you can view all the hall assignments made for the resident. There are two representations of this - one to make changes in (the right side called “Changes”) and one to view what you are changing from (the left side called “Current”) In order to make a change, press the Adjust button. You will then be able to enter your changes.

For example, suppose our resident would like to move to a multiple occupancy room in another hall for weeks 7 and 8. We would press Adjust, check week 7, choose a hall and a plan, check week 8, choose a hall and plan, then press return. (figure 9.17) We enter a received date and the adjustment is done. Remember that the Residence Life functions will need to be looked at for week 7 in order to place the resident into a bed space.

Cancellations are handled similarly. If you are canceling the resident for the entire summer, press the Cancel button and follow the prompts. If you are canceling a portion of the resident's stay, press Adjust and uncheck the weeks the resident will no longer be in housing. You will always need to enter a received date for your changes and will be prompted to enter either an extension fee or a cancellation fee (if your institution utilizes such things). If you don't charge fees for extensions or cancellations, just leave the data elements blank and press OK or Cancel.

## 9.4 SUMMER HOUSING GROUPS

Summer Housing groups are handled in much the same way as Academic Year groups with a few important differences. As you may recall from chapter 8, *CHARLES* allows you to create groups, into which you can associate similar types of people, give them special rates and generally keep track of them (like fraternity members or special study organizations and the like). Summer Housing groups do the same things, allowing you to add a group and associate your residents to the group. A big difference, however, is that in summer housing, the group has the ability to either choose the individuals to pay for themselves (a non-pay group) or amass all of the participant's charges unto themselves (a group pay group).

Figure 9.17 shows the Summer Groups function's working window. At first glance, it may look a lot like the Academic Year Group Entry working window. There are some important differences however.

**Summer Groups** Life for Education Systems

Summer 2008 (dropdown) CU2 Group Code Cushing Two Group Name

Contact: Heddy Lamar Phone: (212) 999-1212

Street Address: 123 Main

City, State, Postcode: New York NY 10003

Group Email: heddy@thatshedley.com

☐ Institution affiliated Entered on:

**Term Group Data** Correspondence

Arrival Date: MAY 18 2008 Departure Date: JUN 14 2008

GroupPay: ☒ N12-29-9999

Reservations: ☐

Multiple Occ: 2 Plan: A Hall: Baldwin Rate: 2

Single Occ: 2 Plan: B Hall: Baldwin

**Group Members** Payments

Name	ID	Gender	Assigned Room	Assignment Status
Squirrel, Rocket T.	N22-22-2222	M		UN

Print

Find, Insert, Edit, Reserve, UnReserve, Next, Prior, Delete, Process Payment, Print Charges, OK, Cancel

Figure 9.17 - Summer Groups working window with a Group Pay group.

The upper portion of the working window looks much the same as we saw in chapter 8. The term group data tab, however, gives you the option to affix arrival and departure dates, indicate whether the group will be responsible for its members associated charges (the group pay check box), and enter the number of expected multiple or single occupancy members you will attach to the group. You



can indicate the plan and hall you expect to assign them to, as well as the rate all group members will be charged.

Also take note of the functions down the right side of the working window. *CHARLES* gives you the option of making reservations for the members of the group, in lieu of having received their individual applications.

Our example in figure 9.16 shows a group that will be in summer housing for the first four weeks. The group expects to have two multiple occupancy residents and two single occupancy residents utilizing the plans A and B respectively. The group members will all be assigned to a hall named “Baldwin” and be charged the amounts in rate 2 of the rate table. Their charges will be assigned to the group whose ID number in *CHARLES* is “N12-29-9999”.

When you reserve space for the group, *CHARLES* removes space from the inventory of available spaces as though an actual assignment has been made. As you assign group members (the same way you assign non-group members), *CHARLES* sees that the person is affiliated with the group and will utilize the group reservations (if you so stipulate). Figure 9.18 shows Harry making one of these assignments.

The screenshot displays the CHARLES system interface. A modal dialog box is centered on the screen, asking: "Rocket Squirrel is assigned to group CU2 which has reserved spaces. Assign him to one of those reserved spaces?". The dialog has "No" and "Yes" buttons. The background shows a student profile for "Squirrel" with the following details:

- Term:** Summer 2008
- SSN:** 123-45-6789
- University ID:** N22-22-2222
- Last Name:** Squirrel
- First Name:** Rocket
- MI:** T
- Gender:** M
- Email Address:** rocky@frostbite.com
- Cell Phone:** (646) 123-2222
- Permanent Address:** 123 Tree Road, Frostbite Falls, MN 11111, US - United States
- Current Address:** 123 Tree Road, Frostbite Falls, MN 11111, US - United States
- Phone:** (212) 222-2222
- Valid Until:** MAY 18 2008
- Emergency Contact Person:** Moose, Bullwinkle
- Relation:** OT - Other
- Home Phone:** (212) 222-2222
- Work Phone:** (212) 222-2222
- Academic Year Student:** YES
- Ever Live In Housing:** NO
- Long-Term Group:** CU2 - Cushing Two
- Points:** 3
- Rate:** 2
- Arrival Date:** MAY 18 2008
- Departure Date:** JUN 14 2008
- Room Selection:** A grid of 12 rooms (1-12) with checkboxes. Rooms 1, 2, 3, and 4 are marked with an 'X'.

Figure 9.18 - Assigning a group member whose group has reservations.

Notice that in figure 9.18, *CHARLES* is asking the user whether or not to utilize one of the group’s reserved spaces. This is how the reservations get allocated. The user could decide not to assign to one of the reserved spaces, in which case the group’s reserved pool of spaces is untouched. Otherwise, the reservations for the group would be reduced by one for the type of assignment (single or multiple occupancy) you complete.

Once you have assigned the group member, a record must be charged. If the group has the group pay flag checked, the member's charges will be sent to the group's ID (otherwise it will be charged to the member). Figure 9.19 shows an example of the group's charges using the Summer Charge History function's working window.

Code	Description	Account	Debit/Credit	Effective Date	Processed Date	End Date	Amount	ID
Bal-B	Baldwin - Single Occ	66-44-33	D	05/18/2008	01/22/2009	08/09/2008	352.00	N2222222

Total: 352.00

Figure 9.19 - The Summer Group's Charge History.

As you can see, our user assigned the group participant to a Baldwin single occupancy space for the four weeks. Since the group was a group pay group, the charge is attached to the group. Notice also that the ID in the far right column is different from the group id. This allows us to keep track of the resident who was actually responsible for generating the charge.

The Summer Groups working window also allows us to enter payments received by the group and to print out a summary of charges. (figure 9.20)

Type	Description	Paid	Processed	Transmitted	Batch	UserID	Amount
Check Payment	Payment on retainer	AUG 15 2008		Never	A12	harry	400.00

Figure 9.20 - The Summer Group payments can be entered, and charges printed.

As we will see, much of the functionality we have just reviewed in Summer Group functions pour into the Short Term Summer Housing functions.

## 9.5 SHORT TERM SUMMER HOUSING

Let's begin this section with a shot of the Short Term Summer Housing working window:

Figure 9.21 - The Short Term Summer Housing working window.

As you can see, the short term working window works with short term summer groups. Groups are entered, with contact names, addresses and anticipated participants. There are also arrival and departure dates. *CHARLES* calculates the total number of bed nights the group will be in residence. The rates you enter for short term housing are free form in that you can enter any charge for your single occupancy or shared occupancy spaces - meaning you are not tied to the rate table. This allows you to recruit groups of people to help fill your spaces by offering them the best rate you can negotiate.

Note also that there is a choice of rate status. You can charge daily, weekly or a package rate. This means that the number you enter in the rate elements (single or double) will be charged daily, weekly or once for the whole stay respectively. This, we think, offers you the most flexibility.

Once you enter the parameters of the group, including your number of anticipated participants, you can reserve spaces for the group. When you reserve short term spaces for a group, *CHARLES* removes those spaces from the Long Term system for the time they are reserved. Therefore, in order to utilize the remaining time (if any) on those spaces, you should unreserve them after you have finished assigning all the short term group participants.

Once you have entered the short term group, you are able to add and delete participants. To add someone to the group, press the Add Participant button. You will see the working window jumps to the Participants tab (figure 9.22) where you can enter the participant's information.

Summer 2008

Group Information | **Participants** | Assignments | Correspondence

Group Name: Short Term One  
 Arrival: MAY 19 2008 12:00 AM  
 Departure: JUN 4 2008 12:00 AM

**New Participant**

First Name: Jamie MI: R Last Name: Jix Gender: F  
 Arrival: MAY 19 2008 Departure: JUN 4 2008  
 Type: Choose Roommate:

Group Participants (double-click list line to edit participant)

Name	Gender	Arrival	Departure	Type	Roommate	Charge
Jaq, Jimmy S	M	05/19/08	06/04/08	Shared	nobody too	57.25
Jax, Jimmy R	M	05/19/08	06/04/08	Single	nobody	80.15

Total Participants: 2 Total Charge: 137.40

Find Add Participant Delete Participant Ok Cancel

Figure 9.22 - Adding a participant to the short term group.

You may notice that we don't have to enter id numbers for short term group members. Nor do we have to enter very much information at all. Also, once the participant has been added to the group, the group's charges are immediately calculated and updated.

To assign participants, click on the Assignments tab. (figure 9.23) Notice that you can filter your group participants by assigned or unassigned. Also, when you click on someone in the participants list, all the available spaces that match that person's occupancy type appear. Select one and press the Assign button. To move a participant's assignment, do the same thing. *CHARLES* will move the person to the newly selected space.

If you don't have any spaces to move a person, you will have to un-assign someone before that person can be moved. So, if you are trying to swap spaces between two participants, un-assign them both, then reassign them. To un-assign a participant, simply double-click their line in the participants list and follow the prompts.

The only other function that appears in the Short Term Summer Housing working window is the Correspondence tab - but we've seen that before so we won't rehash it here.

Summer 2008 ▼

Group Information   Participants   **Assignments**   Correspondence

**Group Participants ST1**  
Jax, Jimmy R  
From: 05/19/08 To: 06/04/08  
Roommate: nobody Room: BA-0006A-1  
Jaq, Jimmy S  
From: 05/19/08 To: 06/04/08  
Roommate: nobody too Room: BA-0007A-1  
Jix, Jamie R  
From: 05/19/08 To: 06/04/08  
Roommate: no one as well Room:  
Total Listed: 3

Select All  
Assigned  
Unassigned

Assign

**Available Spaces**

Bedspace	Room Type	Gender
BA-0004A-2	Shared	
BA-0004A-1	Shared	

  
Spaces Listed: 2

Figure 9.23 - Assigning short term participants.

## 10. Residence Life

In the last chapter, we saw a couple of functions in the Residence Life section, specifically those pertaining to summer housing. This chapter will deal with some of the other things we expect to happen at the residence hall level of your institution.

The Residence Life functions we didn't talk about in chapter 9 deal mostly with academic year housing functions. There is an Inquiry function, a function to check residents in and out of their spaces, a Lottery Results function and a Work Orders function to help with reporting and tracking of resident complaints about their room condition.

### 10.1 ACADEMIC YEAR INQUIRY

Figure 10.1 shows the Academic Year Inquiry function's working window. It is very much the same as the one we saw earlier when we discussed Academic Year Housing. It exists here so that if your hall managers don't have access to Academic Year functions, they will still be able to get some information about their residents; things that are important to the governing of the hall like emergency contact information or new assignment information or the entry of correspondence or notes to attach to the resident's record.

**Acad Yr Student Inquiry** Life for Education Systems

Fall 2008 A14-22-3313 Dervish, Whirling F. Male OCT 15 1989 0612 wfd1  
Student ID Name Gender Birthdate Grad Date NetID

Find Student  
Find Suite

Addresses Cell Phone: (413) 555-3224

Type	Street	City	State	Zip	Country	Phone
P	523 Park Place Apartment #34	Spin City	MA	01100-2323	US	(413) 555-3223
M	P.O. Box 332 Friendly Station	Spin City	MA	11200-	US	
C	8822 Main Lane	Bolton	MA	01100	US	(413) 555-3323

Current Status History Correspondence Room History Miscellaneous

UG-F UG UA 1 1 CU  
Student Type School College Year Group/Rate Confirmation Number

Emergency Contact Dervish, Mother MO (413) 555-3223 (413) 555-3111  
Person to Contact Relation Phone Number Alternate Phone

Contract Standard Contract ACTIVE AUG 24 2008 8 1 7  
Contract Type Contract Status Contract Date Semesters Used Remaining

Assignment CU-0002A-1 TR AS JAN 14 2009 AUG 24 2008  
Bedspace Suite Type Phone Number Assignment Status Assignment Date Effective Date

Roommates Cushing - suite: CU-0002

Name	Student ID	Room	Effective Date
Dervish, Whirling F.	A14-22-3313	CU-0002A-1	AUG 24 2008
Pierce, Harry A.	N44-11-1111	CU-0002A-2	AUG 24 2008

OK  
Cancel

Figure 10.1 -Academic Year Student Inquiry as it appears in Residence Life.

There are some other differences in this window. The number of tabs that appear in the lower section is reduced. Residence Life staff members may, or may not have access to student charge generation functions. If they do, they will be able to query information about charges and payments. But, as you

can see from figure 10.1, our currently logged in user doesn't have that access and the Charges/ Payments tab is missing. You could also limit access to Correspondence for Residence Life - but we'll discuss more about access issues when we discuss Systems Administration functions.

## 10.2 CHECK IN & CHECK OUT

Check In & Check Out functions allow your hall staff to log when a student checks in and print out a signature sheet to file. It also allows you to keep a record of when you issue which keys for the resident's space. Figure 10.2 shows the function's working window.

Hall	Bedspace	Suite Key	Room Key	Mailbox	Check In Date	Check Out Date	Status	Active
Cushing	CU-0002A-1	CU2-1122	2A-123	2A-1			AS	YES

Figure 10.2 - Check in & Check Out.

We can see all the assignments this resident has had for the Fall 2008 term on this working window. You'll notice from figure 10.2 that the resident we looked up has only one assignment in the term and has yet not checked in. If this person were to come to our hall desk and want to check in, we would find them using this function (as above) and click on their assignment in the Assignments list. *CHARLES* looks at the assignment we've selected in the list to determine what we might want to do. If the student has not yet checked in, *CHARLES* presents us with a Check in button.

There is also a check box above the Assignments list which allows us to print a signature card when we process this check in. Pressing the Check In button prompts you for the check in date (figure 10.3). Check in dates default to the current date (which makes sense if you are handling things in real time) but you can enter any date that exists in the date range of the term.

Once you say OK to the date prompt, your resident is checked in. If you selected the signature card check box, *CHARLES* would send a report to your printer that looks like the one in figure 10.4.

Check In & Check Out

Life for Education Systems

Fall 2008

Find

A14-22-3313 Student ID

Dervish, Whirling F. Name

wfd1 Net ID

M Gender

☒ Print Signature Card when Processed

Check IN

Assignments (assignments are sorted in reverse chronological order - newest first)

Hall	Bedspace	Suite Key	Room Key	Mailbox	Check In Date	Check Out Date	Status	Active
Cushing	CU-0002A-1	CU2-1122	2A-123	2A-1			AS	YES

Check In Date:

082508

OK

Cancel

Ok

Cancel

Figure 10.3 - A prompt asking for the Check In date.

JAN 23 2009

Best Music College

Check In Notice for:

<u>Student ID</u>	<u>Name</u>	<u>Hall</u>	<u>Bed Space</u>	<u>Effective Date</u>
A14-22-3313	Whirling Dervish	Cushing	CU-0002A-1	08/24/2008

Suite Key Issued: CU2-1122

Room Key Issued: 2A-123

Mailbox Issued: 2A-1

Checked In: 08/25/2008

Signature: \_\_\_\_\_

Figure 10.4 - A check in notice (signature card) that prints according to your instructions.



At the end of the semester, perhaps after fall finals, your student may (again) appear at your hall desk to tell you they are going home for the winter break. Depending upon your institution's requirements, you may want to check the student out of their assignment. If you go to the Check In & Check Out function and find the student, you would handle the check out in exactly the same way as the check in. Click on the assignment in the Assignment list (*CHARLES* sees that the student is already checked in so it offers you a Check Out button), press the Check Out button, fill in the date (if it isn't the correct date) and you're done. (figure 10.5)

You can also print out a signature card again if you want a hard-copy record that keys have been returned or simply for completeness.

**Check In & Check Out** Life for Education Systems

Fall 2008 Find

A14-22-3313 Student ID    Dervish, Whirling F. Name    wfd1 Net ID    M Gender

☐ Print Signature Card when Processed

Assignments (assignments are sorted in reverse chronological order - newest first)								
Hall	Bedspace	Suite Key	Room Key	Mailbox	Check In Date	Check Out Date	Status	Active
Cushing	CU-0002A-1	CU2-1122	2A-123	2A-1	AUG 25 2008		AS	YES

Check Out Date:

121508

Figure 10.5 - Checking a resident out at the end of the term.

### 10.3 WORK ORDERS

One of the biggest problems in running a residence hall is making sure that your facilities are in top shape. This takes a lot of work, communication and coordination with your facilities staff to make your student's experience in your residence hall a good and comfortable one. To that end, we have included a method of tracking work orders.

In *CHARLES*, a work order is generated on the web, submitted into the database and routed to the appropriate department or division by the residence hall manager. When you look at the Work Orders function's working window, you can see what we mean (figure 10.6).

Figure 10.6 shows you the work orders that have been newly added for a residence hall named Cushing for the Fall 2008 term. As you can see, it displays a description of the problem, as well as a specific problem category.

Id	Active Date	P-ID	Room	Name	Contact
13	FEB 4 2009	2	CU-0002A	hap22	harry@pierce.com

☒ Active
 ☐ Pending
 ☐ Closed

Problem Description:

Description

It has gone out and now I can only do my homework during the day. We have a no candle policy in Cushing so please handle this right away.

Thank you.

Figure 10.6 - Handling student requests for work to be done.

In the above instance, our student in room CU-0002A is having problems with his lighting. In order to initiate a response, your hall manager should set the status of this work order to pending. To do that, simply click on the work order in the Requests List, then press the Pending button. You will be prompted to compose a message to the student indicating that the request has been received and is being looked into. (figure 10.7) You may decide to send the message to either the student's internal email address or to the contact method he/she indicated. In our example, it is another email address but it could also be a phone number if that's how the student wants to be contacted.

Once you approve the text, press the OK button.

☒ Use NetID email: 
☐ Use contact method:

Your work order request has been processed by your building manager and forwarded to Facilities. If necessary, you may be contacted by Facilities before the work is done. You will receive a final email when the request is closed by the manager.

Please do not reply to this email address.

Your submitted problem is as follows:

It has gone out and now I can only do my homework during the day. We have a no candle policy in Cushing so please handle this.

Figure 10.7 - Composing a follow-up message to the work order.

When you press the OK button, *CHARLES* will generate a report to forward to the hall's facilities manager, as well as send the message to the student's contact. If the contact is the student's email address, it will be sent there. A sample of the report for facilities can be seen in figure 10.8.

<b>Best Music College</b>		Sequence Number:	13
		Cushing	
hap22 Submitted by	FEB 4 2009 Received	harry@pierce.com Preferred Contact	CU-0002A Room Number

**Description - Overhead Lightbulb**  
It has gone out and now I can only do my homework during the day. We have a no candle policy in Cushing so please handle this right away.

Thank you.

Action Taken:

---

---

---

---

☐ Defer cost of repair because... \_\_\_\_\_  
☐ Charge student for damages \_\_\_\_\_

Materials \_\_\_\_\_  
 Labor \_\_\_\_\_  
**Total Cost** \_\_\_\_\_

Figure 10.8 - Work Order Tracking form for facilities and maintenance.

Once you have informed the student and your maintenance professionals, the status of the work order changes to Pending and the working window is redrawn.

Note on the work order working window that the Requests List has three tabs across its bottom. The work order goes from one status to the next along its completion, from new to pending to closed. These three tabs are how you can track the work order.

Figure 10.9 shows that the work order in its pending state. Once we hear back from the maintenance personnel that the work has been completed, we would go to the pending tab to close the work order and let the student know (although they probably already do) that we've closed the work order.

The screenshot shows the 'Work Orders' application window. At the top, there's a header bar with 'Work Orders' on the left and 'Life for Education Systems' on the right. Below the header, there's a 'Requests' section with a table. The table has columns: Id, Active Date, P-ID, Room, Name, and Contact. The first row is highlighted in blue and contains the following data: Id: 13, Active Date: FEB 24 2009, P-ID: 2, Room: CU-0002A, Name: hap22, Contact: harry@pierce.com. Below the table, there are three tabs: 'Active', 'Pending', and 'Closed'. The 'Pending' tab is selected. To the right of the tabs, there's a 'Problem Category' dropdown menu set to 'Overhead Lightbulb' and an 'Edit Categories' button. Below the tabs, there's a 'Description' text area containing the text: 'It has gone out and now I can only do my homework during the day. There is a no candle policy in Cushing so please handle this right away. Thank you.' At the bottom of the window, there are five buttons: 'Refresh', 'Edit', 'Delete', 'Active', and 'Close'.

Id	Active Date	P-ID	Room	Name	Contact
13	FEB 24 2009	2	CU-0002A	hap22	harry@pierce.com

Problem Category: Overhead Lightbulb

Description: It has gone out and now I can only do my homework during the day. There is a no candle policy in Cushing so please handle this right away. Thank you.

Figure 10.9 - Selecting the Pending tab allows you to close work orders that have been completed.

As before, to close a work order, select the work order from the requests list and press the close button. You are again prompted to formulate the message you'd like to send to the student (figure 10.10)

There are a couple of other points we should raise about handling work orders. One is that you have the ability to reset the status of a work order. If a student has not received a response from you, you may want to set the status back to active and begin the process anew. Or if there is a work order that has been

The screenshot shows a dialog box for formulating a response for a completed order. It has two radio buttons at the top: 'Use NetID email:' (selected) and 'Use contact method:'. The 'Use NetID email:' option has a text field containing 'hap22@bestcollege.edu'. The 'Use contact method:' option has a text field containing 'harry@pierce.com'. Below the radio buttons, there's a text area containing the following text: 'Your work order request has been closed by your building manager, who you can feel free to contact about any problems or concerns. Unless otherwise noted, the work was completed successfully. The cost was billed as follows: --No additional costs have been incurred.-- Please do not include this email address.' At the bottom of the dialog box, there are three buttons: 'Change Text', 'Cancel', and 'OK'.

Use NetID email: hap22@bestcollege.edu

Use contact method: harry@pierce.com

Your work order request has been closed by your building manager, who you can feel free to contact about any problems or concerns. Unless otherwise noted, the work was completed successfully.

The cost was billed as follows:

--No additional costs have been incurred.--

Please do not include this email address

Figure 10.10 - Formulating a response for a completed order.

closed inappropriately (maybe the work done was substandard or incomplete), you could set the work order back to pending. You could even take a closed work order from closed to pending and then back to active - effectively reversing anything that has happened to the work order and start the process again. (This could also be used as a method to correct mistakes in data entry.)

You are also able to view all of the completed work orders on a hall by hall basis for the term by choosing the closed tab. This may give your institution an indication of where they might want to direct resources for future residence hall enhancements.

# 11. The Web Portal

As we've mentioned several times throughout this document, *CHARLES* has some web functions that assist your institution in the collection of data and the transmission of information using the internet. In order to fit in well with modern data collection techniques, we have designed a user configurable web portal scheme for your institution to use.

Figure 11.1 shows you the Function List as it pertains to the Web Portal. As you can see, it is divided into three distinct areas: Web page access, Web page generation and Web page headers.

First, a brief discussion about the theory of the portal and and some information about the structure.



Figure 11.1 - The Web Portal Functions

A student at your institution has a number of attributes which define them: a student type or a college they belong to; a gender, a class year, a housing type or some other preferences. All of these things are part of what defines a student in *CHARLES* and at your institution. Utilizing these pieces of information, we are able to define groups of students with certain similarities to allow them access to particular web pages to view and submit information to your system. The Web Portal allows you to define these groups and provide them access to a specific set of web pages. This is how you control which segments of your residents get access to which web pages.

The Web Page Generation function allows you to insert, edit and modify web pages. As we will see, it actually generates web pages with embedded data elements to populate your database (cgi style pages), complete with error checking. Although it does all of this, there are some limitations in the design of *CHARLES* generated web pages. These limitations can be overridden, however, if you are proficient in web design or html. More on this in section 11.2.

The Web Page Headers function gives your institution the ability to brand your web pages with your institution's personality. It is basically a way to control some of the look and feel of the web pages you give your student's access to.

## 11.1 WEB PAGE HEADERS

In the *CHARLES* portal, all web pages must have a header. So we'll begin our discussion of the portal functions from the bottom up. Figure 11.2 shows you the working window for the Web Page Headers function. As you can see, this is a simple table containing a name for your header and some HTML text. The text is simple HTML and begins with the `<html>` tag. It can include either a reference to your institutions css styles, or have the style included in the header.

As you may have guessed, some familiarity with html would be required to devise a header for you to use. As it is not the purpose of this manual to provide html training, we will basically leave it at that. Suffice it to say that if you have a header that you believe is appropriate, you can paste the html text into this field.



Figure 11.2 - Web Page Header for our work order..

A couple of things you should know: one is that the header should begin with the `<html>` tag. The header should end with the `</head>` tag. *CHARLES* will automatically end your web pages with the html close `</html>` tag so you shouldn't have to worry about that.

This working window behaves as you'd expect. To find a header, press the find button and enter the header name. To Insert, press the insert button and name your header, then past in (or write in) your html text. To edit a header, first find a header, then press the edit button and enter your changes. To delete a header, first find the header record, then press the delete button and follow the prompts.

You may also notice that there is are next and prior buttons. Pressing these will fetch the next or prior header record in alphabetical order so you can use these to page from header to header (if you can't remember what you named something). We suggest that you name header records so that their names correspond to their purpose. Note that in figure 11.2, we have named this header `WorkOrderHeader`, so we will know which header record to use when we design our work order web page.

The only other button on this working window is the print header button. Pressing this will send the header to an html text file so you can open it in your web browser (or text editor) and see how it will look on the web. You will be prompted for a location on your computer for where to save the file. Figure 11.3 shows you this prompt.

Once your header has been created, it can be used on any of the web pages you create. It will appear in a list of headers on the Web Page Generation function's working window - as we will see in the next section.

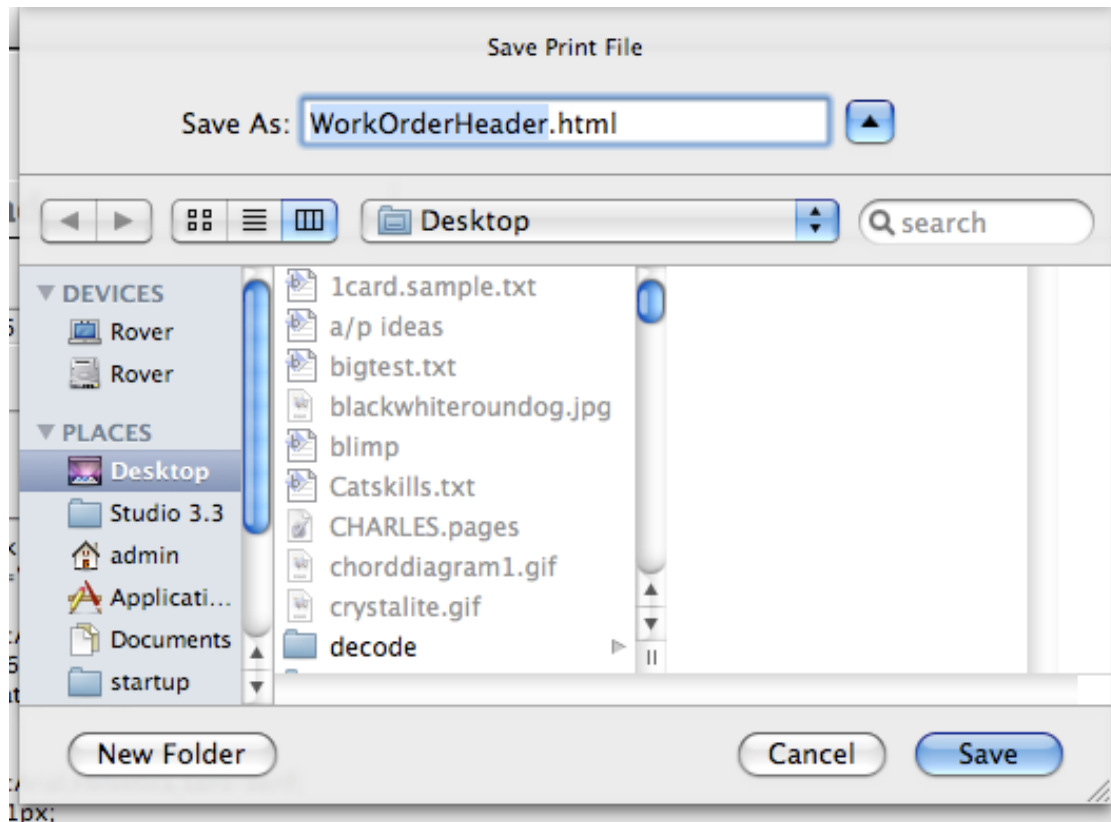


Figure 11.3 - Printing the header html to a file for browser viewing.

Once you have a copy of your header on the disk, go into your browser and select the Open File command. Then direct your browser to the file.

## 11.2 WEB PAGE GENERATION

Since it doesn't really make sense to discuss access to web pages when no web pages exist, we will turn our portal discussions to web page generation. Let's continue with our example that we touched on in the last chapter - the work order request form.

Figure 11.4 shows you the working window for the Web Page Generation function. As you can see, web pages in *CHARLES* are term specific. When you think about it, they would have to be since every semester brings changes in student attributes. If they were not term specific, it would be very difficult to determine what information needed to be included on student web forms. As you can see in figure 11.4, the default term of Fall 2008 is automatically selected when we choose the function.

There are many different sections on this working window. Some of them relate to the information we place on the form (the data we need to pre-populate the web form) and how to fetch that data from the database. Other sections relate to what happens with the information we collect and what it does when it comes back into *CHARLES*.



This webpage will not work until you enter some html.

**CHARLES<sup>©</sup>**  
College Housing Assignment & Residence  
Life for Education Systems

Web Page Generation

Form Name:  Header:  Term:

**Page Specifics**

☐ Page is active ☐ Override HTML

HTML

Next Page:  Button Text:

Confirm Message:

Form Link:

**Page Submission Behavior**

Tables on This Form			
Table Name	Do Nothing	Update Values	Insert Values
<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

☐ Effect Assignment Status?

☐ Send eMail Confirmation?

Page SQL ☐ Override SQL

Figure 11.4 - The Web Page Generation working window.

### 11.2.1 Inserting a new web page

In order to create a web page, press the insert button and begin filling out the available data elements. We will call our first web page the WorkOrderForm.

Note that the Header field is a drop list. It should contain the header we created in the last section. The data fields in the Page Specifics box pertain to web page behavior and can be initially left blank. Once you have finished entering the data, press the OK button to save your page. Figure 11.5 shows you a completed page.

You'll notice also that when you enter the web page initially, you may not have any html for the page. In fact, html is generated as you add data elements to your web page so that is fine. You will be warned that the page won't work until it has some html. This is to remind you to either edit the record and paste in some html (if you have already designed your page) or add data elements to it. If you do have some html and would like to enter it when you create the page, you can do so by clicking the Override HTML check box and pasting it (or writing it) in.

When you have created your initial web page, you are not yet finished. Figure 11.5 shows you the working window for your web page after the web page record has been created. Let's look at the sections on the window to help you understand the layout of the working window.

Web Page Generation

Form Name:  Header:  Term:

Embedded Data Elements

Order	Element Name	Table Name	Use in Fetch
(double-click line to remove element from form)			

Page SQL ☐ Override SQL

Page Specifics ☐ Page is active ☐ Override HTML

HTML

Next Page  Button Text

Confirm Message

Form Link

Page Submission Behavior

Tables on This Form			
Table Name	Do Nothing	Update Values	Insert Values

☐ Effect Assignment Status?

☐ Send eMail Confirmation?

Figure 11.5 - The new web page record has been created.

On the left side of the working window in figure 11.5, there are two sections, one above the other, showing you the embedded data elements and the page sql. These two sections are related in that as you enter data elements to to page, *CHARLES* builds an sql query designed to fetch the records you need out of the database so your web page will be populated with values when the student receives it through their web browser. As you add more and more data elements to the form, *CHARLES* generates a proper query, adding (or not) the data fields and tables to which they belong to the query. So, it could be said that the left side of the working window deals with the fetching of data from the database to send with the form your students receive.

The sections on the right side of the working window, Page Specifics and Page Submission Behavior, deal with the look and feel of the web page, how it behaves and what happens to the data when the student is ready to submit their information. We'll talk about all the elements included in these areas but it could be said that the right side of the working window deals primarily with what happens when the web page is returned to you.

In short, you should think about the left side of the working window as effecting the web page as it goes out to students, and the right side of the working window as effecting the web page as it comes back into your database.

There are a few other things we entered as we set up our initial web page record. In figure 11.5, you can see that we have designated a Next Page, Button Text, Confirm Message and Form Link. Let's explain each of these in turn.

The Next Page data element tells the web portal which page will come next once this page is submitted by a student. The means that upon successful receipt of the work order form we design here, students will receive a web page called “WorkOrderConfirm”. Since we have designated that this happens each time we receive a work order, we had better remember to design that page once this one is finished. If we had left this data element blank, *CHARLES* would expect that no additional page will be sent to the student and would, instead, return them to the general forms page. (We’ll deal more with the forms page in the next section.)

The Button Text data element is the text you want placed on the standard submit button of the web page. If you leave the field blank, the default is to place the word “Submit” on the button.

The Confirm Message data element allows you send a final message before the student actually sends the form back to us. If the field is left blank, a standard confirmation message is sent instead of something you may want to point out as a final reminder to the student before they submit their data. The standard confirmation message says “Are you ready to submit this form?”

The Form Link data element is the actual text that will appear on the student’s form page to allow them to access this form. We will see this in the next section.

### **11.2.2 Adding data elements to your web page**

Web forms designed in CHARLES and containing data elements from your database will always generate javascript error checking code. As we enter data elements, we’ll see how this occurs.

In order to add some data to the form, press the Add Data Element button. Figure 11.6 shows you the working window behavior as you do this. Notice that the Embedded Data Elements list on the left side of the working window changes to a Data Elements section. This is where you add the elements to the form.

CHARLES needs to know the table to which the data element belongs, then the actual data element name. As you select a table from the table drop list, CHARLES builds a list of the data fields that belong to that table. Figure 11.6 shows that we have selected the F\_studentdemo data table and the data field called s\_id or Student ID.

Notice the options we are presented with in this scenario. We have a check box to tell the system whether to use this table to fetch. This means that this table (and field) will be used in the sql query that populates the page as it goes out to the student. We can place the field on the form as an entry field (the default), a text area or as a hidden field that won’t be visible on the form at all.

We also have check boxes where we can set the display only attribute of the field or determine that the field be made mandatory We can also give the field a tab order.

In this case, since we wouldn’t want a student who submits a work order to edit or modify their student id, we should make this element display only and give it a tab order of 1 - it will be the first data element on the form. The tag for this field on the web page is shown in the Data Field list. When we are finished entering this field, we press Ok and the field is added to our web page. (figure 11.7).

**Web Page Generation** Life for Education Systems

Form Name:  Header:  Term:

**Data Elements**

☒ Use Table to Fetch

- Student ID

Data Field

☒ Entry Field ☐ Display Only

☐ Text Area ☐ Mandatory Field

☐ Hidden

Web Tab Order:

HTML:

```
<tr><td align="right"></td><td align="left"><input name="Select Field" type="text" tabindex="0" value="",pro.w.Select Field,""></td></tr>
```

Page SQL ☐ Override SQL

**Page Specifics** ☐ Page is active ☐ Override HTML

HTML

Next Page:  Button Text:

Confirm Message:

Form Link:

**Page Submission Behavior**

Table Name	Do Nothing	Update Values	Insert Values
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

☐ Effect Assignment Status?

☐ Send eMail Confirmation?

Figure 11.6 - Adding a data element to our web page.

**Web Page Generation** Life for Education Systems

Form Name:  Header:  Term:

**Embedded Data Elements**

Order	Element Name	Table Name	Use in Fetch
1	s_id	F_studentdemo	<input checked="" type="checkbox"/>

(double-click line to remove element from form)

Page SQL ☐ Override SQL

Select s\_id from F\_studentdemo where s\_id=s\_id

**Page Specifics** ☐ Page is active ☐ Override HTML

HTML

```
<script language="JavaScript">
var submittimes = 0;
function submittonce() {
submittimes++;
if (submittimes == 1) {
document.form.submit();
}
}
```

Next Page:  Button Text:

Confirm Message:

Form Link:

**Page Submission Behavior**

Table Name	Do Nothing	Update Values	Insert Values
F_studentdemo	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

☐ Effect Assignment Status?

☐ Send eMail Confirmation?

Figure 11.7 -Our web page now has one data element.

You may have noticed, as we entered the data element, that the HTML text box (figure 11.6) changed as you gave your data element some attributes. The html is dynamically generated as you add the element to the form. Another thing that occurs is the generation of some JavaScript as data elements are added. This is so *CHARLES* and your web page can handle the confirmation message, redirection and error checking you can add to the form. This JavaScript is also dynamically generated.

In figure 11.7, we can see our web page and some more information about its behavior now that we have a data element embedded. It has generated some Page SQL for us to look at, shows us the field (or fields) embedded, and has placed a list of tables in the Page Submission Behavior section of the working window.

Remember that we said that the information on the right side of the working window governs how the page is treated as it comes back into your web server. The Tables on This Form area is where you can designate how to treat data coming in - you can either Do Nothing to the table date, Update Values that belong to the table, or Insert Values into the table. In our instance in figure 11.7, we will choose to do nothing to this table since we are trying to generate a new work order. We are not interested in adding or updating student demographic information in this example.

As we continue to design our page, we add more and more data elements that we want the student to see when they receive their page. We may want to give them some information about where they currently live (in fact, we should do this because we'll need to know where to send our maintenance people to address the problem they will be submitting). We can show them their current hall and bed space assignment. Then we can begin to add fields we want them to fill out.

The screenshot shows the 'Web Page Generation' tool interface. The top bar includes 'Form Name: WorkOrderForm', 'Header: WorkOrderHeader', and 'Term: Fall 2008'. The 'Embedded Data Elements' table lists 8 elements with their names, table names, and 'Use in Fetch' status. The 'Page SQL' section shows a SQL query for selecting student and room information. The 'Page Submission Behavior' section includes a 'Tables on This Form' table with radio buttons for 'Do Nothing', 'Update Values', and 'Insert Values'. The 'Page Specifics' section contains HTML code for a submit button, a confirmation message, and a form link. The right sidebar has buttons for 'Insert', 'Edit', 'Delete', 'Print Web Page', 'Add Data Element', 'Copy WebPage', 'Next', 'Prior', 'Ok', and 'Cancel'.

Order	Element Name	Table Name	Use in Fetch
1	s_id	F_studentdemo	<input checked="" type="checkbox"/>
2	s_firstname	F_studentdemo	<input checked="" type="checkbox"/>
3	s_lastname	F_studentdemo	<input checked="" type="checkbox"/>
4	tbuilding	F_termrooms	<input checked="" type="checkbox"/>
5	troomnum	F_termrooms	<input checked="" type="checkbox"/>
6	woissueid	F_workorders	<input type="checkbox"/>
7	wodescription	F_workorders	<input type="checkbox"/>
8	wocontact	F_workorders	<input type="checkbox"/>

Table Name	Do Nothing	Update Values	Insert Values
F_studentdemo	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
F_termrooms	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
F_workorders	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Page SQL: ☐ Override SQL

```

Select
s_id,s_firstname,s_lastname,tbuilding,troomnum
from F_assignment,F_studentdemo,F_termrooms
where s_id=asnsid and asnterm='083' and
asnterm=1 and s_id=s_id and
troomidentity=asnterm and troomterm='083'

```

Page Submission Behavior: ☐ Effect Assignment Status?

Page Specifics: ☒ Page is active ☐ Override HTML

HTML: `<script language="JavaScript">
var submittimes = 0;
function submitonce() {
submittimes++;
if (submittimes == 1) {
document.form.submit();
}`

Next Page: WorkOrderConfirm Button Text: Submit Form

Confirm Message: Are you ready to submit this form?

Form Link: Residence Hall Work Order Form

Send eMail Confirmation? ☒ NetID email address ☐ Term email address ☐ Alternate email address

Figure 11.8 -Our complete web page to collect work order requests.

Figure 11.8 shows the completed web page for this work order form.

Let's look at the sections to try to understand what is going on here. On the left side in the Embedded Data Elements, we can see that we will fetch student data using the tables F\_studentdemo and F\_termrooms. We can see in the Page SQL that we are selecting the fields that are checked to be fetched in the above list. We have flagged the web page to be active, meaning we can assign it to a web access group (which we'll see in the next section). The page will have a submit button which will say "Submit Form" and will confirm submission with a message.

Also, the form will be accessed using a link from the student's forms page that says "Residence Hall Work Order Form". When the form is submitted, the next page the student will see is something called "WorkOrderConfirm". We have also decided to send an email confirmation to the student's NetID email address.

In the Page Submission Behavior box, we can see the tables included on the form and tell the page what to do with them. We have set the first two tables to do nothing, while that last table, F\_workorders, is directed to insert values. Inserting values is what will put the data from the form into our database.



## Work Order Request Form


Student ID	<input type="text" value="',prow.s_id,'"/>
First Name	<input type="text" value="',prow.s_firstname,'"/>
Last Name	<input type="text" value="',prow.s_lastname,'"/>
Hall Name	<input type="text" value="',prow.tbuidng,'"/>
Room Number	<input type="text" value="',prow.troomnum,'"/>
Issue	<input type="text" value="Select"/>
Description	<input type="text"/>
Contact Information	<input type="text" value="',prow.wocontact,'"/>

Submit Form

Figure 11.9 -Inspecting our form to see how it will look on the web.

We also have the ability to print out a copy of our form to a file for viewing in a browser. Figure 11.9 shows our completed form. Using this tool, we can see how our form will look before we begin serving it to our students.

You may notice in figure 11.9 that there are values in the fields of the form. These are representations of the values you decided to either send to the form or receive from the form. Of course, when the student receives this form, the values will be their values.



**Work Order  
Request Form**

---

Student ID

First Name

Last Name

Hall Name

Room Number

Issue 

✓ Select

Bugs in Tub

Overhead Lights

Slime in Toilet

Description

Contact Information

Figure 11.10 -Actual values for a fictitious student filling out our work order form..

### 11.3 WEB PAGE ACCESS

Of course, no one will be able to get to your web forms unless you set up a system of access allowing the proper selection of students from your population to access the proper web forms at the proper times during your housing cycle. This seems very proper, but it leads to a discussion of how the web portal allows for changes in student statuses and, thereby, the forms they have access to as they travel through your cycle.

Figure 11.11 shows you the Web Page Access working window. On it, we see the access record for a population called “General Assigned Population”.

**Web Page Access** Life for Education Systems

Population Selection: **General Assigned Population** Population Name: General Assigned Population Term: Fall 2008

**Population Definition** Describe

SQL Definition: Select \* from F\_assignment,F\_termstudent where term\_id=asnsid and asnterm='083' and term\_term='083' and asnactive=1 and asnstatus='AS' and term\_stutype='UC'

☐ Override SQL

Forms Page Header: FormsHeader

☐ Contract Check  
Check this box if this population requires validation of their current housing contract.

Web Pages: ☐ FreshmanApp ☒ WorkOrderForm

Pages Granted: WorkOrderForm

Insert Edit Delete Print Forms Page

Portal Download Forms: (links to pdf documents and web page instructions)

For further information, please visit <a href="https://www.bestcollege.edu/housing" target="\_blank">Best Housing</a>.

This window allows you to build a forms page for each population you define.  
Enter the population name, then modify the definition of your population.  
Choose the web pages you'd like to make available to the population.

OK Cancel

Figure 11.11 -Inspecting our form to see how it will look on the web.

As we’ve already discussed, *CHARLES* is a term based system and uses a system of assignment statuses to track your student’s progress through your housing cycle. Utilizing this fact, we’ve designed the web portal to leverage these statuses to allow a certain type of flow to the forms that are available to your students as these statuses change.

So looking at figure 11.11, we can see that the population has a name, a term, a definition (in SQL parlance), a header (just like a web form), a number of pages that can be assigned, and a section where you can provide links to information documents or other web pages.

Remember when we talked about web forms, we mentioned that the active flag (refer to figure 11.8) should be checked. That active flag is what determines which forms appear in the Web Pages list in figure 11.11.

There are some instances where you might have web forms that should not be set to active. Examples of this might be a confirmation page, or an error page. For more complete details on designing these types of web forms, please consult the Web Portal manual.

### 11.3.1 A Note About Web Portal Theory

Since *CHARLES* handles student applications through a series of assignment statuses, it is best if you think of the portal, not so much as distributing forms to populations - but as providing access to forms when people become part of the population to which the web form belongs.



For instance, when a student is first entered into the system, they are given an assignment status of “IN” or inactive. In order to provide inactive students access to forms, you would want to create a web access population for students with this assignment status. These web access populations can be further defined to include only undergraduate students or freshman students or any other definition that makes sense to you.

You may also want to provide people who are assigned to rooms to have access to forms like our work order form, or a status page, or a web form to change their room or meal plan assignment. There are endless possibilities.

Just remember that when a student’s assignment status changes, so does their access to forms as they move from one population definition to another.

### 11.3.2 Web Page Access Creates the Forms Page

While you are developing your web access structure, you should begin to think about the web access functions as though you were designing an introductory web page for your students to see when they enter your portal. In fact, that is what you are doing. As you create a web access population, you are actually designing a web page with links to the forms the population will be given access to.

On the working window in figure 11.11, you can see the explanation stating that this window allows you to build a forms page for each population. You can also see that you can print the forms page as you design it so you can keep track of how it will look to your population when they see it.

But rather than just talking about it, let’s add a population definition and see what’s involved.

As other entries, press the insert button to add a population. The fields will become active and you can begin the process by providing a name for the population, determining the term, which header to use from the drop-list, checking off the web forms that will be accessed, and providing links to other information (or simply entering text for your population to see).

Once you have finished, press the OK button to save the definition. When you do that, you will be prompted that the population has no definition and you will need to add one (figure 11.12).



Figure 11.12 -A reminder that the population still needs a description..

Once you press OK on this prompt, your population has been created. You can then (even without the description) take a look to see how the forms page for this population will look by pressing the “Print

Forms Page button. Figures 11.13 and 11.14 show the completed entry and the forms page as viewed in your browser.

**Web Page Access** Life for Education Systems

Population Selection: **General Assigned Population** Population Name: General Assigned Population Term: Fall 2008

**Population Definition** Describe

SQL Definition: ☐ Override SQL

Forms Page Header: FormsHeader

☐ Contract Check  
Check this box if this population requires validation of their current housing contract.

Web Pages: ☐ FreshmanApp ☐ WorkOrderForm

Pages Granted: WorkOrderForm

Portal Download Forms: (links to pdf documents and web page instructions)

For further information, please visit [Best Housing](https://www.bestcollege.edu/housing).

This window allows you to build a forms page for each population you define. Enter the population name, then modify the definition of your population. Choose the web pages you'd like to make available to the population.

Insert Edit Delete Print Forms Page OK Cancel

Figure 11.13 -A completed entry with no definition.

**RDS CO.**

**Housing Forms and Information**

**Available Forms for Firstname Lastname**

- Residence Hall Work Order Form

**Download Forms and Instructions**

For further information, please visit [Best Housing](#).

Figure 11.14 -Use your browser to view the Forms Page you just created.

All of the forms pages you create will be laid out in a similar fashion to the one you see in figure 11.14. The header will be the one you chose. It will be followed by a section containing an ordered list of the forms you've made available to the population (in this case, we have only one). Following that will be another section containing any download forms and instructions (or links and explanations).

We've decided to make all the forms pages look the same so all of our student populations will have a similar experience. You could create different headers to better identify the populations if you like - but we leave that to you.

Remember that we left this population without a description. In order to define the people who will see this forms page, we must enter the description of who they are.

When we press the Describe button, the Population Definition box in the upper section of the working window changes (figure 11.15). We are then able to select data elements to help us fully describe the group we want to give access to. In our case, we have selected the assignment status field to be equal to “AS” or assigned. That may (or may not) fully define your population, but if it doesn’t you can add more data elements to your definition by pressing the “more” button. If, however, you have finished with your definition, press the “done” button.

Web Page Access Life for Education System

Population Selection: General Assigned Population Population Name: General Assigned Population Term: Fall 2008

Population Definition

Data Element: ☐ Active Status ☒ Assign Status ☐ Assigned Room ☐ Assignment Term

equal to Value: AS

Describe more... done

SQL Definition: ☐ Override SQL

Figure 11.15 -Allowing the entry of data elements and values to define the population..

As you add more and more data elements to your definition, the SQL Definition field fills up with what you selected. While this is a simple representation of the fields and values you select while you are processing your definition, you’ll see that when you press the “done” button, the text will change to an executable SQL statement and saved to your population. (figure 11.16 and 11.17)

Web Page Access Life for Education System

Population Selection: General Assigned Population Population Name: General Assigned Population Term: Fall 2008

Population Definition

Data Element: ☒ Active Status ☐ Assign Status ☐ Assigned Room ☐ Assignment Term

equal to Value: 1

Describe more... done

SQL Definition: asnstatus='AS',term\_stutype='UG',term\_stutype='GD'

☐ Override SQL

Figure 11.16 -A simple comma delimited list of fields and values as we select data elements and values.

Population Definition

Describe

SQL Definition: Select \* from F\_assignment,F\_termstudent where term\_id=asnsid and asnterm='083' and term\_term='083' and (term\_stutype='UG' or term\_stutype='GD') and asnactive=1 and asnstatus='AS'

☐ Override SQL

Figure 11.17 -The comma separated list becomes an actual executable SQL statement.

At this point, our population is completely defined. When a student matching the SQL definition accesses our portal, they will be given the appropriate forms page.

Notice that there is an Override SQL check box on this working window. If you are proficient with generating SQL queries, you could define your own selection criteria without going through the Describe button - more button - done button scenario we outlined above. To do this, choose your

population, press the edit button and check the Override SQL check box to begin entering your SQL query.

### **11.3.3 Cautions for Forms Page Generation**

Since the Web Page Access functions allow you to define which populations have access to the forms you create, you should be extremely careful to try to make the populations you define as mutually exclusive as possible. Perhaps this goes without saying but we feel it is very important and could cause a lot of student confusion. If, for example, an assigned, returning student is presented with the web form for completing a freshman application, it would be a bit misleading and your office would probably get calls about it.

A good rule of thumb is to try to keep the Access Populations from “stepping on each other”. A good way to do this is to utilize the different assignment statuses available. You should plan for your cycle and define populations to handle each eventuality as students pass from one assignment status to the next.

Create a table of student types and assignment status for each step along the cycle. This will help you determine which types of populations to design web form for.

In the appendix to this manual, we include an explanation of the various assignment statuses to assist you in this process.

## 12. Systems Administration

The Systems Administration high level function is where you control most of the setup of your version of *CHARLES*. It contains access to all the tables included in the system. It allows you to define user groups and assign individuals to these groups. It gives you access to definitions to help make your version of *CHARLES* as personal to your institution as possible. Figure 12.1 shows the expanded Systems Administration high level function.

*CHARLES* has tables to help control everything you do in your housing cycle. Some of the table are term specific and are listed under the Term Tables higher level function. Others are not term specific and are listed under System Tables. Other functions are not tables at all but are functions you may wish to access to help administer the system. These fall under the global Systems Administration title.

Lets begin looking at these according to their grouping.

### 12.1 TERM TABLES

Term tables in *CHARLES* are, as you may have guessed, term specific. They require a term to be created in order to make them work, and must be updated each term to continue working.

#### 12.1.1 Setting Up a Term

Since the term must exist, it is probably best to begin this discussion with the creation of a term using the Term Set-Up function. Figure 12.2 shows the Term Set-Up working window for the default term we have been using in this manual (Fall 2008).

Terms have a number of parameters that must be set. You'll notice in figure 12.2 that we need a term code (we used 083 but you can decide upon another nomenclature). The term name is a combination of the semester you choose from the drop-list and the year (another drop-list).

The weeks belonging to the term must begin on a particular day. If your term charges weekly, this would be the check-in day for your system. If your system charges daily, then this value is not taken into account (although it is good to have a dat specified).

The term must begin on a particular day (usually your semester check-in day) and end on a particular day (usually the final check-out day). The date classes begin must also be specified. The

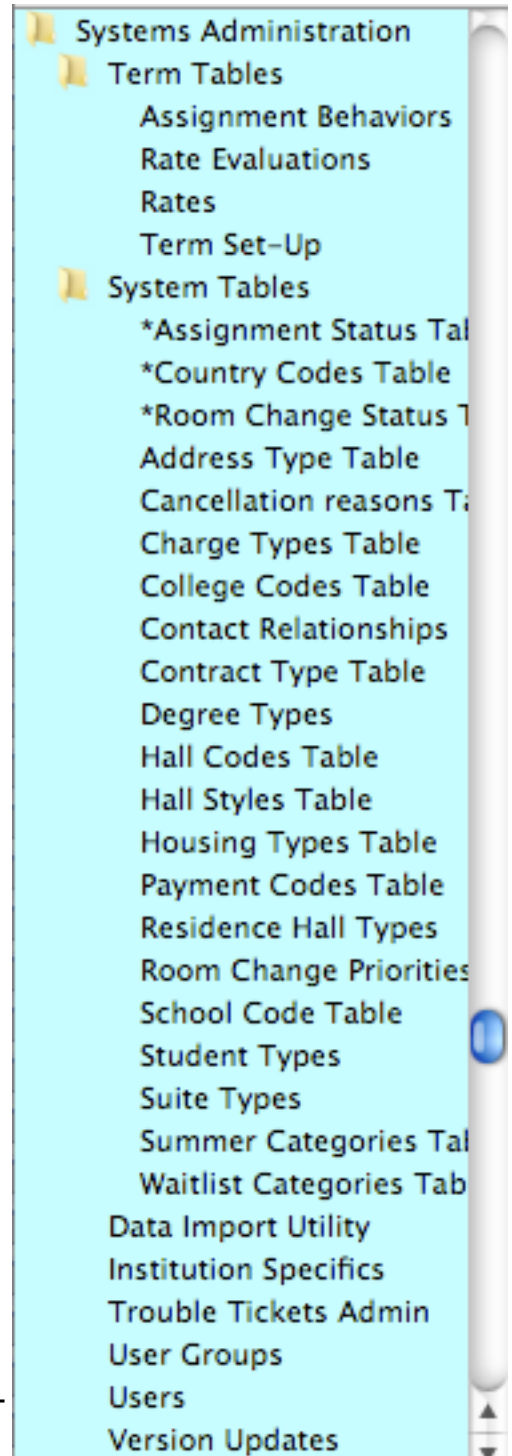


Figure 12.1 -System Administration Functions

number of days in the term is determined by subtraction of the start of classes and the term end. You

must also specify the cancellation deadline date (the date students must cancel by without penalty)

Figure 12.2 -The Term Set-Up working window.

and the date that current contracts end. Contracts are usually set up to run for the entire academic year so a fall contract end date would actually be the ending of the spring term. The spring term's contract ending date would be the same.

You must also indicate whether the term is to be charged by week or by day.

There are a few other things you should note in this window. There are two buttons in the lower right section called Term Rollover and Close Term. The Term Rollover button is used when you want to roll a fall term into a new spring term. For example, at the beginning of processing for spring, you may want to "roll" all students and assignments from their current fall spaces to their spring spaces. This is called Rollover.

Rollover is quite a complicated procedure (usually) but *CHARLES* makes it fairly easy. When you press the Term Rollover button, you are prompted to verify that this is what you want to do (figure 12.3). Then a window appears prompting you to enter the source term and the target term. Once this is done, *CHARLES* takes over by moving all of your currently assigned students into the target term, moving

Figure 12.3 -The prompt to continue with the rollover.

the room (space) file into the new term, assigning all the people to their spaces, flagging all the

genders for rooms, and setting all the contracts to indicate that another semester of use has occurred. The only thing it doesn't do is charge everyone (but there's a way to do that globally too).Figure 12.4 shows you a sample of the Term Rollover window and it's explanation. Of course, the new spring term must exist in your system before CHARLES will allow you to select it.

The other button on the bottom right side of the Term Set-Up working window is called Close Term. Closing a term is pretty much what you'd expect. Basically, closing a term involves checking contracts to see if any of them should be expired due to maximum use of the number of semesters allowed. It should be noted that closing a term will not end charges being generated for that term. You would have to uncheck the Charges On checkbox to do that. Fall terms should be closed around the same time that

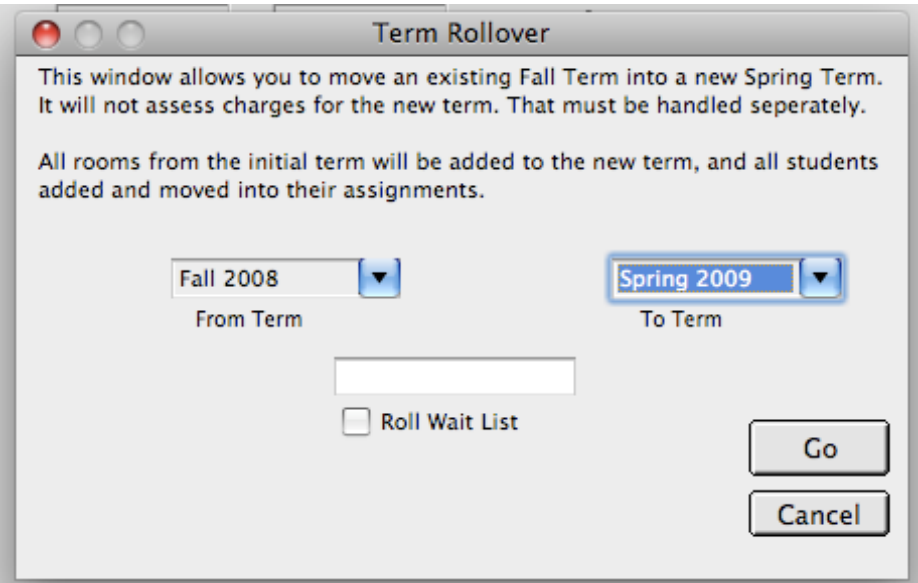


Figure 12.4 Term Rollover Function Window

spring terms begin. Spring terms should be closed a few days after final check out.

12.1.2 Rate Evaluations

The rate evaluation working window appears in figure 12.5. Rate evaluations and the rate table go

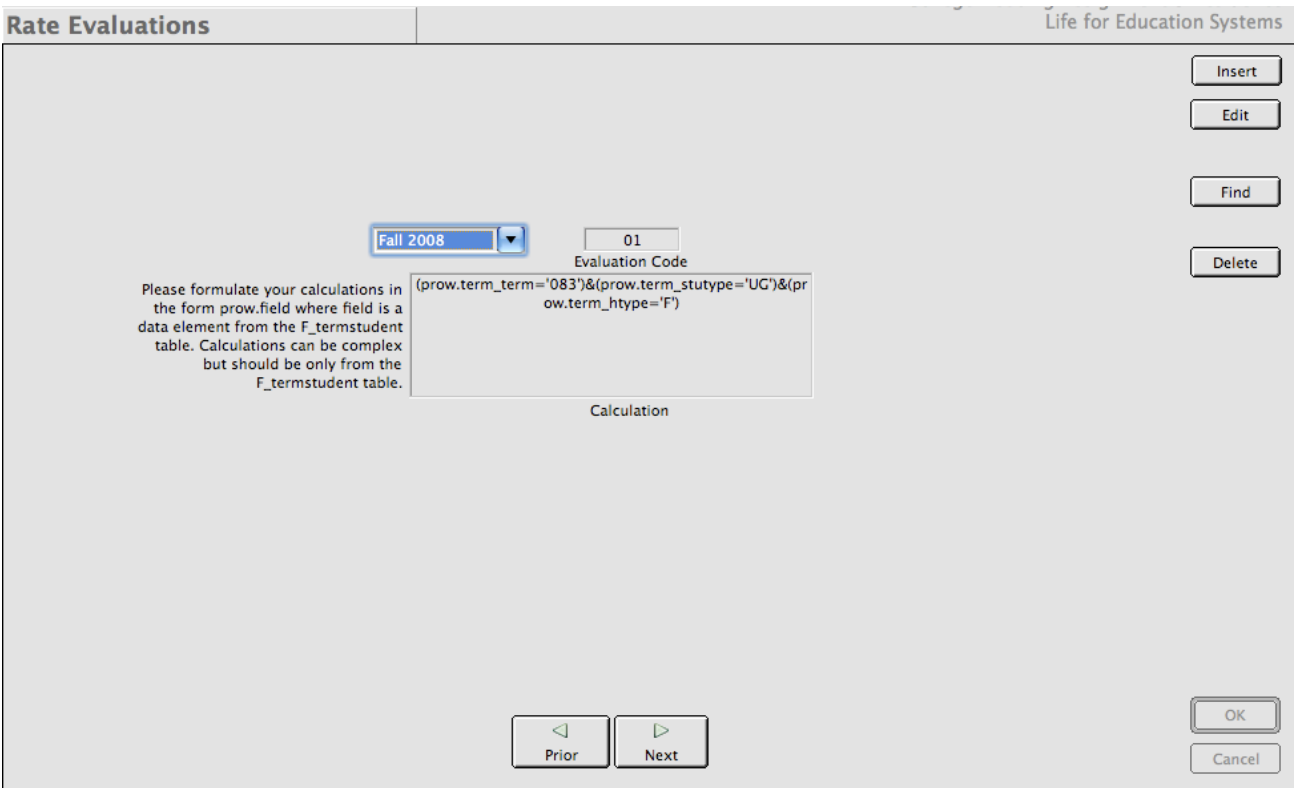


Figure 12.5 The Rate Evaluation Working Window



hand in hand in that the rate evaluations you enter in this table determine which of the possible 15 rates will be used when a charge is assessed. You can see from figure 12.5 that we have defined a rate of “01” for a calculation that finds all Undergraduate Freshman students in term “083”. If an application is entered and this calculation is met, the application would be given a rate number of “01” and that would be the rate used no matter what space that person was assigned to.

A nice feature of *CHARLES* is that you can determine different rates for different types of students based upon student term data. If you want to charge a higher or lower amount for incoming students or for a particular college or major code, you are able to do that. This allows for a lot of flexibility in defining your rate structure.

Rate evaluation codes must be entered as text for ascii ordering purposes (sorting by evaluation code). For example, begin with “01”, then “02”, etc. all the way to “15” (if you will have that many rates).

As we just mentioned, we can have up to 15 different charge rates per charge code and we’ll talk about charge codes now.

### 12.1.3 Rates

First let’s take a look at the Rates working window (figure 12.6).

The screenshot shows the 'Rates' window with the following elements:

- Title Bar:** 'Rates' on the left, 'Life for Education Systems' on the right.
- Top Section:** A dropdown menu showing 'Fall 2008', a 'Rate Type' dropdown set to 'Room', and an 'Insert' button.
- Form Fields:** 'Rate Code', 'Account', 'Alt Account Number', and 'Description' input fields, with a 'Find' button to the right.
- Rate List:** A vertical list of 15 rates, each with an input field: Rate 1, Rate 2, Rate 3, Rate 4, Rate 5, Rate 6, Rate 7, Rate 8, Rate 9, Rate 10, Rate 11, Rate 12, Rate 13, Rate 14, Rate 15.
- Buttons:** 'Edit', 'Next', 'Prior', 'Delete', 'Roll', 'OK', and 'Cancel' buttons are located on the right side of the window.
- Instructions:** Text on the left states: 'Your system provides 2 types of rates to enter: Room Rates and Estimates. Room Rates must be entered for each hall and key off of charge types from the charge type table. Estimates key off of student housing types.'

Figure 12.6 -The Rates working window.

As you can see, there are 15 different rates that can be entered for a single rate code. Carrying over our discussion in section 12.1.2 then, rate evaluations that are carried out on your entered students are attached to their term records. This way, CHARLES knows, by looking at the student’s charge



rate, which of the 15 rates to use when a charge is calculated.

Notice on figure 12.6 that there are some explanatory notes in the left margin. This tells us which types of rates may be entered - in this case there are two because our system has been set up to use either estimated rates or room rates. The working window also explains that room rates key off each hall and charge type while estimates key off of a student's housing type.

*CHARLES* has the ability to use estimated rates when students are ready to be assigned but don't yet have an assignment. Say it is time for your bursar (or yourself) to generate charges for your first fall billing. This may come in June or July - well before you have completed all of your assignments for September. If there are active, unassigned students who you would like to bill for housing, you would need to formulate an estimated charge in lieu of an actual charge based on a physical space. This would allow you to bill all your students at the same time and give you the extra time needed to place students into spaces before fall check in. When an actual bed space assignment is made, the estimate would be adjusted in favor of the student's actual charge.

As with other functions, entering a rate is straightforward. Press the insert button, tab from field to field and provide the information you need.

In the example in figure 12.7, we have entered a rate record for a one-bedroom in a residence hall named "Halsey" for the Fall 2008 academic term.

**Rates** Life for Education Systems

Fall 2008 Rate Type Room Halsey

1B - One Bedroom

Rate Code Hal-1B Account 44332-27

Alt Account Number

Description Halsey - 1 Bedroom

Your system provides 2 types of rates to enter: Room Rates and Estimates. Room Rates must be entered for each hall and key off of charge types from the charge type table. Estimates key off of student housing types.

Rate 1	3300.00
Rate 2	3500.00
Rate 3	1750.00
Rate 4	
Rate 5	
Rate 6	
Rate 7	
Rate 8	
Rate 9	
Rate 10	
Rate 11	
Rate 12	
Rate 13	
Rate 14	
Rate 15	

Insert Find Edit Next Prior Delete Roll OK Cancel

Figure 12.7 -The rate for a one-bedroom in Halsey Residence.

In looking at figure 12.7, we notice that we are only using the first three rates. This is because we have only issued three different rate evaluations and they will evaluate to rate 1, 2 or 3. We have also

entered and account number for passing to your bursar or ledger system. The Description (“Halsey - 1 Bedroom” is what will print out as a description on the student’s bill or as a display when looking at a student’s record in the Inquiry functions.

On the right side of the working window in figure 12.7, we see the standard pushbuttons for accessing the functions. But notice the button called “Roll”. This button is how you can roll an entire semesters-worth of rates into another semester. Say you have the same rates for spring as you had in the fall semester (a usual case). You wouldn’t want to have to re-enter all the rates and codes again if you didn’t really have to. This button allows *CHARLES* to create a whole new semesters-worth of rates without all the entry. Pressing the Roll button uses the current term or semester (the one currently selected in the working window) and prompts you for a term to roll the rates into. So, theoretically, you can clone the rates of any term and pass them to another.

This can even be done when starting a new fall term. If you are using the same ledger codes, halls, bed space identifiers (or charge types), you could roll last fall’s rates into this fall. Of course, if the prices are changing, you would have to edit the rates once they were created.

#### 12.1.4 Assignment Behaviors

As we mentioned in chapter 4 (section 7), assignment behaviors determine how student contracts are modified when assignment statuses change from one value to another. Assignment behaviors are term specific because assignments are. Figure 12.8 shows the working window for Assignment Behavior.

Figure 12.8 Assignment Behaviors working window.

As you can see in this example, we determine the starting status (From) and the final status (To) and

give the eventuality a behavior and description. In general, you should have a record for every permutation of from and to that the system can handle. When you enter these, the drop lists in the From and To data elements are pre-populated with the possible assignment status choices.

Since there are many possible combinations of from and to, you may end up with a lot of behavior records. In order to keep from having to enter these over and over from semester to semester, we have also included the “Roll” function (as in the last section) to assist you in setting up new terms.

## 12.2 SYSTEM TABLES

The selections of functions under the System Tables higher level functions fall into two different categories - the ones with a preceding asterisk and those without. The functions that have the asterisk are system supplied tables and should not be modified. They can be viewed, but values cannot be changed or deleted.

The other functions contain tables that you should supply to help with the tracking of your data. We can help you in deciding what values and records should be added, but these are tables which feed the rest of your database and help you to organize your student records.

But let’s begin by discussing the asterisked system tables.

### 12.2.1 System Supplied Tables

The tables supplied in the initial installation of *CHARLES* are there because they are mandatory to

The screenshot shows a software window titled "\*Assignment Status Table" in the top-left corner and "Life for Education Systems" in the top-right corner. The main area of the window is light gray and contains the following elements:

- A small box labeled "AS" with "Assignment Status Code" written below it.
- A larger box labeled "Assigned" with "Assignment Status Definition" written below it.
- A checkbox labeled "Status Active" which is currently checked.
- At the bottom center, two buttons labeled "Prior" and "Next" with left and right arrow icons respectively.

On the right side of the window, there is a vertical column of buttons: "Insert", "Edit", "Find", "Delete", "Show Values", "OK", and "Cancel".

Figure 12.9 Assignment Status working window.

the operation of the system. They should not be modified without first contacting customer support and explaining the reason for your modification.

The first table in this list is the Assignment Status table. since CHARLES works by tracking a student's assignment status from one value to the next, it is very important that assignment statuses not be changed. Figure 12.9 is the working window for Assignment Statuses.

While these tables look much like any other, you will be warned if you try to delete any of these records. The status has a code, a definition (which can be modified) and an Status Active flag. If the Status Active flag is selected (checked), this status will appear in lists of assignment statuses throughout the system. If not, it won't. AA is the only assignment status whose status active flag is not checked upon installation.

The next system supplied table is the country code table. This is a standard list of all the countries in the world. In actuality, this table can be modified, although only when a country name changes. Figure 12.10 is the working window for the Country Codes Table.

Figure 12.10 Country Codes working window.

The final system supplied table is the room change status table. Since room changes require a series of statuses to track the progress or room change requests, these codes should also be left alone to insure correct behavior when processing room changes. There are, however, some modifications that can (and probably should) be made.

Figure 12.11 shows the working window for the Room Change Status table. As you can see, there is an email message which is attached to all but the active status. The text of this email should be

modified so that your students are told what you'd like to tell them at each step in the room change process. While you are certainly free to use the boiler-plate text that we've provided, you will probably want to modify the text surrounding the embedded fields to give the email messages your own flavor.

**\*Room Change Status Table** Life for Education Systems

Offered  
Room Change Status

Email Text

The email text included here can be modified with your own messages. If you wish to include embedded data elements, please pay attention to the way they are entered and follow that format.

Embedded data elements must be entered in the format "prow.datafield" since the row named prow contains any data fields you are able to use.

Also, please be mindful of the format of the concatenate function and the positioning of the quotes and commas surrounding your embedded data.

con("Thank you for your room change offer dated ",prow.rcsubmitdate,". We are prepared to offer you a space in ",prow.tbldng," space ",prow.troomidentity,".

If you want to accept this space, you must inform us by "prow.trcholddate," or this request will be declined by our office and you will not receive a room change until you submit another request.

If you accept this offer, we will email you a confirmation and information about your move to ",prow.tbldng,".

The Housing Department")

Insert Edit Find Delete Show Values

Prior Next OK Cancel

Figure 12.11 Room Change Status working window.

Note the cautionary text on the left side of the window. If this appears daunting, please call or email customer support for assistance.

## 12.2.2 Customizable System Tables

The rest of the system tables are for you or your system administrator to modify for yourself. Some of these tables may be populated with a sample record - but some of the will not. It is up to you whether you utilize the supplied values or delete them in favor of your own nomenclature.

The Address Types Table working window appears in figure 12.12. It is simply a code and a description of what the code means. You can have as many address types as you'd like, although it is most likely that you will only use two or three different address types.

The next function is the Cancellation Reasons Table (figure 12.13). Cancellation reasons are records which allow you to tag a cancellation of housing with a reason and thus a particular assignment status. There is a cancellation reason code, a description for the code, and an assignment status drop list to determine which assignment status to attach to a person when you select this reason for cancellation. It helps to automate assignment statuses among different cancellation types, as well as provide a more complete description of the reason students leave housing.

Address Type Table	Life for Education Systems
<input type="button" value="Insert"/> <input type="button" value="Edit"/> <input type="button" value="Find"/> <input type="button" value="Delete"/>	
<div style="display: flex; justify-content: center; align-items: center; gap: 10px;"> <div style="text-align: center;"> <input style="width: 50px; border: 1px solid black;" type="text" value="P"/>              Address Type Code           </div> <div style="text-align: center;"> <input style="width: 200px; border: 1px solid black;" type="text" value="Permanent"/>              Address Type Definition           </div> </div>	
<input type="button" value="Show Values"/>	
<div style="display: flex; justify-content: center; align-items: center; gap: 10px;"> <div style="border: 1px solid black; padding: 5px; text-align: center;">             &lt; Prior           </div> <div style="border: 1px solid black; padding: 5px; text-align: center;">             Next &gt;           </div> </div>	
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

Figure 12.12 Address Types working window.

Cancellation reasons Table	Life for Education Systems
<input type="button" value="Insert"/> <input type="button" value="Edit"/> <input type="button" value="Find"/> <input type="button" value="Delete"/>	
<div style="display: flex; justify-content: center; align-items: center; gap: 10px;"> <div style="text-align: center;"> <input style="width: 100px; border: 1px solid black;" type="text" value="MPD"/>              Cancellation Reason           </div> <div style="text-align: center;"> <input style="width: 200px; border: 1px solid black;" type="text" value="Missed Payment Deadline"/>              Reason Description           </div> </div> <div style="display: flex; justify-content: center; align-items: center; gap: 10px; margin-top: 10px;"> <div style="text-align: center;"> <input style="width: 50px; border: 1px solid black;" type="text" value="CA"/> <input style="width: 20px; border: 1px solid black;" type="button" value="v"/> </div> <div style="text-align: center;">             Translate To Status           </div> </div>	
<input type="button" value="Show Values"/>	
<div style="display: flex; justify-content: center; align-items: center; gap: 10px;"> <div style="border: 1px solid black; padding: 5px; text-align: center;">             &lt; Prior           </div> <div style="border: 1px solid black; padding: 5px; text-align: center;">             Next &gt;           </div> </div>	
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

Figure 12.13 Cancellation Reasons working window.

The next function is Charge Type Tables function (figure 12.14). Charge types are the tags you place on your bed spaces to determine how that space will be charged. The values you use in this table should make sense to your institution. You may want to have them also describe the type of bed space you are charging for (as in our example). But you could, just as easily, use the codes to indicate and expense level (C for cheap, M for moderate, E for expensive, etc) or whatever else to categorize the types of charges you will generate.

In our example, we use a room type to indicate the charge. This is simply an option. The records you enter here will appear in the Rates table (the charge type drop list) as we saw in section 2.1.

Figure 12.14 Address Types working window.

We have a function to enter College Codes (figure 12.15). These are the colleges within your institution. These are shorthand ways for you to designate your students to their appropriate college so you can report, group or track them in that manner.

Note that college codes have a code, a definition and an affiliation to a school. Before entering college codes, you should first set up your school codes (even if you only have one) since you must affiliate a college with a school.

The Contact Relationships table is a way for *CHARLES* to standardize titles for emergency contact relationships. It also allows this list of relationships to appear on application windows and web pages as select lists. (figure 12.16)

The Contract Type table function is where you define the different types of contracts your system

College Codes Table	Life for Education Systems	
<div style="text-align: center;"> <div>UA</div> <div>College Code</div> </div>		<div>Insert</div>
<div style="text-align: center;"> <div>Undgraduate Arts &amp; Sciences</div> <div>College Definition</div> </div>		<div>Edit</div>
<div style="text-align: center;"> <div>UG – Undergraduate College</div> <div>School Affiliation</div> </div>		<div>Find</div>
		<div>Delete</div>
		<div>Show Values</div>
<div style="text-align: center;"> <div>&lt; Prior</div> <div>Next &gt;</div> </div>		<div>OK</div>
		<div>Cancel</div>

Figure 12.15 College Codes Table working window.

Contact Relationships	Life for Education Systems	
<div style="text-align: center;"> <div>MO</div> <div>Relationship Code</div> </div>		<div>Insert</div>
<div style="text-align: center;"> <div>Mother</div> <div>Relationship Definition</div> </div>		<div>Edit</div>
		<div>Find</div>
		<div>Delete</div>
		<div>Show Values</div>
<div style="text-align: center;"> <div>&lt; Prior</div> <div>Next &gt;</div> </div>		<div>OK</div>
		<div>Cancel</div>

Figure 12.16 Contact Relationships working window.



Figure 12.17 Contract TypeTable working window.

uses. The contract type code should correspond to an entry housing type (F, T or whatever you use) so that when you enter a person of that type, *CHARLES* will know to assign the correct type of contract. Figure 12.17 shows the working window for this function.

Notice in figure 12.17 that there is a default number of semesters. This is how *CHARLES* keeps track of when a student has used up their allocation of semesters. We can also decide whether the contract type has variable semesters or is the default contract.

A default contract is the contract that *CHARLES* assigns if an entered application has no contract and *CHARLES* cannot determine which contract to assign to the application. *CHARLES* only allows one default contract in the system.

You may have noticed while we've been looking at all these similar pictures of the system tables the buttons on the right side of the working windows. There is one called Show Values. This is a way you can look at all the entered values for these tables in one list. Figure 12.18 shows the values for the Contract Type table.

Of course you could use the next and prior buttons at the bottom of the working window to page through the values. We thought it might be nice to have the ability to see all the values in the table in a single glance. In figure 12.18, you can see the actual server field names, the title of the server table and all the fields in the table. If there are more fields than will fit on the window, you can use the grow handle in the lower right corner to expand the window, or use the horizontal and vertical scroll bars to view the values.

contract_type	contract_definition	contract_seq	contract_semester	contract_isfixed	contract_isdefault
F	Standard Undergra	1	8		True
T	Transfer Contract	2	4		False
W	Waitlist Contract	3	1		False
GD	Graduate Contract	4	2		False

Figure 12.18 Results from the Contract Types table after pressing the Show Values button.

The next function in the list is the Degree Types table. This is where we enter the types of degrees students may be working toward and associate them with a particular student type. Therefore, before degree types can be entered, we must first enter some student types. (figure 12.19)

**Degree Types** Life for Education Systems

Insert Edit Find Delete

BA  
Degree Code

Bachelor of Arts  
Definition

UG - Undergraduate Students  
Student Type

Show Values

Prior Next OK Cancel

Figure 12.19 Degree Types function working window.

The Hall Codes Table function is a back-door way to view and modify the codes we attached to our spaces in chapter 3. Remember, we associated codes to our residence halls (AL for Albright, etc) and those codes were used as prefixes for our bed spaces. These codes are stored in this system table - and we can get to them one by one. Figure 12.20 shows the working window.

The screenshot shows a software window titled "Hall Codes Table" with a subtitle "Life for Education Systems" in the top right corner. The window contains a form for editing hall code information. The form fields are as follows:

- Hall Code:** A text box containing "AL".
- Hall Tag:** A text box containing "Albright".
- Hall Title:** A text box containing "Albright Residence".
- Hall Street:** A text box containing "14 College Way".
- Hall City:** A text box containing "Nontown".
- Hall State:** A text box containing "MA".
- Hall ZipCode:** A text box containing "01006-".
- Active/Inactive:** Two radio buttons, with "Inactive" selected.

On the right side of the window, there are five buttons: "Insert", "Edit", "Find", "Delete", and "Show Values". At the bottom of the window, there are three buttons: "Prior", "Next", and "OK". A "Cancel" button is also present at the bottom right.

Figure 12.20 The Hall Codes Table function working window.

Notice that each hall code has its own title and address. Remember that there can be more than one hall code attached to a hall. In this way, we are able to group particular buildings to a single hall name if that is how our campus is set up.

The Hall Styles Table function is to allow us to define the styles of halls we have in our inventory of residence halls. We can then use the values in this table to tag our halls with the style to better clarify the type of accommodations that exist in our residences. This also helps us define charge types for summer housing (if it is not treated as another academic term). Figure 12.21 is a representation of the working window for this function.

The Housing Types Table function is where we define the types of students we have in our housing database. This is where we indicate which classification of student the housing type belongs to and which contract needs to be assigned to the type. Figure 12.22 shows the working window for this function.

Note that since we must tie a housing type to both a student type and a contract, these tables must be populated before housing types can be entered.

The only other thing you need to provide is the definition of the housing type.

Hall Styles Table	Life for Education Systems	
<div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 60%;"> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: inline-block;">APT – WITH A/C</div> <div style="margin-left: 10px;">Hall Style Name</div> </div> <div style="width: 35%; text-align: right;"> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: inline-block;">Apartment Style with A/C</div> <div style="margin-left: 10px;">Hall Style Description</div> </div> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 100px;"> <div style="width: 30%;"> <div style="border: 1px solid #ccc; padding: 5px; text-align: center; width: 40px; margin: 0 auto;">             ◀ Prior           </div> </div> <div style="width: 30%;"> <div style="border: 1px solid #ccc; padding: 5px; text-align: center; width: 40px; margin: 0 auto;">             ▶ Next           </div> </div> <div style="width: 30%; text-align: right;"> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px; display: inline-block;">OK</div> <div style="border: 1px solid #ccc; padding: 5px; display: inline-block;">Cancel</div> </div> </div>		<div style="text-align: right;"> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: inline-block;">Insert</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: inline-block;">Edit</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: inline-block;">Find</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: inline-block;">Delete</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: inline-block;">Show Values</div> </div>

Figure 12.20 The Hall Styles function working window.

Housing Types Table	Life for Education Systems	
<div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 60%;"> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: inline-block;">F</div> <div style="margin-left: 10px;">Housing Type</div> </div> <div style="width: 35%; text-align: right;"> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: inline-block;">UG Undergraduate Stude...</div> <div style="margin-left: 10px;">Student Type</div> </div> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 10px;"> <div style="width: 30%;"> <div style="border: 1px solid #ccc; padding: 5px; text-align: center; width: 40px; margin: 0 auto;">             ◀ Prior           </div> </div> <div style="width: 30%;"> <div style="border: 1px solid #ccc; padding: 5px; text-align: center; width: 40px; margin: 0 auto;">             ▶ Next           </div> </div> <div style="width: 30%; text-align: right;"> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px; display: inline-block;">OK</div> <div style="border: 1px solid #ccc; padding: 5px; display: inline-block;">Cancel</div> </div> </div>		<div style="text-align: right;"> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: inline-block;">Insert</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: inline-block;">Edit</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: inline-block;">Find</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: inline-block;">Delete</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; display: inline-block;">Show Values</div> </div>

Figure 12.21 The Housing Types Table function working window.

If you accept payments, you will need to enter some payment codes. This will allow your system to choose the type of payment you are receiving from a drop-down list as we saw in figure 4.5. The Payment Code Table working window is in figure 12.22.

The screenshot shows a software window titled "Payment Codes Table" with a subtitle "Life for Education Systems". The main area is a light gray rectangle. In the center, there are two buttons: "AMEX" with "Payment Code" below it, and "American Express" with "Payment Code Definition" below it. On the right side, there are five buttons stacked vertically: "Insert", "Edit", "Find", "Delete", and "Show Values". At the bottom center, there are two buttons: "Prior" and "Next". At the bottom right, there are two buttons: "OK" and "Cancel".

Figure 12.22 The Payment Codes Table function working window.

If you have a number of hall types and you'd like to group them for reporting purposes, you can enter some hall types using the Residence Hall Types function (figure 12.23). This provides you the ability to sort your reports by hall types. Your institution may use college owned facilities and leased spaces and temporary use hotel spaces, etc. This is how you would be able to select them from the list when you define your residence halls (as in figure 6.1).

You should be aware of the differences in Residence Hall Types and Residence Hall Styles. Hall Styles pertain more to the attributes of the hall while Residence Hall Types are a grouping of your halls. Figure 12.24 shows the values for our fictitious residence halls.

The Room Change Priorities function allows you to provide priorities for residents to choose when filling out or submitting a room change request. You may have as many priorities as you'd like. When they appear in a drop-list or as a select table on a web form, they will be ordered by number and contain the description. Figure 12.25 shows the working window for this function.

The School Codes Table function is where you define the school codes and descriptions that we saw when we had to define entries in the College Codes table (figure 12.15). Some institutions will only have one entry in this table, specifically small liberal arts colleges. But others, like large universities, may have several schools and, therefore, many entries. Figure 12.26 is the working window for the School Codes Table function.

Residence Hall Types Life for Education Systems

APT STYLE  
Hall Type Code

Apartment Style  
Hall Type Description

Figure 12.23 The Residence Hall Types function working window.

Server Table Values for F\_halltypes

halltype_id	halltype_code	halltype_desc
1	HOTEL	Hotel Accommodation
2	TRAD STYLE	Traditional Dorm Style
3	APT STYLE	Apartment Style
4	LEASED	Lease Apartment Spaces

Figure 12.24 Values from our fictitious college's hall types table.

Room Change Priorities	Life for Education Systems
	<input type="button" value="Insert"/> <input type="button" value="Edit"/> <input type="button" value="Find"/> <input type="button" value="Delete"/>
<div style="border: 1px solid black; display: inline-block; padding: 2px 10px;">2</div> Priority Number	
<div style="border: 1px solid black; display: inline-block; padding: 2px 50px;">Very Important</div> Definition	<input type="button" value="Show Values"/>
<div style="display: inline-block; border: 1px solid black; padding: 5px; margin: 5px;"> <span style="color: green;">◀</span> Prior           </div> <div style="display: inline-block; border: 1px solid black; padding: 5px; margin: 5px;"> <span style="color: green;">▶</span> Next           </div>	<input type="button" value="OK"/> <input type="button" value="Cancel"/>

Figure 12.25 The Room Change Priorities function working window.

School Code Table	Life for Education Systems
	<input type="button" value="Insert"/> <input type="button" value="Edit"/> <input type="button" value="Find"/> <input type="button" value="Delete"/>
<div style="border: 1px solid black; display: inline-block; padding: 2px 10px;">UG</div> School Code	
<div style="border: 1px solid black; display: inline-block; padding: 2px 50px;">Undergraduate College</div> School Definition	<input type="button" value="Show Values"/>
<div style="display: inline-block; border: 1px solid black; padding: 5px; margin: 5px;"> <span style="color: green;">◀</span> Prior           </div> <div style="display: inline-block; border: 1px solid black; padding: 5px; margin: 5px;"> <span style="color: green;">▶</span> Next           </div>	<input type="button" value="OK"/> <input type="button" value="Cancel"/>

Figure 12.26 The School Code Table function working window.

The Student Types Table function is where we define our student types. Remember that we needed this before entering housing types since housing types are a subset of student types. Basically, student types are things like Undergraduate or Graduate or Student Staff and the like. Figure 12.27 is the working window for entering and modifying this table.

The screenshot shows a software window titled "Student Types" with a subtitle "Life for Education Systems". The window contains a central area with a text input field containing "UG" and a label "Student Type" below it. Below this is another text input field containing "Undergraduate Students" and a label "Definition" below it. To the right of the central area are five buttons: "Insert", "Edit", "Find", "Delete", and "Show Values". At the bottom of the window are three buttons: "Prior" (with a left arrow), "Next" (with a right arrow), and "OK". A "Cancel" button is also present at the bottom right.

Figure 12.27 The Student Types function working window.

The Suite Types function is where you define the suite types for your residence halls to use. When we create our spaces, we begin by entering a suite type. Here is where those came from. Suite types are a way to let your students know how many rooms or spaces might be in their assignment and these codes can be used in reports and other information you may want to pass along to your student. For instance, if a suite type code has a description of "Three Bedroom", then you may surmise that there are three individual rooms in the suite. The working window for this function is in figure 12.28.

The screenshot shows a software window titled "Suite Types" with a subtitle "Life for Education Systems". The window contains a central area with a text input field containing "3B" and a label "Suite Type Code" below it. Below this is another text input field containing "Three Bedroom" and a label "Suite Type Definition" below it. To the right of the central area are five buttons: "Insert", "Edit", "Find", "Delete", and "Show Values". At the bottom of the window are three buttons: "Prior" (with a left arrow), "Next" (with a right arrow), and "OK". A "Cancel" button is also present at the bottom right.

Figure 12.28 The Suite Types function working window.



The Summer Categories Table working window appears in figure 9.4 in section 9.3 so we won't rehash it here.

The last function in the System Tables higher level functions is called the Waitlist Categories Table. Generally, on-campus housing systems have more demand than they have spaces. If this is your case, you would probably want to prioritize this demand into a table of ordered categories. These are wait list categories (figure 12.29).

The screenshot shows a software window titled "Waitlist Categories Table" with a subtitle "Life for Education Systems". The main area contains a form with the following elements:

- A text input field labeled "Waitlist Code" containing the letter "A".
- A text input field labeled "Missed Payment Deadline" containing the text "Description".
- Three radio buttons for selecting a category: "Non-commutable" (which is selected), "International", and "Commutable".

On the right side of the window, there is a vertical column of buttons: "Insert", "Edit", "Find", "Delete", "Categorize", and "Show Values". At the bottom of the window, there are four buttons: "Prior" (with a left arrow), "Next" (with a right arrow), "OK", and "Cancel".

Figure 12.29 The Wait List Categories function working window.

Wait list categories consist of a code, description and a commutability flag with three values. This table allows you to assign some simple priority to people you place on the wait list.

There are several ways for people to become denizens of the wait list. If a person wants to live in housing but did not apply on time (according to your stated deadlines), you may still allow that person to apply - but stipulate that their application is placed on a wait list and that once all of your residents are assigned for the term, you will be able to make them a housing offer if and when space becomes available.

Residents may cancel their housing at some point during their tenure at your institution and want back in once they realize the harsh reality of the "real" housing market.. Of course,your guidelines are your own to make but placing people onto a wait list is one way to retain potential residents without having to deal with them until after your system is fully assigned for the term.

## 12.3 OTHER SYSTEM ADMINISTRATION FUNCTIONS

There are a few other very important functions that should be discussed in detail and we will look at them in their own sub categories.

### 12.3.1 User Groups

User groups is a very important concept for how your system behaves and who, among your staff, gets access to the various functions in *CHARLES*. As we've seen before, groups are the entity which is given access. Individual staff members are then associated to these groups. So, in effect, users are endowed with privileges which are bestowed upon them by their group membership.

Figure 12.30 shows the User Groups function.

**User Groups** Life for Education Systems

Group ID: 2 Group Name: Acad Yr Housing Admin Group Tag: AY Admin

**Main Functions**

System Administration	Academic Year Housing	Summer Housing	Hall Operations	Facilities Maintenance	Web Portal Administration
<input checked="" type="radio"/> No Access <input type="radio"/> Full Access	<input type="radio"/> No Access <input checked="" type="radio"/> Full Access	<input checked="" type="radio"/> No Access <input type="radio"/> Full Access	<input type="radio"/> No Access <input checked="" type="radio"/> Full Access	<input type="radio"/> No Access <input checked="" type="radio"/> Full Access	<input type="radio"/> No Access <input checked="" type="radio"/> Full Access

**Activity Components**

Applications	Assignments	Space Management	Accounts	Correspondence	Judicial	Meal Plans
<input type="radio"/> No Access <input type="radio"/> View Only <input type="radio"/> Limited Access <input checked="" type="radio"/> Full Access	<input type="radio"/> No Access <input type="radio"/> View Only <input type="radio"/> Limited Access <input checked="" type="radio"/> Full Access	<input type="radio"/> No Access <input type="radio"/> View Only <input type="radio"/> Limited Access <input checked="" type="radio"/> Full Access	<input type="radio"/> No Access <input type="radio"/> View Only <input type="radio"/> Limited Access <input checked="" type="radio"/> Full Access	<input type="radio"/> No Access <input type="radio"/> View Only <input type="radio"/> Limited Access <input checked="" type="radio"/> Full Access	<input checked="" type="radio"/> No Access <input type="radio"/> View Only <input type="radio"/> Limited Access <input type="radio"/> Full Access	<input checked="" type="radio"/> No Access <input type="radio"/> View Only <input type="radio"/> Limited Access <input type="radio"/> Full Access

Prior Next OK Cancel

Figure 12.30 The User Groups function working window.

As you can see, user groups have names and tags. This is to identify them in the Users function which we will look at next. User groups also have a number of Main Function switches that pertain to the various working functions within *CHARLES*. In our example, the AY Admin group has full access to all the main functions except Summer Housing. This means that a member of this group would not see Summer Housing functions in their function list within *CHARLES*.

Activity components are how you can give access to functions within the various working windows in *CHARLES*. We see in figure 12.30 that this group has full access to applications. This means that members of the group will be able to access all the buttons which initiate actions on the application working windows within the system. No access removes applications from their functions list. View only allows them to see but not touch (basically giving them only a find button). Limited access allows

them to edit but not insert. Full access allows them to do anything they want on those working windows.

You are able to define as many user groups as you want to create as much granularity of access as your institution requires.

### 12.3.2 Users

The Users function allows you to set up individuals with log in names and assign them to groups (figure 12.31).

The screenshot shows the 'Users' function window. The title bar reads 'Users' and the subtitle is 'Life for Education Systems'. The main area contains the following fields and controls:

- User ID:** 2
- User Login:** harry
- User Name:** Pierce, Harry
- User's Title:** Asst. Director Central Housing
- Group:** AY Admin (dropdown menu)
- Status:** Inactive (radio button), Active (radio button, selected)
- Entered:** MAR 2 2009
- Activated:** MAR 2 2009

On the right side, there are buttons for 'Insert', 'Edit', 'Find', 'Delete', 'Reset Password', and 'Show Values'. At the bottom, there are buttons for 'Object Access', 'Prior', 'Next', 'OK', and 'Cancel'.

Figure 12.31 The Users function working window.

As you can see, our user (Harry) has been given access to the group in figure 12.30. He will log by entering his User Login. Passwords are initially created by using the first two characters of his first name and the first two characters of his last name. *CHARLES*, when seeing a person logging in using this password, flags the user to change their password. Users can use any password they like. So Harry's password when we create his user record is set to "HaPi". It is important that user names be entered in the format seen in figure 12.31 - last name, followed by a comma and space, then first name. This way, *CHARLES* will know how to set up the user's initial password.

Note also that users have an inactive/active flag on their record. Users who are not active will no longer have access to the system meaning they will not be able to log in. If you need to suspend a user's access, simply edit the record and select the Inactive button.

We also have the ability to reset a person's password. If a user forgets their password, we can set it back to its default by pressing the Reset Password button. Again, when the user logs into *CHARLES*, they will be prompted for a new password.

### 12.3.3 Data Import Utility

We have briefly discussed the data import utility in section chapter 4, section 2. The data import utility is a method to allow you to import data from some other source outside of *CHARLES*. Our discussion dealt with populating the system with newly admitted freshmen when we begin our housing cycle but you can use the utility to import or update many other combinations of data items simply by designing an import scheme.

Figure 12.32 shows the data import utility working window. You can see in our example some of the features it uses.

double-click a line to remove a field  
drag a line to rearrange the order

Order	Field Name	Table Name	Match
1	s_id	F_studentdemo	<input checked="" type="checkbox"/>
2	s_firstname	F_studentdemo	<input type="checkbox"/>
3	s_middlename	F_studentdemo	<input type="checkbox"/>
4	s_lastname	F_studentdemo	<input type="checkbox"/>
5	s_gender	F_studentdemo	<input type="checkbox"/>
6	s_birthdate	F_studentdemo	<input type="checkbox"/>
7	s_gradate	F_studentdemo	<input type="checkbox"/>
8	sa_street1	F_addresses	<input type="checkbox"/>
9	sa_street2	F_addresses	<input type="checkbox"/>
10	sa_city	F_addresses	<input type="checkbox"/>
11	sa_state	F_addresses	<input type="checkbox"/>
12	sa_postcode	F_addresses	<input type="checkbox"/>
13	sa_country	F_addresses	<input type="checkbox"/>
14	sa_phone	F_addresses	<input type="checkbox"/>
15	sa_sid	F_addresses	<input checked="" type="checkbox"/>
16	term_id	F_termstudent	<input checked="" type="checkbox"/>
17	term_term	F_termstudent	<input type="checkbox"/>

☐ Add Records only  
☐ Update Records only  
☒ Add... update if found  
☒ Import has column header

☒ Create results file  
 Rover:Users:david:Desktop:FRimport.results

☒ Create error file  
 Rover:Users:david:Desktop:FRimport.errors

Run This Import

Print A Template

Date Entered: 03/03/2009

Insert  
 Edit  
 Delete  
 Find  
 Next  
 Prior  
 Ok  
 Cancel

Figure 12.32 The Data Import Utility working window.

Imports are actually records in your database. *CHARLES* allows you to save import formats and run them whenever you need to. You are also able to specify a name for your import, the format the import file has (either tab or comma delimited). You determine how records are treated when they are imported. The three selections are to:

- 1) Add records only - this will insert new records into your database. This is sometimes a very good option but could result in inserting duplicate records. You should be careful about when you use this option.
- 2) Update records only - this will look to see whether the record exists and, if it does, will update the values. If the record does not exist, it will generate an error.

- 3) Add... update if found - this is probably the safest import method as it will not generate duplicate records. This choice will look for an existing record and if it exists, it will update it. If it does not exist it will insert the record.

The Import has column header check box is to let *CHARLES* know whether to treat the first line of the import file differently. If the check box is checked, the import will ignore anything in the first row of the import file. If it is not checked, the import will treat the first row as it treats all the others, based upon your Add, Update or Add...update if found selection.

If you'd like to create a results file and/or an error file, you may designate a path for these files. Checking the boxes prompts you for a destination and name for your results or error file. When you run your import, results and errors will be written to these files so you can verify the results of your import.

The two buttons at the bottom of the working window allow you to Run the import, or Print a Template. Printing a template prompts you for a file location for the template. Templates use the import format you decide upon to generate a file containing a header with your data field names. These can be opened in a work sheet program (excel or open office or numbers) to help you provide data for your system.

Running the import is as simple as pressing the button. *CHARLES* will prompt you for the location of the import file and process its data.

Of course, data imports will not work without having some data fields to insert or update. Figure 12.33 shows us editing our import.

The screenshot shows the 'Data Import Utility' window. The 'Import Name' is 'Freshman Import' and the 'Import Format' is 'tabs'. The 'Import Fields' section contains a table with columns 'Order', 'Field Name', and 'Match'. A dropdown menu is open over the 'Field Name' column, showing options: 'Select Table', 'Placeholder', 'F\_addresses', 'F\_assignment', and 'F\_contractasgn'. The table lists 17 fields, with 's\_id', 'term\_id', and 'term\_term' checked in the 'Match' column. On the right, there are radio buttons for 'Add Records only', 'Update Records only', and 'Add... update if found' (selected). There are also checkboxes for 'Import has column header', 'Create results file', and 'Create error file'. The 'Date Entered' is '03/03/2009'. Buttons for 'Run This Import', 'Print A Template', 'Insert', 'Edit', 'Delete', 'Find', 'Next', 'Prior', 'Ok', and 'Cancel' are visible.

Order	Field Name	Match
1	s_id	<input checked="" type="checkbox"/>
2	s_firstname	<input type="checkbox"/>
3	s_middlename	<input type="checkbox"/>
4	s_lastname	<input type="checkbox"/>
5	s_gender	<input type="checkbox"/>
6	s_birthdate	<input type="checkbox"/>
7	s_graddate	<input type="checkbox"/>
8	sa_street1	<input type="checkbox"/>
9	sa_street2	<input type="checkbox"/>
10	sa_city	<input type="checkbox"/>
11	sa_state	<input type="checkbox"/>
12	sa_postcode	<input type="checkbox"/>
13	sa_country	<input type="checkbox"/>
14	sa_phone	<input type="checkbox"/>
15	sa_sid	<input checked="" type="checkbox"/>
16	term_id	<input checked="" type="checkbox"/>
17	term_term	<input checked="" type="checkbox"/>

Figure 12.33 Editing the import to add additional data fields.

When we create our import, we select the tables we are importing into from the drop list. Selecting a table populates the drop list on the right side with the data elements from the table in the left drop list. When tab off of the data field drop list, the data field is added to the import.

Notice in figure 12.33 that there is a table called “Placeholder”. This is used to inform *CHARLES* to ignore a particular column in your import file. You would select this “field” if there are data elements in your import that you don’t want or need.

It should also be noted that as you add data elements to your import format, *CHARLES* will check to be sure that fields that are necessary for the proper functioning of the database are included. This is to maintain data integrity and ensure that links between your data tables are not compromised. If you are familiar with database theory, this can be said to cause adherence to constraints.

But let’s try to import some data and see how it works. Pressing the run this import button gives us a prompt to locate the file we are importing (figure 12.34). This file is on our desktop and called FRimport.txt. We select this and press the Open button.

The import will then process. When it is complete, you are told that it has done and where to find the results as in figure 12.35.

We can use our text editor to view the results and error files. The results file tells you everything that was done. The error file tells you only where *CHARLES* had a problem.

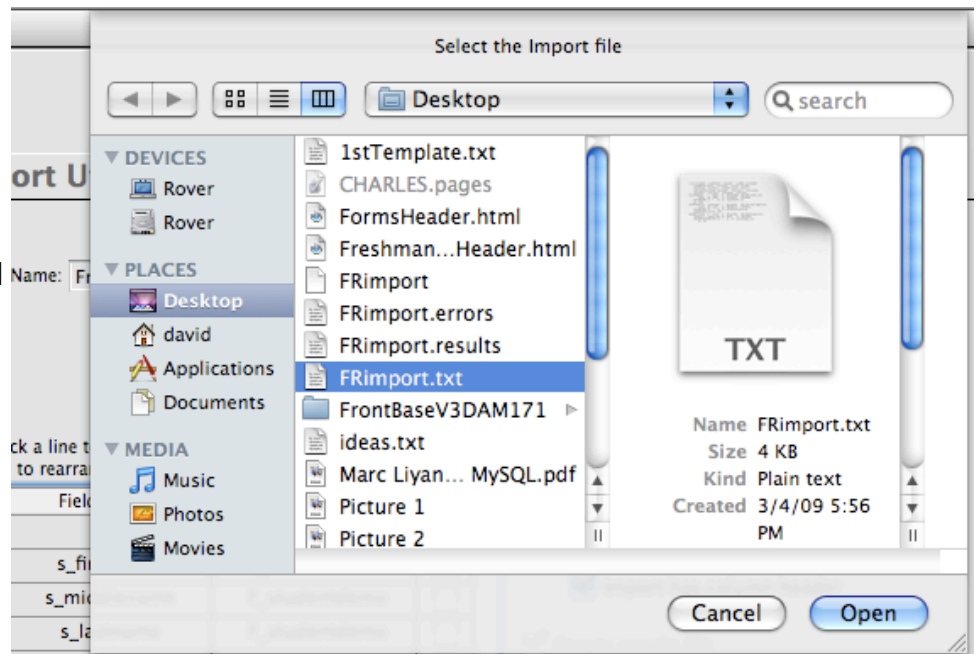


Figure 12.34 CHARLES prompts us for the location of the import file.

Figure 12.36 shows the results file and gives you an indication of what you can expect to see there. As you can determine, *CHARLES* was able to insert all the values that were supplied in our import file.

Error reports attempt to explain why something couldn’t be done. Figure 12.37 is a sample of an error report for this same import file but attempting to Add records only.

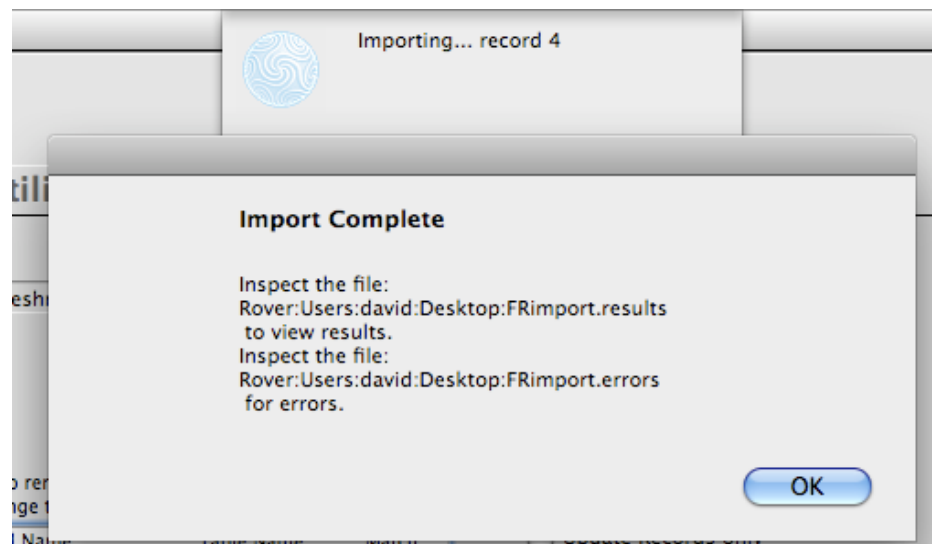
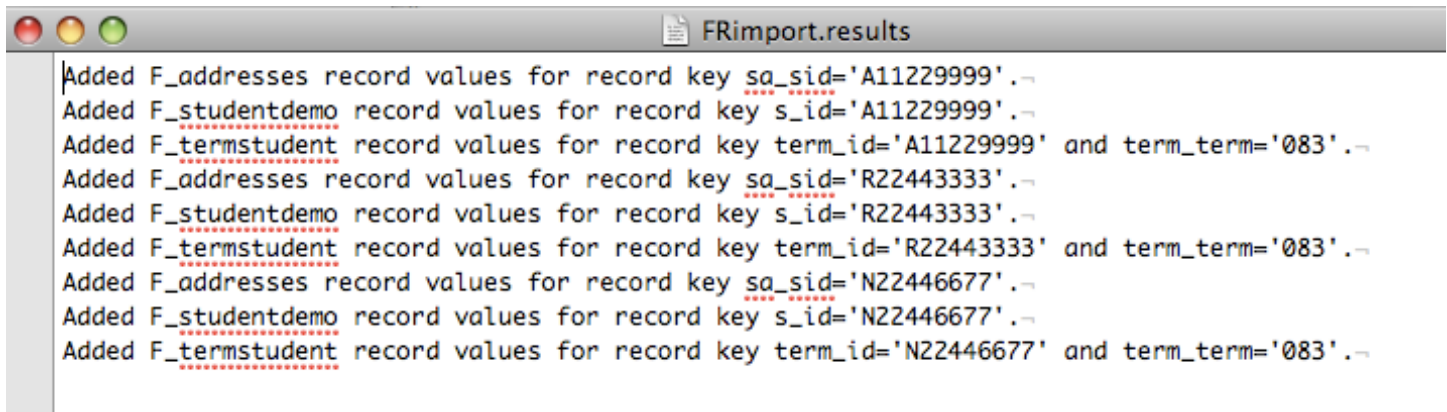


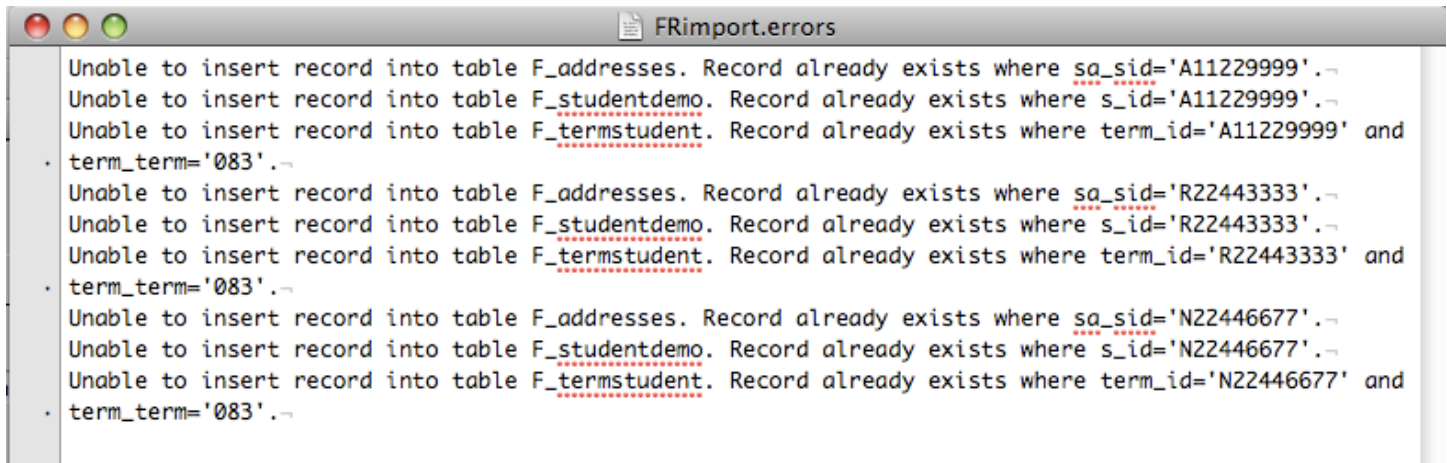
Figure 12.35 Import complete message displays where to find results and errors.





```
Added F_addresses record values for record key sa_sid='A11229999'.-
Added F_studentdemo record values for record key s_id='A11229999'.-
Added F_termstudent record values for record key term_id='A11229999' and term_term='083'.-
Added F_addresses record values for record key sa_sid='R22443333'.-
Added F_studentdemo record values for record key s_id='R22443333'.-
Added F_termstudent record values for record key term_id='R22443333' and term_term='083'.-
Added F_addresses record values for record key sa_sid='N22446677'.-
Added F_studentdemo record values for record key s_id='N22446677'.-
Added F_termstudent record values for record key term_id='N22446677' and term_term='083'.-
```

Figure 12.36 The results file from running our import.



```
Unable to insert record into table F_addresses. Record already exists where sa_sid='A11229999'.-
Unable to insert record into table F_studentdemo. Record already exists where s_id='A11229999'.-
Unable to insert record into table F_termstudent. Record already exists where term_id='A11229999' and
• term_term='083'.-
Unable to insert record into table F_addresses. Record already exists where sa_sid='R22443333'.-
Unable to insert record into table F_studentdemo. Record already exists where s_id='R22443333'.-
Unable to insert record into table F_termstudent. Record already exists where term_id='R22443333' and
• term_term='083'.-
Unable to insert record into table F_addresses. Record already exists where sa_sid='N22446677'.-
Unable to insert record into table F_studentdemo. Record already exists where s_id='N22446677'.-
Unable to insert record into table F_termstudent. Record already exists where term_id='N22446677' and
• term_term='083'.-
```

Figure 12.37 A sample error report.

### 12.3.4 Institution Specifics

We have mentioned the Institution Specifics several times in this manual. There's not much else to say about this important piece of your system except to restate how your system's look and feel, as well as some of its behavior derive from this function.

This function is, really, your first introduction to your system. Upon system initiation, it is this function which pops up and asks you to name your college.

Figure 12.38 is a picture of this function for our fictitious college. Looking at this picture, we should just go over the importance of the data there.

First, there are three ways to identify your system. An identifier called acronym is a short way to name your institution. The full name is a larger field where you can fully describe the name of your school. The abbreviated name is the name that appears in report headers.

The lower section with the tabs across the top is where you define various actions for you system to carry out. The left side of the System Preferences pane contains a number of radio buttons where you can choose various system tools to help your institution interface with various external systems. Some of these are built into *CHARLES*, some are not. In order for you to utilize any external systems, you would have to contact customer support to determine your system's interface requirements. Interfaces to any external bursar, meal plan tracking system or registration system can be devised.

The screenshot shows the 'Institution Specifics' window with the 'System Preferences' tab selected. The window is titled 'Life for Education Systems' in the top right corner. At the top, there are three input fields: 'Acronym' (BEST), 'Full Name' (Best College of Music Somewhere, USA), and 'Abbreviated Name' (Best Music College). To the right of these are buttons for 'Insert', 'Find', 'Edit', and 'Delete'. Below the input fields are two tabs: 'System Preferences' (active) and 'Pictures'. The 'System Preferences' tab contains several sections of radio buttons and text input fields. The 'Billing System' section has four options, with 'Use Internal Billing System' selected. The 'Meal Plans' section has four options, with 'Do not use Meal Plan Functions' selected. The 'Registration System' section has four options, with 'Do not export addresses' selected. The 'Facilities System' section has two options, with 'DO NOT Include Facilities Tracking' selected. There are also four checkboxes at the bottom: 'Use SSN as Student ID' (unchecked), 'System allows smoking' (unchecked), 'System generates estimates' (checked), and 'Summer as Academic Term' (unchecked). On the right side of the 'System Preferences' tab, there are several text input fields: 'SMTP server' (smtp.best.edu), 'eMail Return Address' (housing@best.edu), 'eMail Suffix' (@best.edu), 'SMTP User Name' (admin), 'SMTP User Password' (masked with dots), 'Web Home' (http://www.best.edu), and 'Web Server' (http://127.0.0.1/). At the top right of the 'System Preferences' tab, there is a dropdown menu for 'Default Term' (Fall 2008) and a text input field for 'Current Version' (1.6). At the bottom right of the window are 'OK' and 'Cancel' buttons.

Figure 12.38 The Institution Specifics working window.

You'll notice in our sample that we are using the internal billing mechanism. That means we will bill and track all housing residents ourselves.

Also we have decided to not use Meal Plan functions. Meal plan functions in *CHARLES* are an additional assignment scheme whereby your institution is able to assign and track meal plan assignments, charges and changes. When you decide to assign and bill for meal plans, this data element must be turned on. That would not only activate an interface to a meal plan transaction system like Blackboard or Diebold, but it would also allow you to populate a table of available meal plans for your population to select from and create rates and billing records for them.

The Registration System series of radio buttons allows your institution to export local addresses to your institution's registration system. Since you are the housing department, there is no one better to provide that data to your school's system of record. The interface specifics must be delivered to customer support before we will be able to activate this interface.

Information on the right of the System Preferences tab pertain to how certain pieces of data interact with your institution's environment. This is where you determine what the default term is that appears on all the working windows. If a default term is not set, all the working windows will default to the newest term you have created.

The SMTP server, email return address, SMTP user name and password together determine how *CHARLES* is able to send email to your student population. The Web Home field indicates where the main web page for your institution is located. The Web Server field is the address of your *CHARLES* web server where your population can access the *CHARLES* web portal.



The rest of the parameters on this tab are preferences for how certain functions behave within *CHARLES*. The Room Changes parameter tells *CHARLES* to place the Room Change functions either in Academic Year Housing (where they will show up in the Assignments higher level function) or in the Residence Life higher level functions (if you want your hall managers to handle room changes).

There has been a move away from using Social Security numbers as student ID's all across universities in recent years. While *CHARLES* is built to use any 9 character alpha/numeric as a student ID, you can still utilize a SSN as a student ID if you select this check box.

The System allows smoking check box will add a smoking parameter to halls and student records so that if you allow smoking in your halls, you will be able to match students to their smoking habits.

The System generates estimates allows you to set up estimated rates for students to be charged in lieu of assignment. If this is not checked, students who are not assigned will not be charged until they receive an assignment.

The Summer as Academic Term allows you to use summer as another academic term - meaning all summer housing functions will be handled inside the Academic year higher level functions. The Summer Housing higher level functions we spoke about in chapter 9 would not be relevant.

The Pictures Tab (figure 12.39) is where you can paste in the splash screen picture for your system. You can also paste in your institution's logo and a letterhead image for use in the reports you generate. It is also where you can fully enter the address of your institution.

**Institution Specifics** Life for Education Systems

Acronym  Full Name  Abbreviated Name

System Preferences **Pictures**

Letterhead

Logo

SplashScreen Picture

Academic Year

Summer

Street Address

City, State PostCode

Phone  Fax

Figure 12.39 Paste in your own pictures to make *CHARLES* look like it's yours.

### 12.3.5 Version Updates

As with any system, CHARLES is constantly being developed with new and useful functions and enhancements. The Version Updates function is a way for you to ensure that all your users are receiving these updates at the moment you receive them from customer support.

The entire CHARLES program is actually stored in your database as a binary. When updates are delivered to you, whether they contain small bug fixes or an entirely new version of CHARLES, you can place the updates directly inside your database. Then, when one of your users logs into CHARLES, their local version of CHARLES is overwritten with the newer one. Everybody will be receiving the most current version of the program at the time it is available.

As the system administrator for your system, you will be responsible for placing the program into the database. The Version Updates function is how you do that. You will notice in figure 12.40 that there are two different buttons to go along with the two different components to your CHARLES installation.

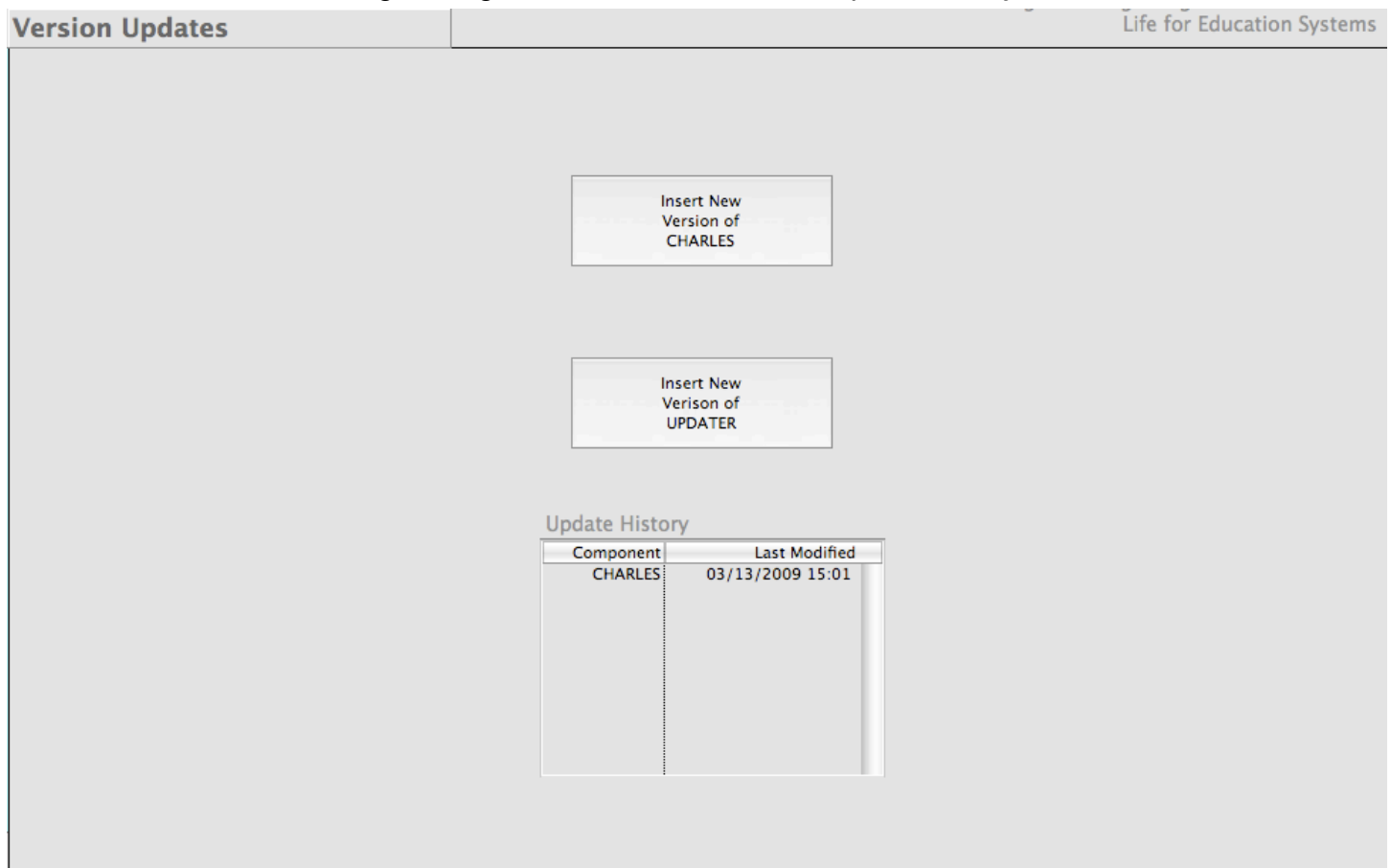


Figure 12.39 The Version Updates working window.

The first is the new version of CHARLES. When our company needs to push out a new version of CHARLES, whether it is a minor release or just for a bug fix, we will send a file to your system administrator. This file will be the update. In order for all of your users to gain access to the updates, you must place this file into your database. The Insert New Version of CHARLES button is how you do this. Place the file in a directory on your computer where you can find it. Then press the button. You will be prompted for the location of the file that was sent to you. Once you have selected the file, press the OK button. CHARLES will inform you of your success or failure in updating the system.

The second is the Insert New Version of Updater. The Updater is the program which controls the process of updating your users. While the Updater hardly ever needs to be updated, we can never be sure that back end RDM systems do not change from time to time and this is here for the possible eventuality that we will need to update the Updater. The Updater is inserted into your CHARLES environment in the same way as new versions of CHARLES are. Follow the prompt, point to your new updater file and press OK.

The last section on the working window is a history of updates that have been performed. This is here so you can tell how responsive we are being to your needs.

## System Provided Tables

### Assignment Statuses (F\_asgnstats)

asn_statuscode char(5)	asn_statusdesc char(32)	asn_active int(smallint)
AA	Awaiting Assignment	null
AS	Assigned	1
CA	Cancelled	1
IN	Inactive	1
ML	Medical Leave Cancellation	1
SA	Study Abroad Cancellation	1
UN	Unassigned	1
WL	Wait List	1

## Country Codes (F\_countrycodes)

countrycode char(2)	countryname char(30)
AA	Afghanistan
AB	Albania
AG	Algeria
AI	Andorra
AJ	Azerbaijan
AK	Angola
AL	Anguilla
AM	Armenia
AN	Antigua & Barbuda
AR	Argentina
AS	Australia
AU	Austria
AZ	Azores
BA	Bahamas
BB	Bahrain
BC	Byelarus
BD	Barbados
BE	Belgium
BG	Belize
BI	Benin
BJ	Bermuda
BL	Bangladesh
BN	Bhutan
BO	Bolivia
BP	Botswana
BR	Brazil
BS	British Virgin Is.
BT	Brunei
BU	Bulgaria
BV	Burkina Faso
BY	Burundi
CA	Cameroon
CB	Canada
CC	Canary Islands
CE	Cape Verde
CF	Cayman Islands
CG	Cent African Rep.
CH	Chad
CL	Chile
CN	China
CO	Colombia
CQ	Comoro Islands

CR	Costa Rica
CT	Croatia
CU	Cuba
CY	Cyprus
CZ	Czech Republic
DA	Denmark
DJ	Djibouti
DO	Dominica
DR	Dominican Republic
EC	Ecuador
EG	Egypt
EL	El Salvador
EN	England
UD	United Kingdom
EQ	Equatorial Guinea
ER	Serbia
ES	Estonia
ET	Ethiopia
FA	Falkland Islands
FI	Fiji
FN	Finland
FR	France
FU	French Guiana
FY	French Polynesia
GA	Gabon
GB	Gambia
GF	Germany
GH	Ghana
GI	Gibraltar
GK	Greece
GL	Greenland
GN	Grenada
GP	Guadeloupe
GR	Georgia
GT	Guatemala
GU	Guinea
GW	Guinea-Bissau
GY	Guyana
GZ	Gaza
HA	Haiti
HD	Honduras
HK	Hong Kong
HU	Hungary
IC	Iceland
II	India

IN	Indonesia
IP	Iran
IQ	Iraq
IR	Ireland
IS	Israel
IT	Italy
IV	Ivory Coast
JE	Jericho
JM	Jamaica
JP	Japan
JR	Jordan
KA	Kampuchea
KE	Kenya
KI	Kiribati
KK	Kazakhstan
KP	North Korea
KR	Rep of Korea
KU	Kuwait
KY	Kyrgyzstan
LA	Laos
LC	Latvia
LE	Lebanon
LH	Lesotho
LI	Liberia
LM	Libya
LN	Lithuania
LT	Liechtenstein
LU	Luxembourg
MA	Macao
MB	Myanmar
MC	Madagascar
MD	Madeira Islands
ME	Malawi
MG	Malaysia
MH	Maldives
MI	Mali
MJ	Macedonia
ML	Malta
MM	Mariana Islands
MN	Martinique
MO	Moldova
MP	Mauritania
MQ	Mauritius
MR	Mexico
MT	Midway Islands

MU	Monaco
MW	Mongolia
MX	Montserrat
MY	Morocco
MZ	Mozambique
NA	Nauru
NC	Nepal
NE	Netherlands
NG	Netherlands Antilles
NI	New Caledonia
NL	New Zealand
NN	Nicaragua
NP	Niger
NQ	Nigeria
BQ	Bosnia-Herzegovina
NW	Norway
OM	Oman
PA	Pakistan
PB	Panama
PD	Papua New Guinea
PF	Paraguay
PG	Peru
PH	Philippines
PI	Pitcairn Islands
PO	Poland
PU	Portugal
QA	Qatar
RO	Romania
RU	Russia
RW	Rwanda
S2	Slovakia
SA	St. Helena
SB	St. Kitts - Nevis
SC	St. Lucia
SD	St Pierre & Miquelon
SE	St Vincent/Grenadine
SF	San Marino
SG	Sao Tome & Principe
SH	Saudi Arabia
KO	Kosovo
SJ	Senegal
SK	Seychelles
SL	Sierra Leone
SM	Sikkim
SN	Singapore

SO	Solomon Isles
SP	Somalia
SQ	South Africa
SR	Southwest Africa
SS	Spain
ST	Sri Lanka
SU	Sudan
SV	Surinam
SW	Swaziland
SX	Sweden
SY	Switzerland
SZ	Syria
TA	Taiwan
TE	Tanzania
TH	Thailand
TI	Timor
TJ	Tajikistan
TK	Turkmenistan
TO	Togo
TR	Tonga
TT	Trinidad & Tobago
TU	Tunisia
TW	Turkey
TX	Turks & Caicos
TZ	Tuvalu
UG	Uganda
UK	Ukraine
UR	United Arab Emirates
US	United States
UV	Upper Volta
UY	Uruguay
UZ	Uzbekistan
VA	Vanuatu
VE	Venezuela
VN	Vietnam
MK	Montenegro
WF	Wallis & Futuna Is.
WS	Western Samoa
YA	Yemen Arab Republic
YP	Peopls Dem Rep Yemen
ZA	Zaire
ZB	Zambia
ZI	Zimbabwe
ZY	Other
S1	Slovenia

PR	Puerto Rico
GM	Guam
AV	Aruba

## Room Change Statuses (F\_rcstatuses)

rcstatuscode char(15)	rcstatustext char(10,000)
Active	null
Offered	<p>con("Thank you for your room change offer dated ",prow.rcsubmitdate,". We are prepared to offer you a space in ",prow.tbuidng," space ",prow.troomidentity,".</p> <p>If you want to accept this space, you must inform us by ",prow.trcholddate," or this request will be declined by our office and you will not receive a room change until you submit another request.</p> <p>If you accept this offer, we will email you a confirmation and information about your move to ",prow.tbuidng,".</p> <p>The Housing Department")</p>
Approved	<p>con("Thank you for your room change offer dated ",prow.rcsubmitdate,". We have processed your re-assignment into ",prow.tbuidng," space ",prow.troomidentity,". This assignment is effective beginning ",prow.asneffectivedate,".</p> <p>Your hall desk will have information for you regarding this move. Please contact them to arrange your move. Remember that you will need to check out of your current space. You will need to follow all procedures and submit a room condition report for both your current space (before you leave) and your new space (when you check into it).</p> <p>We hope this move helps to address your housing issues and wish you a comfortable and productive stay in ",prow.tbuidng,".</p> <p>The Housing Department")</p>
Declined	<p>con("Thank you for your room change offer dated ",prow.rcsubmitdate,". Your room change request has been declined for the following reason:</p> <p>",prow.rcexplanation,".</p> <p>If you believe this is inaccurate in any way, or if you would like to discuss this action, please contact the housing department at (phone number). Of course, you may submit another request if you feel that is an appropriate action.</p> <p>Thank You,</p> <p>The Housing Department")</p>



## OPTIONAL FUNCTIONS

There are some optional functions in *CHARLES* that your institution may wish to utilize. One of these is the Meal Plan tracking and assignment function.

### a) - Meal Plan Functions

*CHARLES* is able to assign meal plans and generate charges for them. While it is not able to track individual meal plan usage transactions (like a card swipe at a dining hall), it can transmit (through an interface) meal plan student's information to any number of meal plan transaction systems like ATT Blackboard.

Meal plan tracking begins with entering meal plans that your residents can choose from. Figure A2.1 shows the working window for the Meal Plan Setup function.

Figure A2.1 The Meal Plan Setup Function's working window.

In figure A2.1, we see that meal plans have a code and a name. They also contain some number of meals which can be set up as recurring weekly meals (participants are allowed this number of meals every week) or per term (when participants run out, they can no longer eat).

You also have the ability to flag a meal plan as the default. *CHARLES* assumes that if you are using meal plan functions, then all your residents must be assigned to a meal plan. If this is not the way

your system operates and you have optional meal plan registration, then you must create an empty meal plan (or exempt from meals plan). Note that only one meal plan in your system per term can be designated at the default meal plan.

We also have the ability to restrict the meal plan to certain halls by providing a list of hall in which the meal plan cannot be used.

Finally, you are able to attach some number of declining dollars to your meal plan if your institution sells meal plans as a package of “contract meals plus discretionary-use declining dollars” (which many institutions now provide).

Figure A2.2 shows a meal plan with some hall restrictions placed on it.

The screenshot displays the 'Meal Plan Setup' window from the 'Life for Education Systems' application. At the top left, a dropdown menu shows 'Fall 2008'. The main area contains several input fields: 'Meal Code' with the value '5', 'Meal Name' with '5 Meal Plan', 'Number of Meals' with '5', and 'Declining Dollars' with '300.00'. The 'Frequency' section has 'Weekly' selected with a radio button. Below this is a checkbox for 'Flagged as Default' which is unchecked. The 'Not available in:' section features a dropdown menu currently set to 'Select Hall' and a list box containing the name 'Halsey'. At the bottom, there are 'Prior' and 'Next' navigation buttons. On the right side, a vertical toolbar contains buttons for 'Find', 'Insert', 'Edit', 'Delete', 'OK', and 'Cancel'. The 'Entered Date' field shows 'MAR 16 2009'.

Figure A2.2 Hall restrictions can prevent you from assigning a meal plan to anyone assigned to a particular hall or halls.

If you have meal plan functions turned on in your system, many of the working windows we’ve seen will change to allow meal plan assignment or information sections to appear on them. Obviously, we will be able to view information about meal plans anywhere we make assignments to meal plans. We will also be able to see meal plan information on Inquiry working windows, as well as provide rates for *CHARLES* to use when assessing meal plan assignment charges. The following few figures (A2.3 - A2.7) show where meal plan information appears throughout the system.

When you assign a student (manually), you will be prompted to select a meal plan for them. When you move a person from one space to another, *CHARLES* will check to see that the person’s current meal plan matches the meal plan’s restricted hall list.

Rates

Life for Education Systems

Fall 2008

▼

Rate Type Meal ▼

5 - 5 Meal Plan ▼

Rate Code 5 Account 55-me22

Alt Account Number

Description Dining - Five Meal Plan

Your system provides 3 types of rates to enter: Room Rates, Meal Rates and Estimates.

Room Rates must be entered for each hall and key off of charge types from the charge type table. Meal Rates key off of the meal plan entry code. Estimates key off of student housing types.

Rate 1 900.00

Rate 2 950.00

Rate 3 425.00

Rate 4

Rate 5

Rate 6

Rate 7

Rate 8

Rate 9

Rate 10

Rate 11

Rate 12

Rate 13

Rate 14

Rate 15

Insert

Find

Edit

Next

Prior

Delete

Roll

Show Values

OK

Cancel

Figure A2.3 Meal Plan rates can be entered once meal plan functions are activated.

Term Set-Up

Life for Education Systems

083

Fall

2008

▼

7

Sunday

AUG 24 2008

DEC 19 2008

SEP 2 2008

108

15

Week Begin Day

Term Begin

Term End

Classes Begin

# of days

# of weeks

AUG 1 2008

MAY 19 2009

☒ Charge Weekly

☐ Charge Daily

Cancellation Deadline

Contract End

Meal Plans

AUG 31 2008

DEC 12 2008

103

15

SEP 7 2008

MEALS BEGIN

MEALS END

# of DAYS

# of WEEKS

CANCEL BY

Interface Switches

☒ Charges (internal) ON

Term Rollover

Close Term

Find

Insert

Edit

Delete

< Prior

Next >

Show Values

OK

Cancel

Figure A2.4 Meal Plan dates appear in the Term Set-Up function.

**Assignments & Moves** Life for Education Systems

Fall 2008

N12-34-5678 Student ID Flintstone, Fred T. Name yabba NetID M Gender 0612 Grad Date

UG F Student Type UA School Standard Undergraduate Contract Contract Status ACTIVE # remaining 7

**Preferences**

Meal Plan: APT - WITH A/C Lifestyle

Hall Choices: Roommates: Early Decision:

**Matching**

☒ Match Gender  
☒ Match Staff Status  
☒ Match Grad/Ugrad

**Assignment**

Start Date: AUG 24 2008 Hall: Halsey BedSpace: H1-0001A-1

NetID	Roommate Name	BedSpace	Gender
yabba	Flintstone, Fred T.	H1-0001A-1	M

1B Suite Type 2 Suite Capacity STU Program Use  
M Gender NO GradUse  
AS Status AUG 24 2008 Effective Date MAR 9 2009 Assigned Date

**Meal Plan**

10 Meal Plan Meal Plan AUG 31 2008 Start Date AS Status AUG 31 2008 Effective Date MAR 16 2009 Assigned Date

Find Assign Move Bed for Bed Meal Plan Print Notice OK Cancel

Figure A2.5 We assign our residents to meal plans on the Assignments & Moves function working window.

**Student Inquiry** Life for Education Systems

Fall 2008

N12-34-5678 Student ID Flintstone, Fred T. Name Male Gender JAN 5 1990 Birthdate 0612 Grad Date yabba NetID

**Addresses** Cell Phone: (212) 333-3333

Type	Street	City	State	Zip	Country	Phone
P	123 Dino Way	Bedrock	NY	11000-	US	(212) 333-3333

**Current Status** History Charges/Payments Correspondence Room History Miscellaneous

UG-F Student Type UG School UA College 1 Year 1 Group/Rate Confirmation Number

**Emergency Contact** Flintstone, Wilma Person to Contact OT Relation (212) 333-3333 Phone Number (212) 333-3333 Alternate Phone

**Contract** Standard Undergraduate Contract Contract Type ACTIVE Contract Status AUG 24 2008 Contract Date 8 Semesters 1 Used 7 Remaining

**Assignment** H1-0001A-1 Bedspace 1B Suite Type Phone Number Assignment Status AS Assignment Date MAR 9 2009 Effective Date AUG 24 2008

**Roommates** Halsey - suite: H1-0001

Roommate Name	NetID	BedSpace	Effective Date
Flintstone, Fred T.	N12-34-5678	H1-0001A-1	AUG 24 2008

**Meal Plan**

10 Meal Plan Meal Plan Name AS Status AUG 31 2008 Start Date MAR 16 2009 Assigned Date

Find Student Find Suite OK Cancel

Figure A2.6 Meal Plan assignments appear at the bottom of the Student Inquiry working window.



Notice that in figure A2.8, there are two entries in the “Not available in” halls. This would work if you required everyone in those two halls to have a meal plan. If you did not include any halls in that listing, the meal plan could be assigned to anyone in any hall.

## b) - Smoking Parameter

*CHARLES* contains a (legacy) smoking parameter that can be utilized if your institution allows smoking in the residence halls. We saw this when we looked at the Institution Specifics function and its working window. If you check the box “System allows smoking” (figure 12.38), you would see the smoking parameter appear throughout the system. This parameter can be used as a matching parameter in the Assignment functions for both academic year and summer housing. It is used to set the smoking status for every space of any suite you assign students into.

The way it works is similar to the way gender flags suites as either male or female depending on the gender of the first person who gets assigned to the suite. As with gender, smoking flags will not be taken off of suites unless all residents of the suite are moved out. This eliminates the possibility of putting non-smokers with smoking students.

It differs from gender flags in the following way: There are three possible values for the smoking flag to allow for more flexibility. These are Non-Smoking, Smoking, or Either. Either represents a tolerance for smoking, or towards smokers. In fact the Either category of smoking statuses may represent people who occasionally smoke or people who don’t want their parents to know their proclivity toward this habit. When trying to find spaces for these people, *CHARLES* will allow people flagged as Either to be assigned into spaces that have been flagged as (either) smoking or non-smoking.

**Student Inquiry** Life for Education Systems

Fall 2008 N12-34-5678 Flintstone, Fred T. Male JAN 5 1990 0612 yabba

Student ID Name Gender Birthdate Grad Date NetID

Cell Phone: (212) 333-3333

**Addresses**

Type	Street	City	State	Zip	Country	Phone
P	123 Dino Way	Bedrock	NY	11000-	US	(212) 333-3333

**Current Status** History Charges/Payments Correspondence Room History Miscellaneous

UG-F UG UA 1 Non-Smoking 1

Student Type School College Year Smoking Preference Group/Rate Confirmation Number

**Emergency Contact** Flintstone, Wilma OT (212) 333-3333 (212) 333-3333

Person to Contact Relation Phone Number Alternate Phone

**Contract** Standard Undergraduate Contract ACTIVE AUG 24 2008 8 1 7

Contract Type Contract Status Contract Date Semesters Used Remaining

**Assignment** H1-0001A-1 1B AS MAR 9 2009 AUG 24 2008

Bedspace Suite Type Phone Number Assignment Status Assignment Date Effective Date

**Roommates** Halsey - suite: H1-0001

Roommate Name	Roommate ID	Roommate Suite	Roommate Effective Date
Flintstone, Fred T.	N12-34-5678	H1-0001A-1	AUG 24 2008

**Meal Plan** 10 Meal Plan AS AUG 31 2008 MAR 16 2009

Meal Plan Name Status Start Date Assigned Date

Find Student Find Suite OK Cancel

Figure A2.9 An academic year inquiry working window with smoking status being displayed.

Assignments & Moves

College Housing Assignment & Moves  
Life for Education Systems

Fall 2008

N12-34-5678  
Student ID

Flintstone, Fred T.  
Name

yabba  
NetID

M  
Gender

0612  
Grad Date

Non-Smoker  
Smoking Status

UG F  
Student Type

UA  
School

Standard Undergraduate Contract  
Contract

ACTIVE  
Contract Status

7  
# remaining

Preferences

Meal Plan

APT - WITH A/C  
Lifestyle

Hall Choices

Roommates

Early Decision

Matching

☒ Match Gender  
☒ Match Smoking  
☒ Match Staff Status  
☒ Match Grad/Ugrad

Assignment

Start Date

AUG 24 2008

Hall

Halsey

BedSpace

H1-0001A-1

NetID	Roommate Name	BedSpace	Gender	
yabba	Flintstone, Fred T.	H1-0001A-1	M	Non-Smoker

1B  
Suite Type

2  
Suite Capacity

STU  
Program Use

M  
Gender

Non-Smoking  
Smoking

NO  
GradUse

AS  
Status

AUG 24 2008  
Effective Date

MAR 17 2009  
Assigned Date

Meal Plan

10 Meal Plan  
Meal Plan

AUG 31 2008  
Start Date

AS  
Status

AUG 31 2008  
Effective Date

MAR 16 2009  
Assigned Date

Find

Assign

Move

Bed for Bed

Meal Plan

Print Notice

OK

Cancel

Figure A2.10 The smoking status shows as a matching parameter, in roommate lists and in the room information on the Assignment and Moves working window.