

# 8

## Setting Up a Mini SIS

*It is important for an educational program to maintain a wide variety of data related to each of the students matriculating through the program. This data can range from personal contact information to information about a student's performance in classes. The data, in the case of many schools, might include a student's grade point average, his or her test scores, as well as contact information for parents or guardians and may even include things such as a student's blood type and information about allergies. A Student Information System (SIS) is a computer-based system that organizes all of this data, and more, and makes it easy to access for administrators and teachers. Moodle is not designed specifically to function as an SIS but as much of the data that is often included in an SIS already exists in the Moodle database, it's not that difficult to bend Moodle to meet our needs.*

In this chapter, we will explore two different methods of setting up Moodle to function as an SIS. The two methods we will explore are:

- ◆ Use of the custom user profile fields
- ◆ Use of the Xataface database 'backside' application

## Custom user profile fields

The ability to create custom user profile fields has existed since Moodle 1.8. This feature allows a site administrator to add fields that, when populated with information or data, can be assigned to chosen users in your system. For example, if you administer an assessment test in your program, you can upload students' test scores to profile fields. You can choose to allow students to see the field in their profile or can make it invisible so that only the site administrators will have access to the data. As you increase the amount of information that is incorporated into user profile fields, you will slowly be turning your Moodle site into a mini SIS. In the following explanation, we'll show you how to set up and utilize user profile fields and will also walk you through a couple of different methods of accessing the data from the perspective of an educational program administrator.

### Time for action – setting up a user profile field for assessment test scores

We will start by setting up three user profile fields, two will be used to house achievement scores from an assessment test given in our program twice a year, and the other will be used to record which of three programs a student is enrolled in. After setting up the fields, we will populate them with data and then demonstrate several methods of accessing aggregated data. To set up the profile fields, follow these steps:

1. Log in to your Moodle site as admin and click on the **User profile fields** link found by clicking on **Users** and then **Accounts** from within the **Site Administration** block, as shown in the following screenshot:



2. When you visit the **User profile fields** link, you will see a screen like the following screenshot. Use the drop-down menu to select **Text input**, as shown in the screenshot.

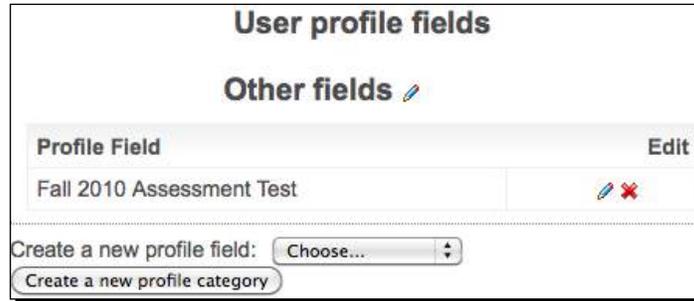


3. You will now see a window titled **Creating a new "Text input" profile field**. Enter a **Short name** and, although it is not stated in the window, use only lower case letters. Give the field a title using the **Name** text entry area. This is the name that will be displayed for this field. Enter a brief description for the field in the **Description** area. Change the **Is this field locked?** setting to **Yes** to prohibit users from changing the values and the **Who is this field visible to?** setting to **Visible to user** using the drop-down menus. At the bottom of the page, change the **Display size** and **Maximum length** settings to **15**, leave the **Default value** blank, and the **Is this a password field?** setting set to **No**. The following screenshot is of this screen with all settings, except the **Display size** and **Maximum length** settings. Click on **Save changes** to create the field.

 A screenshot of the 'Creating a new "Text input" profile field' form. The form is titled 'Creating a new "Text input" profile field' and has a 'Common settings' section. It contains the following fields and settings:
 

- Short name (must be unique)\*:** f10test
- Name\*:** Fall 2010 Assessment Test
- Description of the field:** Results from the fall 2010 assessment test.
- Rich text editor:** Includes a toolbar with options like font face (Trebuchet), size (1 (8 pt)), bold, italic, underline, strikethrough, and a text area containing the text 'Results from the fall 2010 assessment test.'
- Path:** body
- Is this field required?:** No
- Is this field locked?:** Yes
- Should the data be unique?:** No
- Display on signup page?:** No
- Who is this field visible to?:** Visible to user
- Category:** Other fields

4. You will now see the field you just created, as shown in the following screenshot:

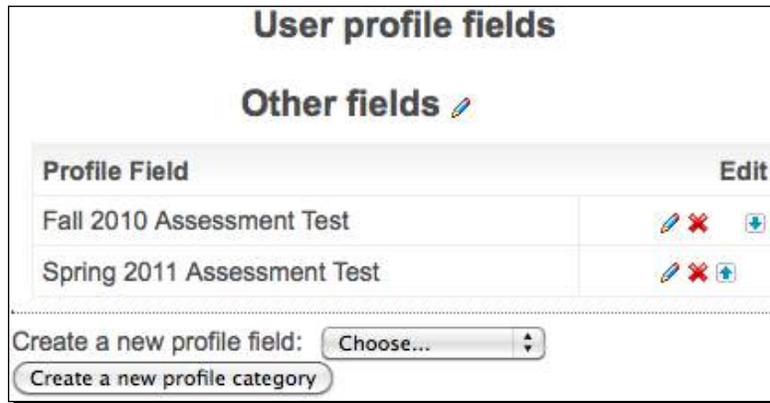


### ***What just happened?***

We have just created a field that will be used to house test scores associated with students in our program. We limited the length of the value stored in this field to 15 spaces, which, for a test score, is certainly more than sufficient. We have also locked the field so that it cannot be edited by the user and have set it so that only the user and site administrators can view the contents of the field.

### **Have a go hero**

Now, set up the next assessment test score field in the same fashion as we did for the first. Change the name to **Spring 2011 Assessment Test**. Once you have completed the creation process for the second field and have navigated back to the **User profile fields** window, you should see something like the following screenshot with both of the fields listed:



## Time for action – a profile field for program enrollment information

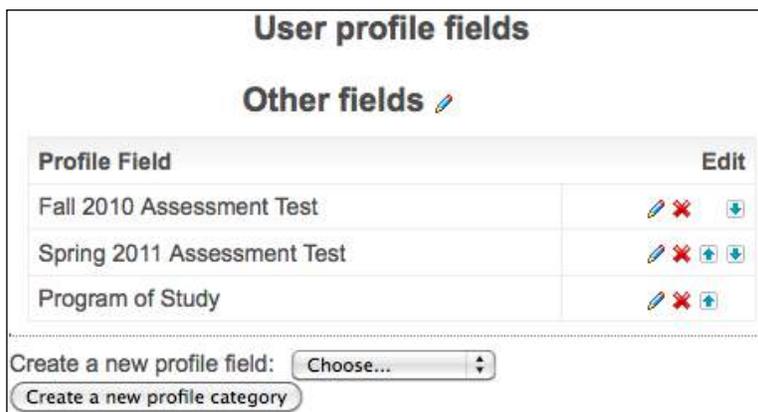
For this example, we'll set up one more profile field to house information about educational programs a student is enrolled in. Notice that in the example site we have constructed thus far, this field will contain information that is related to programs of study that students are enrolled in that is peripheral in terms of the language program focus our site has taken thus far. This illustrates how our SIS is going to start taking shape by enabling us to house information that is at a level beyond our immediate educational program. In this case, the example could be from a college or university. Let's use the following three programs for this example:

- ◆ Teacher training program
- ◆ Pre-law program
- ◆ Business management program

To set up the user profile field that will hold this information, follow these steps:

- 1.** Navigate to the **User profile fields** screen found in the **Site Administration** block under **Users|Accounts|User profile fields** from the front page of your site.
- 2.** Use the **Create a new profile field** drop-down menu to select **Text input**, the same way you did for the previous two fields.
- 3.** Give the field a short name such as **program** (remember to use all lowercase letters) and a **Name** such as **Program of Study**. Enter a short description and set the field as follows:
  - Not required
  - Locked
  - Not unique
  - Not displayed on the sign up page
  - Visible to everyone

4. Change the **Display size** and **Maximum length** to **20** and click on **Save changes** to create the field. Your **User profile fields** screen should now look like the following screenshot:



### ***What just happened?***

We have now created three user profile fields. Two will house assessment test scores and the third will contain information about programs that students are enrolled in. The first two fields have been set to be locked and invisible to everyone aside from each individual user and to the site administrators. The third program information field was also set to be locked but visible to everyone who can view user profiles on the site.

## **Time for action – populating the fields**

The data that will be stored in these fields will, in the case of most programs, exist in spreadsheets or other storage systems external to Moodle, so we will need to migrate the scores and program enrollment information into Moodle. This can be easily accomplished by using the **Upload users** function we visited earlier in *Chapter 3, Student Account Creation and Enrollment*. Follow these steps to prepare a CSV file and use it to populate the user profile fields:

1. Obtain a list of the users in your Moodle site in a spreadsheet format. While this can be accomplished in many different ways, the easiest for this example will be to use the CSV file we used in *Chapter 3, Student Account Creation and Enrollment*, to create new student accounts. The file is available from the Packt website at [www.packtpub.com/support](http://www.packtpub.com/support). In a production site environment, you could use the **Bulk user actions** function in Moodle found from the front page of your site in the **Site Administration** block under **Users | Accounts | Bulk user actions**, to download a list of students.

2. Retain the **username** column in your file and delete all other columns.
3. Create three new columns using the short name for the profile fields we created as column headings. The profile field short names must be preceded by `profile_field`. The columns for the fields from our example will have the following headers:
  - `profile_field_f10test`
  - `profile_field_s11test`
  - `profile_field_program`
4. Once you have added the column headings, add information for all of the desired users. For this example, you will have a file with four columns, as shown in the following screenshot. (Note, in a real world setting you will most likely have data associated with users via a name and possibly some type of ID number that will need to be matched to the username used in Moodle. For this reason, as was mentioned in a previous chapter, the use of student ID numbers as usernames can ease this process of matching data. For our example, the data shown next was simply generated randomly.)

	A	B	C	D
1	username	profile_field_f10test	profile_field_s11test	profile_field_program
2	S400001	98	94	Teacher Training
3	S400002	88	94	Pre Law
4	S400003	80	93	Business Management
5	S400004	100	97	Teacher Training
6	S400005	82	98	Pre Law
7	S400006	84	96	Business Management
8	S400007	82	93	Teacher Training

5. Save the file using the CSV format and navigate to the **Upload users** page found from the **Site Administration** block under **Users|Accounts|Upload users**.

- Use the **Browse** button to find and select your CSV file. Leave the **Preview rows** setting on **10** and, once you have selected your file, click on the **Upload users** button. You will see something like the following screenshot:

**Upload users preview**

username	profile_field_f10test	profile_field_s11test	profile_field_program
S400001	98	94	Teacher Training
S400002	88	94	Pre Law
S400003	80	93	Business Management
S400004	100	97	Teacher Training
S400005	82	98	Pre Law
S400006	84	96	Business Management
S400007	82	93	Teacher Training
S400008	91	98	Pre Law
S400009	100	94	Business Management
S400010	91	100	Teacher Training

Number of preprocessed records: 500

**Settings**

- Upload type: Update existing users only
- New user password: Field required in file
- Existing user details: Fill in missing from file and defaults
- Existing user password: No changes
- Allow renames: No
- Allow deletes: No
- Prevent email address duplicates: No
- Select for bulk operations: No

- Change the **Upload type** to **Update existing users only** and **Existing user details** to **Fill in missing from file and defaults** using the drop-down menus, as shown in the preceding screenshot. Leave all of the other settings on their defaults and click the **Upload users** button found at the bottom of the page. The **Upload users results** window, shown in the following screenshot, indicates a successful update but does not display the profile field information that was added. Click the **Continue** button found at the bottom of the screen.

**Upload users results**

Status	CSV line	ID	Username	First name	Surname	Email address	Password	Authentication	Enrolments	Delete
User updated	2	1065	S400001							
User updated	3	1066	S400002							
User updated	4	1067	S400003							
User updated	5	1068	S400004							
User updated	6	1069	S400005							

8. Check to make sure the fields have been added, select any user by using the **User Administration** or **Browse a list of users** tools. You will see that all users have had the three fields added to their profiles, as shown in the following screenshot:



### ***What just happened?***

We have just populated all of the user profile fields, created earlier, with data using the **Upload users** tool and then checked a user's profile to see how the information is displayed. Adding various types of information and data to customized user profile fields will allow you to begin to construct an SIS in which you can maintain information about the students in your educational program.

### **Pop quiz**

1. Which of the following two settings should be used when using a CSV file and the Upload users tool to populate user profile fields?
  - a. Add new users only, skip existing users, and Fill in missing from file and defaults.
  - b. Add new and update existing users and No changes.
  - c. Add all, append counter to usernames if needed, and Create password if needed.
  - d. Update existing users only and Fill in missing from file and defaults.

## Have a go hero

We used the **Upload users** function in Moodle to insert information into the user profile fields we created. This task can also be accomplished via the PHPMyAdmin tool. It is a slightly more involved process but just as quick and efficient if you know what you are doing. You will need to use a series of SQL statements using `INSERT INTO` and `WHERE`. This method is handy if you created a short name for your profile field that includes capital letters and, for whatever reason, you do not want to change the name to lowercase because Moodle will generate an error if you use capital letters in the profile field column headings when attempting to update user information. Add a new profile field using the process outlined earlier in this chapter and then see if you can populate it using PHPMyAdmin and some basic SQL statements. You'll need to know the ID number used by the profile field you create, which can be determined by viewing the table with PHPMyAdmin. If we create a new field called `Counselor`, one SQL statement, designed to insert the counselor's name, might look like the following:

```
INSERT INTO mdl_user_info_data (data, fieldid, userid) VALUES
('Smith', 8, 49);
```

In this case, we are inserting the value `Smith` into the `counselor` field, which has an ID number of 8, for the user with the ID number of 49. The ID numbers used in the database can be found by browsing the following two tables with PhpMyAdmin:

- ◆ `mdl_user` (for the user's ID number)
- ◆ `mdl_user_info_field` (for the profile field ID number)

## Using the user profile fields as the search criteria

Now that we have created and populated our customized user profile fields, how can we use the information as criteria for searching for groups of students? There are various methods for viewing the information entered into the user profile fields. One very simple method is to search for a student and view his or her profile to see the information stored in the fields. If however, we would like to produce a list of all students who are enrolled in the Pre-law program, for example, this method is not useful. The following are two methods of accessing lists of users based upon information or values stored in the user profile fields.

## Time for action – searching using the Bulk user actions tool

To use the **Bulk user actions** tool to search for a group of students using the criteria based on values or information stored in the user profile fields, follow these steps:

1. Navigate to the **Bulk user actions** page by clicking on the **Bulk user actions** link found under **Users** and then **Accounts** from within the **Site Administration** block.

- Click on the **Show Advanced** button under the **New filter** heading, shown in the following screenshot, to display the advanced settings for the filter:

- With the advanced options displayed, you will see sixteen different criteria that you can use to search for users in your system. For this example, we are going to be using the **Profile** fields so use the pull-down menu to change **any field** to **program** and then the search criteria to **is equal to** and type `Pre Law` into the text entry box. The filter settings should look like the following screenshot:

- Scroll to the bottom of the **Filter** heading and click on the **Add filter** button.
- On the resulting screen, you will see a new heading, with the title **Active filters**, below the **New filter** heading. Below the **Active filters** heading, you will see a heading titled **Users in list** that will display the results of the filter that was just applied. In our example, there were 167 users with Pre-law in their **program** profile field, as shown in the following screenshot:

6. Click on the **Add all** button found at the bottom of the **Users in list** heading to move the users from the **Available** display window to the **Selected** window.
7. For this example, we are going to download this list of users. Use the pull-down menu next to the **With selected users** heading to select **Download** and then click the **Go** button, as shown in the following screenshot:



8. From the subsequent **Download** window, shown in the following screenshot, select the format you wish to download the information in, by clicking on the desired link. We will use Excel for this example. This will initiate the download, which, depending on your computer system and settings, may also present a popup that allows you to approve the download and select its location.



9. Once you have downloaded the file, click on the **Continue** button found at the bottom of the screen to return to the **Bulk user actions** screen.
10. Open the file that was downloaded to view the users. Notice that all of the profile fields are included in the information downloaded.

### ***What just happened?***

Using the `Bulk user actions` function built into Moodle, we searched for, and downloaded, a list of all the students enrolled in the Pre-law program. As mentioned earlier, the more information that is entered into user profile fields, the more possibilities exist in terms of sorting, or filtering, for groups of students within your educational program.

### **Have a go hero**

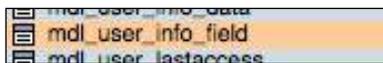
If you want to use the **Bulk user actions** tool to filter users based upon numeric values, such as test scores, that have been entered into custom user profile fields, you will notice that the conditions upon which you can filter are somewhat limited. I have added several mathematical operators to the list of conditions for user profile fields that can be downloaded and added to

your Moodle instance by downloading the modified files or the patch file from <http://tracker.moodle.org/browse/MDL-21979>. Either apply the patch or use the two modified files to find the lines that were added and add them to your core Moodle files. You can use the search string "added" to find the lines of PHP code that were added in each of the two files. Four lines were added to the `filters.php` language file and 12 lines were added to the `profilefield.php` file located at `/moodleroot/user/filters`. Once you have added the mathematical operators, try searching for all users with a **Fall 2010 Assessment Test Score**, using the `f10test` profile field short name, of 80 or higher.

## Time for action – searching using PHPMyAdmin

In most cases, the filters that can be applied via the **Bulk user actions** tool are sufficient to allow you to obtain the students you are searching for in your system. In some cases however, it is much more efficient to search for the information directly using the PHPMyAdmin tool. One such case is when you have a large number of users in your system (several thousand or more) and wish to search using a numeric condition, such as greater than or equal to, using the profile fields. The Moodle core, as of 1.9.9+, does not perform this operation as efficiently as it should and executing such a command via the **Bulk user actions** tool can put a heavy enough load on your MySQL to cause it to freeze up. The Moodle development team is aware of this issue and it is scheduled to be fixed but it may take some time for it to be remedied as it is a minor issue compared to many of the other development projects they are working on. PHPMyAdmin can be used to perform such tasks, usually without over burdening the MySQL server. To search for a list of students who, for example, all scored over an 80 on the fall assessment test, follow these steps:

1. Log in to your site as admin and, from the front page of your site, access the PHPMyAdmin tool by clicking on **Server** and then **Database** from within the **Site Administration** block.
2. Scroll down the table list on the left side of the screen to the table titled `mdl_user_info_field` and click once on this table link. The table link is shown in the following screenshot:



3. The default display for the table, which you will see to the right of the table list in the main portion of your browser window is the **Structure** display. Click on the **Browse** tab found at the top-left of the main window frame, as shown in the following screenshot:



4. Scroll down to view the details of the three user fields that we have created so far. Note that the **id** for the field that we are using to store fall assessment test scores is **4**, as shown in the following partial screenshot:

			id	shortname	name	datatype	description	categ
<input type="checkbox"/>			4	f10test	Fall 2010 Assessment Test	text	Results from the fall 2010 assessment test.	
<input type="checkbox"/>			5	s11test	Spring 2011 Assessment Test	text	Results from the spring 2011	

5. Scroll to the top of the screen and click on the **SQL** tab found two tabs to the right of the **Browse** tab.
6. Delete the default **SELECT** statement found in the **Run SQL query** window and enter the following SQL query:

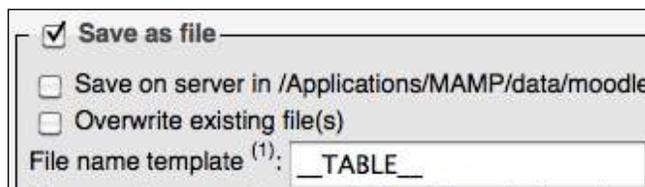
```
SELECT mdl_user.username AS "Username", mdl_user.firstname AS
"First Name", mdl_user.lastname AS "Last Name", mdl_user_info_
data.data AS "Fall 2010 Assessment Test"
FROM mdl_user
INNER JOIN mdl_user_info_data ON mdl_user.id = mdl_user_info_data.
userid
WHERE mdl_user_info_data.data >=80
AND mdl_user_info_data.fieldid =4;
```

7. This query will return a list of all students who scored 80 or higher on the fall assessment test, as shown in the following partial screenshot:

Username	First Name	Last Name	Fall 2010 Assessment Test
s400002	Student4002	Advanced002	88
s400001	Student4001	Advanced001	98
s400003	Student4003	Advanced003	80
s400004	Student4004	Advanced004	100
s400005	Student4005	Advanced005	82

8. Scroll down to the **Query results operations** heading, shown in the following screenshot, and click on **Export**.

9. Select the download format desired and make sure to tick the **Save as file** box, shown in the following screenshot. The default filename will be **\_\_TABLE\_\_**, so change it to an appropriate self-explanatory name and click on the **Go** button to export the results:



The screenshot shows a form with the following elements:

- Save as file
- Save on server in /Applications/MAMP/data/moodle
- Overwrite existing file(s)
- File name template <sup>(1)</sup>:

### ***What just happened?***

We just ran a very simple SQL query that allowed us to retrieve users from the Moodle database who scored an 80 or higher on the fall assessment test. This SQL query returns basically the same information that we filtered for with the **Bulk user actions** tool but can be, in the case of large sites with many users, much more time-efficient. The downside to the query is that it is difficult to retrieve information stored in other profile fields when using a condition based upon a test score such as this one.

## **The Xataface database 'backside' application**

For many tasks related to maintaining student information, working through the Moodle interface is sufficient. If you wish to maintain data that is not used directly by Moodle however, and you also want to provide varying levels of access to that data to selected personnel in your educational program, you may wish to use the Xataface application that we set up in the previous chapter. Use of this application can enable you to set up a student information system that is both tightly integrated with Moodle and at the same time a system that operates in parallel with but independent of Moodle. Because Xataface is in simple terms a browser-based user interface for a database, its possibilities in conjunction with Moodle are virtually limitless. We will explore a few of those possibilities and continue to customize the application we have already set up in the remainder of this chapter. We'll introduce and explain the following:

- ◆ Customizing the Xataface table display
- ◆ Setting up relationships in Xataface
- ◆ Using Xataface to modify core Moodle data
- ◆ Xataface as an enrollment table maintenance tool

## Time for action – customizing the Xataface table display

Xataface is designed to be extremely customizable from its appearance to how it functions. A thorough exploration of Xataface is beyond the scope of this book but we will go through a few of the basics in order that you may get it set up and working efficiently with in your Moodle instance. We'll start by building on the minor display customization we did in the previous chapter. Our current Xataface application is set to display only the user's table in Moodle but we don't need to see all of the fields in that table, so let's hide all but the fields we want to have access to. Follow these simple steps to clean up the table display:

1. Log in to your Xataface application at `www.yourmoodlesite/cims` and click on the **Users** tab to view the user's table.
2. Leave your browser for now and navigate to your Xataface directory, which should be located in your Moodle root directory and titled `cims`, if you have followed all of the tutorials thus far.
3. Navigate to the `tables` directory found inside the `cims` directory. There should be one directory inside the `tables` directory, the `dataface__users` directory.
4. Create a directory in the `tables` directory titled `mdl_user`. This directory represents the user's table in Moodle.
5. Open your text editor and type the following for each field in the user's table that you do not want to have displayed through Xataface:

```
[id]
visibility:browse=hidden
visibility:list=hidden
visibility:find=hidden
```
6. The preceding example starts with the ID field used in the user's table. Save the file with the `fields.ini` filename inside the `mdl_user` directory.
7. Go ahead and type the code from step 5, save the file and go back to your browser and refresh the page. Notice that the ID field, on the far left of the screen, disappears when you refresh the screen. This is because we have told Xataface to hide this field from display.
8. Return to the `fields.ini` file you saved in the `mdl_user` directory and type the code from step 5 for each field in the table. Copy the three visibility statements and then just paste them after typing in each field name enclosed in brackets to be more efficient. For this tutorial, let's keep only the following fields: `username`, `firstname`, and `lastname`. It's a little tedious to enter the code for all of the fields that you don't want to have displayed, but you only have to do it once. Format the file, as shown in the following screenshot, to make it easy to read later:

```

[id]
visibility:browse=hidden
visibility:list=hidden
visibility:find=hidden

[auth]
visibility:browse=hidden
visibility:list=hidden
visibility:find=hidden

[confirmed]
visibility:browse=hidden
visibility:list=hidden
visibility:find=hidden

```

9. When you've finished entering all of the information into the `fields.ini` file, save the file and refresh your browser window. Your **Users** table display should now look like the following screenshot:

	username	firstname	lastname
<input type="checkbox"/>	guest	Guest User	
<input type="checkbox"/>	admin	- Admin	-
<input type="checkbox"/>	teacher1	Teacher	One
<input type="checkbox"/>	teacher2	Teacher	Two
<input type="checkbox"/>	teacher3	Teacher	Three
<input type="checkbox"/>	teacher4	Teacher	Four
<input type="checkbox"/>	teacher5	Teacher	Five

10. Now, before we finish this task, click on the name of one of the users, **teacher5** for example, under the **username** heading to show the **details** view for that user. In the `fields.ini` file, this view is controlled via the **browse** setting.
11. From your Moodle site, find the user you have selected in the previous step using the **User Administration** block, edit the profile and enter a short description in the description field window. For this example, we are using **Teacher Five** so we'll enter "Hi, my name is Teacher Five."

12. Go back to your `fields.ini` file and remove the `visibility:browse` line for the **description** field. Save the file and refresh your browser window. The description line will appear in the details window with the description we just added, as shown in the screenshot displayed after step 15.
13. And, for one last customization, open a new text file with your text editor and type the following code:

```
<?php
class tables_md1_user {
function getTitle(&$record){
return $record->display(<firstname>) . « « . $record-
>display(<lastname>);
} }
?>
```
14. Save the file with the filename `mdl_user.php` in the `mdl_user` directory that is located in your `tables` directory inside your `cims` directory.
15. Refresh your browser window and notice that the **Current Record** title changes from **manual** to the name of the user, as shown in the following screenshot:



## What just happened?

We have just refined our Xataface application by hiding most of the fields in the Moodle user table so that only the username, first name, and last name fields appear in the list view. We then changed one setting to display the **Description** field in the details, or browsed, viewed and finally modified the settings for the table view via a simple PHP file in order to change the record title to the name of the user. These are only a few of the many modifications that can be made in Xataface. Visit the Xataface website forums or documentation wiki found at [www.xataface.com](http://www.xataface.com) for more tips and explanations.

## Time for action – setting up relationships in Xataface

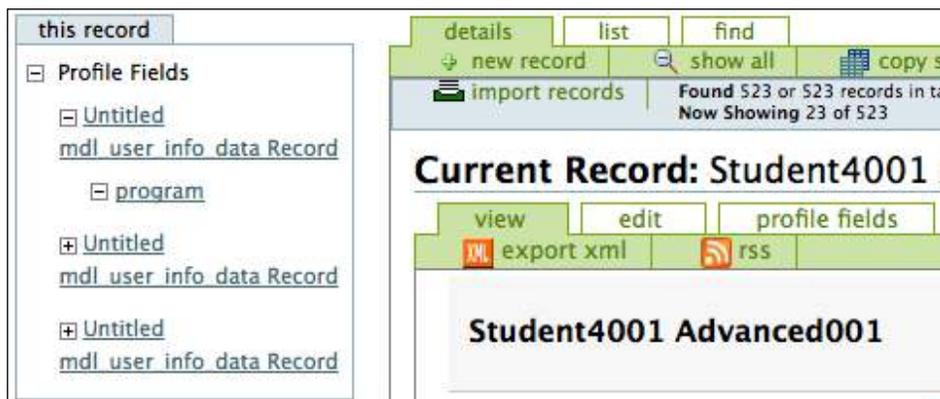
Next, let's set up some relationships in Xataface so that we can see all of the values in the profile fields when we click on a user from the list view to view the details window. This will involve setting up several relationships, but once we are finished we'll have a nice interface that allows us to toggle through the profile field information. Follow these steps to get it set up:

1. In the `tables` directory located in your Xataface `cims` directory, create the following two directories:
  - `mdl_user_info_data`
  - `mdl_user_field`
2. These represent the two tables that contain profile field information.
3. In the `conf.ini` file, located in the Xataface root, titled `cims`, remove the two lines under the `disallowed_tables` list for the `mdl_user_info_data` and `mdl_user_field` tables and save the `conf.ini` file.
4. With your text editor, create a file that contains the following code:
 

```
[Profile Fields]
mdl_user_info_data.userid = mdl_user.id
```
5. Then save the file with the filename `relationships.ini` in your `mdl_user` directory located in the `tables` directory of your Xataface application.
6. Again, with your text editor, create a new file and type the following code:
 

```
[Profile Field Titles]
mdl_user_info_data.fieldid = mdl_user_info_field.id
```
7. Save the file with the filename `relationships.ini` in your `mdl_user_info_data` directory located in the `tables` directory of your Xataface application.

8. Once you have created and saved all of these files, navigate to your Xataface application, log in as admin, and view one of the student accounts from the list view of the **Users** table. We will use Student4001 for this example.
9. In the upper-left portion of the window, you will see a block titled **this record**. Click on the plus sign next to the **Profile Fields** heading to expand the window and display the profile field records. Click on the plus sign next to one of the fields to further expand that field. You should see a display similar to the one shown in the following screenshot:



10. This display isn't very helpful because Xataface is not displaying a title for each record. We can remedy this easily by adding two simple PHP files, one to each of the table directories. In the `mdl_user_info_data` directory, located in the `tables` directory, create a file with the following contents:

```
<?php
class tables_mdl_user_info_data {
function getTitle(&$record){
return $record->display(<data>);
} }
?>
```

11. Then save the file as `mdl_user_info_data.php`.

12. Create another file with the following contents:

```
<?php
class tables_mdl_user_info_field {
function getTitle(&$record){
return $record->display(<name>);
} }
?>
```

13. Then save the file as `mdl_user_info_field.php` in the `mdl_user_field` directory.
14. After creating and saving the files in their appropriate locations, refresh your browser window and notice that you now have field information followed by field titles, as shown in the following screenshot:



### ***What just happened?***

We just added relationships to the directories of three tables in order to be able to view the profile fields associated with a user when viewing the details window for a user. We also added some very simple PHP files in order to change the field used for the record title. This was a very simple introduction to how Xataface can be set up to display data from multiple tables used by Moodle.

### **Have a go hero**

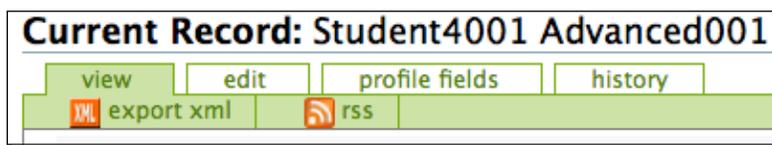
As I mentioned earlier, there are a wealth of customization possibilities with Xataface. Try and change the settings that we just walked through and have the profile fields displayed within each user's details view. If you succeed, remove the **this record** box that is displayed on the left side of the window as you won't need it. For help with this, visit the Xataface forums; specifically, the following forum thread, where I worked through this exact process together with the Xataface head developer: <http://xataface.com/forum/viewtopic.php?f=4&t=4765>.

## Time for action – using Xataface to modify core Moodle data

Now that we have worked a little on modifying the display settings in Xataface, we should briefly discuss and experiment with actually making changes to a Moodle table via the Xataface interface. Remember, we are going to be making changes to data that is used by Moodle, so don't practice on a live site. The experimental site we have set up is the perfect environment to get some hands on experience in order to get used to the tools and processes and to also give you some time to think about how, and if, you want to use them. We'll start by enabling the history function within Xataface and then walk through editing some fields in the Moodle `mdl_user` table. Follow these steps:

1. First, to enable the history function, open the `conf.ini` file located in your Xataface root directory, titled `cims` in our case, and add the following two lines below the `[_auth]` heading and before the `[_allowed_tables]` heading:  

```
[history]
enabled = 1
```
2. Save the file.
3. Log in to your Xataface application located at `www.yourmoodlesite/cims`, click on the **Users** tab and then on a single user to view the details window. We'll use **Student4001** for this example.
4. Notice the **history** tab that is now displayed, as shown in the following screenshot. Clicking on the **history** tab will result in a message that there is no history yet. This is because we have not yet edited a record via the Xataface interface.



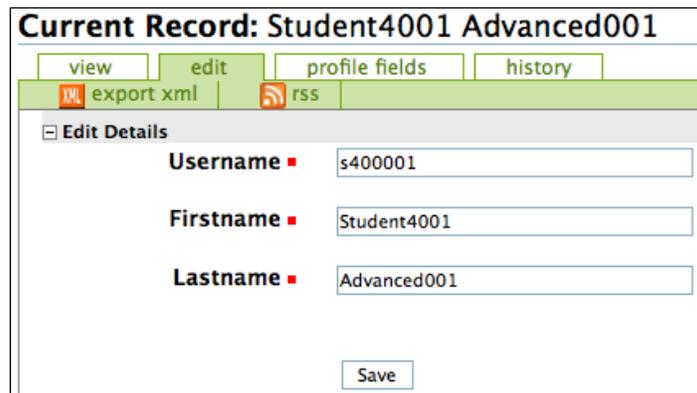
5. Click on the **edit** tab to open the editing window for the selected user. Note that all of the fields for the `mdl_user` table appear in the **editing** window. We need to make some setting changes to make sure that the fields we hid via the `fields.ini` file are not editable via the Xataface interface.
6. Open the `fields.ini` file located in the `mdl_user` directory, which is in the `tables` directory. Below each table heading, add the following line for each table:  

```
widget:type=hidden
```

as shown in the following screenshot. This will completely prevent access to the field. We could have used this setting earlier when we hid the fields but waiting until now allowed for the introduction of both the **visibility** and **widget** settings.

```
[auth]
widget:type=hidden
visibility:browse=hidden
visibility:list=hidden
visibility:find=hidden
```

7. Refresh the browser window and note that you are now presented with only three fields in the editing window for the user's table, as shown in the following screenshot:



**Current Record: Student4001 Advanced001**

view edit profile fields history

export xml rss

Edit Details

**Username** ■ s400001

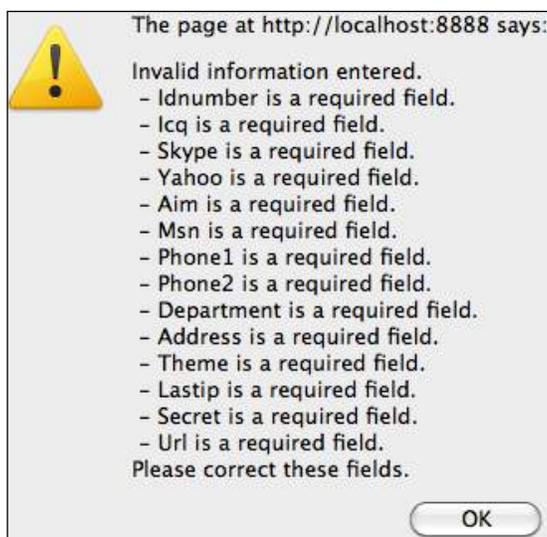
**Firstname** ■ Student4001

**Lastname** ■ Advanced001

Save

8. Change the **Firstname** to Student Four Thousand And One and click on the **Save** button.

9. Here you will get an error stating that invalid information has been entered for a long list of fields. This is because many of the fields in the `mdl_user` table have been created, by Moodle, with the **Null** setting set to **NO** but are then allowed to be left empty by Moodle. When Xataface opens the table for editing it is in a sense opening all of the fields in a row, even the ones we have hidden, and attempting to save them with the values they contain. For the empty fields, this is generating an error because the database setting is not allowing them to be empty. To correct this, you will need to use PHPMysqlAdmin, by clicking on **Server** and then **Database** from within the **Site Administration** block from the front page of your Moodle site, to view the table and change the field **Null** settings, on the `mdl_user` table, from **not null** to **null**. Make a note of the fields listed in the error message, as shown in the following screenshot, and change the setting for these fields. Note, the list of fields that need to be set to null may differ depending upon how you populated the user profile fields. For example, I have entered an institution name for all users so, even though that field is not allowed to be null, I am not getting an error because all fields contain information and are thus not null.



10. Once you have changed all of the null settings, try again to change the name of the user, to *Student Four Thousand And One*, via the editing page. You will see a message at the top of the window indicating that the record was saved successfully.
11. Find this student from within Moodle using the **User Administration** block with the search string **Four Thousand**. Click on the name to view the profile and note that the name change is, naturally, visible via the Moodle interface, as shown in the following screenshot:



- 12.** Go back to the Xataface application and click on the **history** tab to view the history information that is now available. The **history** tab allows you to view changes made and revert to a previous setting. This is only possible however, for changes made via the Xataface interface.

### ***What just happened?***

We have just briefly introduced how to make changes to information in the Moodle database with the Xataface application. We started by enabling the history function in Xataface and then by hiding fields that we did not want to be edited and then made changes to some of the fields in the `mdl_user` table in order to allow Xataface to save rows to the table with empty, or null, values. After making all of the necessary changes, we experimented with a name change and observed the change via both Xataface and Moodle. We also confirmed that the history function in Xataface maintains a record of changes made via the tool.

### **Pop quiz**

1. How can data modified via the Xataface application easily be monitored and fixed in the case of mistakes?
  - a. By making sure that users who are given access to the Xataface application make notes of all the changes they make.
  - b. By turning on the history function in Xataface that will record, in the database, a record of all changes made.
  - c. Once changes have been made to the database, they cannot be reversed so it is important to be careful when working with Xataface.
  - d. Records can be downloaded to CSV files daily and kept as backup copies to reference when a mistake is suspected.
2. What important change must be made to the Moodle database to allow Xataface to change only selected fields in the `mdl_user` table?
  - a. Fields that you don't want to use must be deleted from the Moodle database.
  - b. The settings in the database for fields that you will not access via Xataface must be set such that they are allowed to be `NULL`.

- c. The Xataface tool must be set to display on the fields that you want users to be able to change.
- d. The values that you want to control should be changed via the PhpMyAdmin tool.

## Time for action – Xataface as an enrollment table maintenance tool

One last task Xataface can be used for, that we'll explore in this book, is as a tool to maintain a table in the database to be used by the Moodle enrollment tool. In *Chapter 3, Student Account Creation and Enrollment*, we explored the various enrollment options available in Moodle and referred forward to this chapter in stating that we would explore the external database enrollment plugin here. Actually, what I propose is to use the Xataface tool to maintain a table, added to the Moodle database, that will be used by the external database plugin. As such, it won't be an external database but Moodle doesn't care. To set up the table, follow these steps:

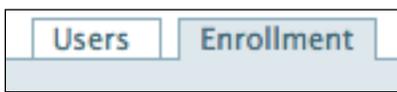
1. Navigate to the Enrolments settings window of your Moodle site from the **Site Administration** block by clicking the **Courses** link and then on the **Enrolments** link.
2. Click on the **Edit** link for the **External Database** settings. For the **External Database Server Settings**, shown in the following screenshot, use the database settings from your `config.php` file found in your Moodle root. For the **enrol\_dtable** field, use the name of the table you intend to use for the enrollment plugin. For this example, I am using **enrolltable**.

External Database Server Settings	
enrol_dbtype: <input type="text" value="mysql"/>	Database type
enrol_dbhost: <input type="text" value="localhost"/>	Server IP name or number
enrol_dbuser: <input type="text" value="moodle"/>	Server user
enrol_dbpass: <input type="password" value="*****"/>	Server password
enrol_dbname: <input type="text" value="moodle19"/>	Database name
enrol_dtable: <input type="text" value="enrolltable"/>	Database table

3. Scroll down and fill the remaining fields with the field values you intend to use. For this example, I will use the following fields and settings:
  - ❑ enrol\_localcoursefield: shortname
  - ❑ enrol\_localuserfield: idnumber
  - ❑ enrol\_db\_localrolefield: role
  - ❑ enrol\_remotecoursefield: course
  - ❑ enrol\_remoteuserfield: enrolled
  - ❑ enrol\_db\_remoterolefield: role
  - ❑ enrol\_db\_defaultcourseroleid: Student (Use pull down menu)
  - ❑ enrol\_db\_autocreate: No (Use pull down menu)
  - ❑ enrol\_db\_category: Ignore this if the above setting is No
  - ❑ enrol\_db\_template: leave blank
  - ❑ enrol\_db\_ignorehiddencourse: NO
  - ❑ enrol\_db\_disableunenrole: No
4. Click **Save changes** after all of the settings have been entered.
5. Use PHPMyAdmin, click on **Database** from the **Site Administration** block under **Server**, to create a new table called **enrolltable**. Create the table with four fields. Use the following:
  - ❑ id – BIGINT – 10 – Not Null – Auto Increment – Key
  - ❑ userid – VARCHAR – 20 – utf8\_general\_ci – Not Null
  - ❑ role – VARCHAR – 20 – utf8\_general\_ci – Not Null
  - ❑ course – VARCHAR – 50 – utf8\_general\_ci – Not Null
6. Save the table.
7. Add the table to the list of tables in your Xataface application by adding the following line to your `config.ini` file found in your Xataface root directory, titled `cims` in our case:

```
enrolltable = "Enrollment"
```
8. Save the file and log in to your Xataface application found at `www.yourmoodlesite.com/cims` as admin.

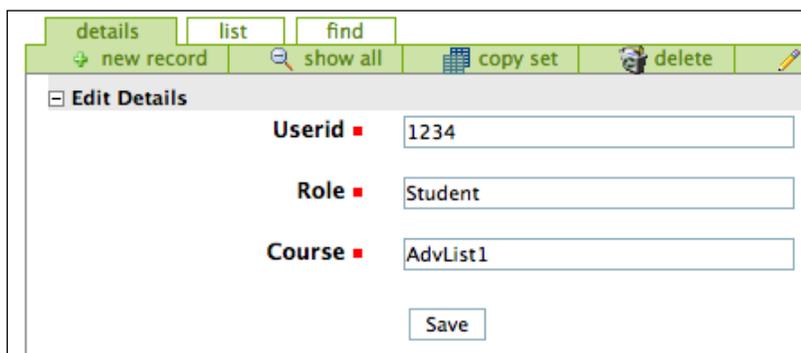
9. You will see the **Enrollment** tab, as shown in the following screenshot, but there will be no records in the table yet.



10. Click on the new record link, shown in the following screenshot, to add a new record to the table:



11. Use 1234 for the **Userid** field, Student for the **Role** field, and AdvList1 for the **Course** field, as shown in the following screenshot:



details list find  
new record show all copy set delete

**Edit Details**

**Userid** ■ 1234

**Role** ■ Student

**Course** ■ AdvList1

Save

12. From your Moodle site, use the **User Administration** block to locate a user, we will use **Teacher Twenty** for this example. Click on the username and then on **edit profile** to edit Teacher Twenty's profile.
13. Click on **Show advanced** to show the **Optional** profile fields and add **1234** as the **ID Number**. Click **Update profile** to make the changes.
14. Finally, go back to the **Enrolments** window found by clicking on **Courses** and then **Enrolments** from the **Site Administration** block and click once on the tickbox under **Enable** and then on **Save changes** to enable the **External Database** plugin.

15. Now, log out of your Moodle site and back in as Teacher Twenty with the username: **teacher20** and password: **1234**. Teacher Twenty is immediately, upon logging in, enrolled in the AdvList1 course as a student as evidenced by the appearance of the **AdvList1** course in the **My Courses** block.

### ***What just happened?***

We just set up a table in the Moodle database to be used by the external database enrollment plugin. We entered all of the appropriate settings to allow the external database plugin to access the table we created and match values with values in the Moodle tables associated with course enrollments. We then changed the settings in our Xataface application to allow it to access the table and created a record in the table. Finally, we added an ID Number to a user, a teacher to match the number used in the record created in the enrollment table, and then logged in as that teacher to confirm that the plugin would enroll the teacher in the specified course, and was thus functioning correctly.

### **Have a go hero**

The use of this external database enrollment setup can enable you to provide enrollment control to an education administrator without the need to allow them access to the Moodle system. A feature of Xataface, that unfortunately cannot be covered here, is its ability to accommodate customized import filters, which allow you to upload records to a table in bulk. Explore the Xataface forums and documentation that can be found from the Xataface website located at [www.xataface.com](http://www.xataface.com) and try to set up an import filter that will allow you to upload CSV, Excel, or other file formats into the enrollment table.

## **Summary**

In this chapter, we started by exploring how to set up an SIS via the use of user profile fields. We set up three fields and populated each with information and data for all of our student accounts and then explored various techniques for aggregating the information. In the second half of the chapter, we explored further customization of the Xataface database administration tool and demonstrated how Xataface can be used to modify information stored in tables in the Moodle database. We finished up with a basic demonstration of how the Xataface tool can also be used to function as an interface for a table in the Moodle database that is set up to function as a table used by the external database enrollment plugin. With some experimentation and imagination, the Xataface tool can serve as an important tool, that functions to greatly expand the CIMS capabilities of your Moodle site, by providing a useful backside access portal for educational administrators working in your program.