

CHAPTER 2

WIZARDS AND FILES

CHAPTER OBJECTIVES

In this Chapter, you will learn about:

- ✓ The Data Block and Layout Wizards Page 32
- ✓ Oracle Forms Files Page 53

It is now time to try your hand at creating some simple applications. The Form Builder has a vast and versatile interface that you will continue to uncover throughout the rest of this book and throughout your relationship with the tool. Lab 2.1 will bring you through the wizards that provide an extremely friendly environment in which to start your acquaintance with the Form Builder.

Once you have used the wizards to create some simple applications, you will want to know what to do with the files that store them. Lab 2.2 will lead you through some of the files you will encounter when using the Form Builder and illustrate how you can run and compile them.

LAB 2.1

THE DATA BLOCK AND LAYOUT WIZARDS

LAB OBJECTIVES

After this Lab, you will be able to:

- Use the Data Block and Layout Wizards
- Reenter the Wizards

Having sat patiently through two introductory Labs, you are now ready to start creating objects and simple applications. Despite your limited experience with the Form Builder interface, you will be able to quickly and easily create a block and assign its items to a canvas by employing the Data Block and Layout Wizards.

The initial part of this Lab will walk you through the pages of the wizards. This will be interactive and will require that you read and work in the Form Builder simultaneously. The Exercises will push you a bit further in using the wizards and test the skills you learned in the Lab.



Throughout this Lab and the Exercises, try to think about the concepts you have already learned. What are the mandatory Forms objects that you are creating with the wizards? Can you see how they are related? Can you identify possible events in your new applications and anticipate which triggers you might use to respond to them?

THE WIZARDS

The Data Block Wizard and Layout Wizard are usually used in sequence, meaning you walk through the Data Block Wizard to create a block, then you go straight to the Layout Wizard to create a canvas and frame.

Each screen in a wizard is referred to as a page. The creators of Oracle Forms have given each page a specific name, and it is these names that will be used here. Each page will be described briefly in the Lab and will be accompanied by a screenshot. You should have the Form Builder open and follow along with each step in the Lab.

You will start by using the Data Block Wizard to create a block based on the `STUDENT` table.



Each wizard has a Welcome! page. These will not be discussed in the Lab.

To set the environment, take the following steps:

- 1) Open the Form Builder.
- 2) From the Main Menu, select `File | New | Form` to create a new form.
- 3) From the Main Menu, select `File | Connect` to connect to the database.
- 4) From the Main Menu, select `Tools | Data Block Wizard` to open the Data Block Wizard.

THE DATA BLOCK WIZARD

The Data Block Wizard has three pages:

- 1) Type page (Figure 2.1).
- 2) Table page (Figure 2.2).
- 3) Finish page.



Figure 2.1 ■ The Data Block Wizard's type page.

TYPE PAGE

Here you have one task—to select the type of database object your block will be based upon. Throughout most of this book, you will base your blocks on tables.

So, here you should leave this page set to `Table or View` (the default) and click the `Next` button.



Figure 2.2 ■ The Data Block Wizard's table page.



Figure 2.3 ■ The Database Wizard's Tables window.

TABLE PAGE

Here you have three tasks:

- 1) Choose the base table.
- 2) Select the database columns to include as block items.
- 3) Decide whether or not to enforce data integrity.

There are two ways to choose the base table. You can either type the table name directly into the `Table` or `view` field, or you can click the `Browse` button to choose the table name from a list. If you click `Browse`, Forms will peer into the database's data dictionary and present you with a `Tables` window with a list of tables and views like the one in Figure 2.3.

Double-click the `STUDENT` table to select it. This brings you back to the table page, which will now look like Figure 2.4.

Now you are on to the second task on this page, which is to select the database columns to include as items in this block. You do not have to include all of them, but it is wise to always include the primary-key columns and foreign-key columns. To select columns, simply move them from the `Available Columns` text list to the `Database Items` text list. There are two ways to do this:

- 1) Use the arrow buttons positioned between the two text lists.
- 2) Double-click individual items.

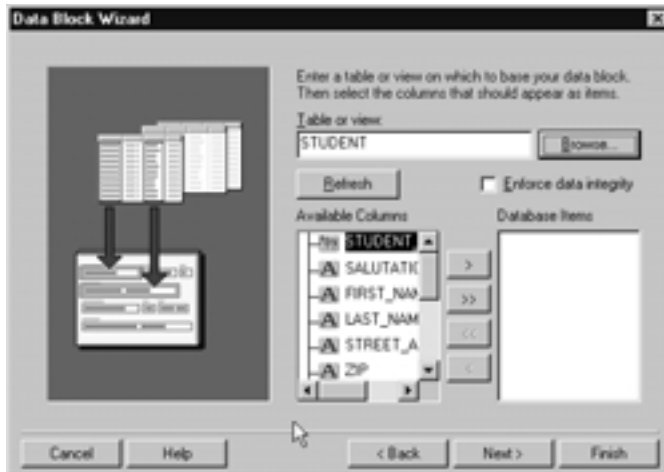


Figure 2.4 ■ The Database Wizard’s table page with the STUDENT table selected.

Windows multi-select functions also work here so that you can move more than one column at a time. Move all of the columns in the STUDENT table to the Database Items list.

Check Enforce data integrity for your STUDENT block. The table page should now resemble Figure 2.5. Click the Next button to move on.

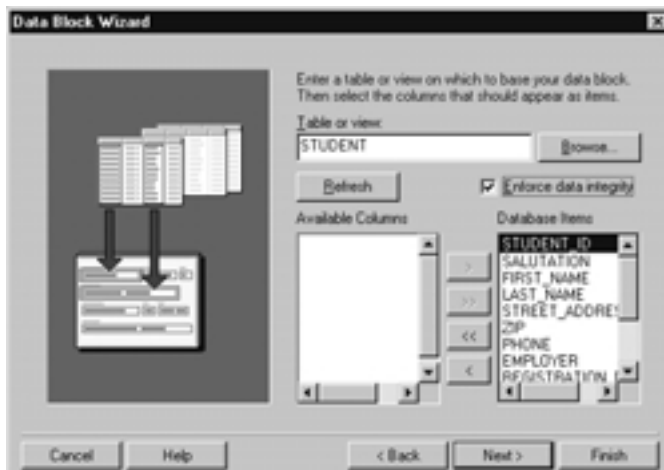


Figure 2.5 ■ Completed table page.

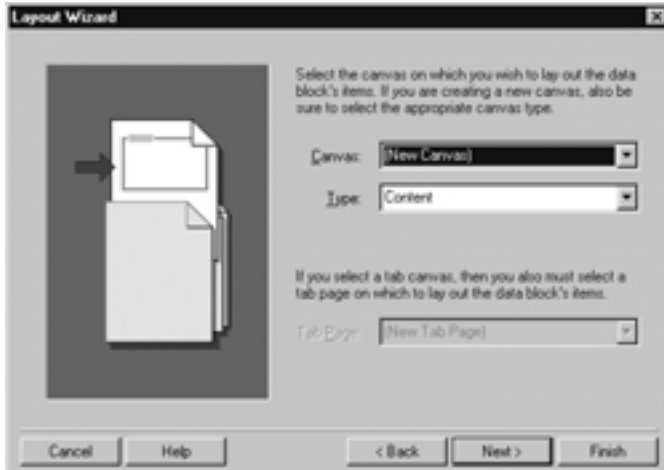


Figure 2.6 ■ The Layout Wizard's canvas page.

FINISH PAGE

This page is self-explanatory, so no screenshot has been included. All you have to do is decide whether or not to continue on to the Layout Wizard.

Select the `Create the Block` and then `Call the Layout Wizard` radio button (the default) and click the `Finish` button to move on to the Layout Wizard.

THE LAYOUT WIZARD

The Layout Wizard has six pages:

- 1) Canvas page (Figure 2.6).
- 2) Data block page (Figure 2.7).
- 3) Items page (Figure 2.8).
- 4) Style page.
- 5) Records page (Figure 2.9).
- 6) Finish page.

CANVAS PAGE

Here you have two or three tasks that you will accomplish by making selections from three list items:

- 1) Choose the canvas.
- 2) Choose the canvas type.
- 3) Choose the tab page.

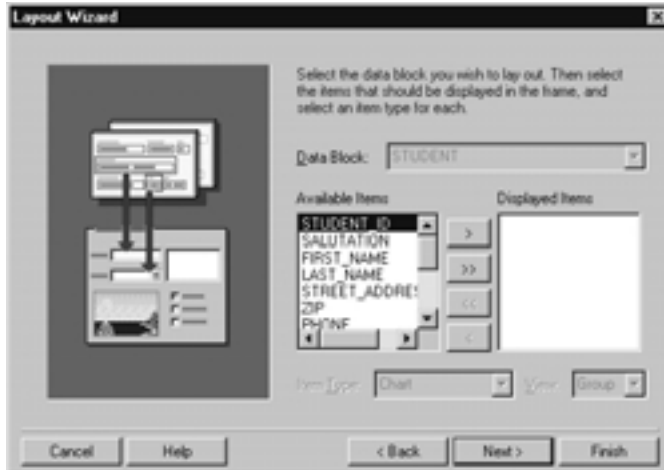


Figure 2.7 ■ The Layout Wizard's data block page.

In this case, you are starting a new form, so your only choice is to select `New Canvas` from the `Canvas` list item. However, if you were working with an existing form and had already created a canvas, you would see its name in the `Canvas` list. Also, because this is a new form, you will leave the `Type` list item set to `Content`.

Had you chosen `Tab` from the `Type` list, the `Tab Page` list would have become enabled, and you would have been able to select from a list of available tab pages.

Click the `Next` button to move on to the data block page.

DATA BLOCK PAGE

Here you have three tasks:

- 1) Select the items to be displayed.
- 2) Order the items.
- 3) Select the item types.

As you can see, this page is nearly identical in layout to the `Data Block Wizard's` table page. It behaves the same way as well, so you already know how to move items from the `Available Items` list to the `Displayed Items` list. Use whichever method you prefer to move all of the items to the `Displayed Items` list. It is possible and common to display only a few of the items that are available. It all depends on how you want your canvas to look.

Your next task is to order the items. The `Layout Wizard` will lay the items out on the screen in the order in which they appear in the `Displayed`



Figure 2.8 ■ The Layout Wizard's items page.

Items text list. To reorder items, simply drag and drop in the list to get them into the proper positions. Put the items in the following order:

- | | |
|-------------------|----------------------|
| 1) SALUTATION | 8) STUDENT_ID |
| 2) FIRST_NAME | 9) REGISTRATION_DATE |
| 3) LAST_NAME | 10) CREATED_BY |
| 4) EMPLOYER | 11) CREATED_DATE |
| 5) STREET_ADDRESS | 12) MODIFIED_BY |
| 6) ZIP | 13) MODIFIED_DATE |
| 7) PHONE | |

Your next task is to set the item type for each item. To do this, take the following steps:

- 1) Select an item in the Displayed Items list.
- 2) Change its type using the Item Type drop-down list.

Change the following items to type Display Item:

- CREATED_BY.
- CREATED_DATE.
- MODIFIED_BY.
- MODIFIED_DATE.

Once you have completed this task, click the Next button to move on to the items page.

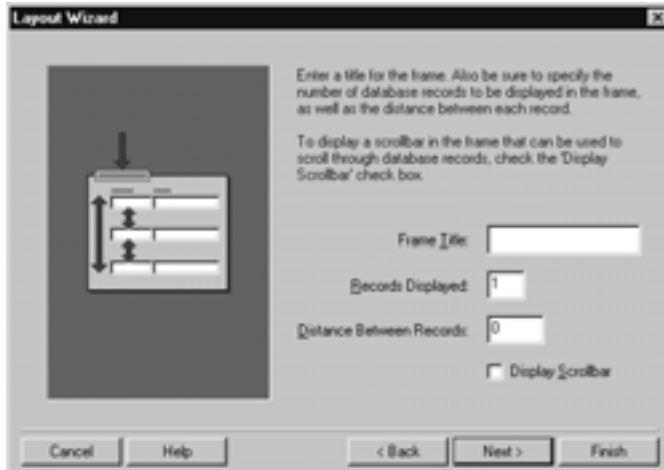


Figure 2.9 ■ The Layout Wizard's records page.

ITEMS PAGE

Here you have two tasks:

- 1) Adjust the `Prompt` values.
- 2) Adjust the `Height` and `Width` values.

To complete this task, simply position the cursor on the value that you'd like to change and edit it. Make the following adjustments:

- 1) Change the `Width` of `CREATED_BY` to 117.
- 2) Change the `Width` of `MODIFIED_BY` to 117.

After you have made these changes, click the `Next` button to continue to the style page.

STYLE PAGE

No screenshot is necessary here as this is a rather simple page. Your task is to decide how you would like the items to be laid out. In form style, the items are laid out so that the screen will resemble a paper-based form. In tabular style, the items are laid out in a grid. Look at the images on the left-hand side of the wizard page to see examples of how each style will look.

Select the `Form` style. Click the `Next` button to move on to the records page.

RECORDS PAGE

Here you have four tasks:

- 1) Choose a `Frame Title`.

- 2) Indicate the number of Records Displayed.
- 3) Set the Distance Between Records.
- 4) Include a scrollbar.

Because this is a new form and because you have chosen `Form` as the layout style, you will only have to complete the first task on this page. You will leave the rest as their default values. Type the name `Student` into the `Frame Title` field; this will be displayed on the canvas. Because the items will be laid out in form style, it is common to display only one record. Therefore, there is no need to change the value in the `Records Displayed` field. For the same reason, there is no need to change the value in the `Distance Between Records` field or the `Display Scrollbar` check box. In this Lab's Exercises, you will have to create some forms with a tabular style, which lends itself to displaying multiple records at one time with a scrollbar.

Click the `Next` button to move on to the finish page. You could also simply click the `Finish` button here as the finish page is purely informational.

FINISH PAGE

No screenshot has been included, as there are no tasks on this page for you to complete. It is purely an informational page.

Click the `Finish` button and the Layout Wizard will close and your new canvas will be displayed in the Layout Editor. It should resemble Figure 2.10.

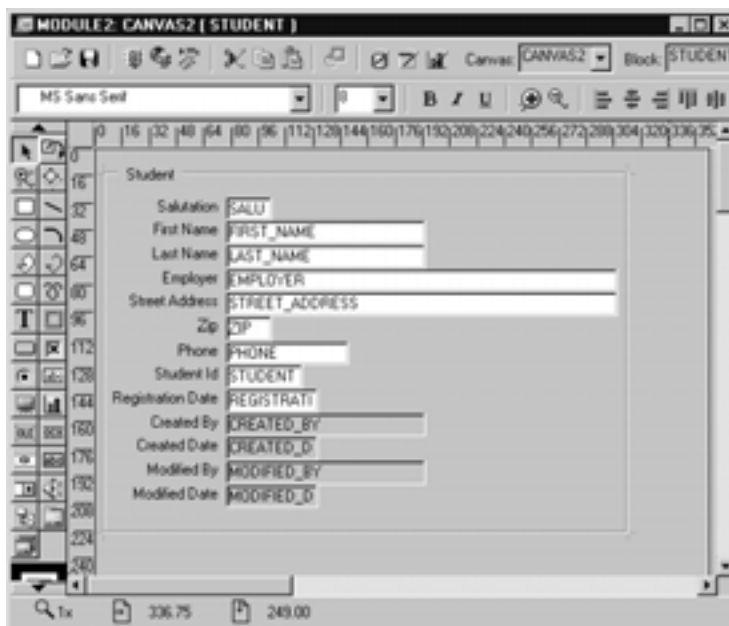


Figure 2.10 ■ The completed form in the Layout Editor.

[go to contents](#)

42 Lab 2.1: The Data Block and Layout Wizards

LAB 2.1

Now that your form is complete, you should save it. From the Main Menu, select `File | Save` and save it to your `\guest\forms\exercises` directory as `R_WIZARD.fmb`.

Now that the wizards are complete, you are free to edit the form. You can do this manually using the Layout Editor and Property Palette, or you can use the wizards again by reentering them.

In this Lab's Exercises, you will practice creating forms with the wizards and then you will edit one of your forms by reentering both the Data Block Wizard and Layout Wizard.

LAB 2.1 EXERCISES

2.1.1 USE THE DATA BLOCK AND LAYOUT WIZARDS

Explore the Form Builder and find at least four different ways to access the Data Block Wizard.

a) What are the four ways you discovered to open the Data Block Wizard?

Open the `R_WIZARD.fmb` form, which you created in the previous Lab. Expand the `Data Blocks` node, expand the `STUDENT` block, and then expand the `STUDENT` block's trigger node.

b) What is the name of this trigger?

Double-click the icon to the right of the trigger to open the PL/SQL Editor and view the trigger code.

c) What function will this trigger code perform?

d) Where did the Data Block Wizard get the information needed to write this trigger?

e) What did you do to make the Data Block Wizard create this trigger?

f) Have any triggers been created for other objects in this block? Fully expand the block and its items to find out.

View the properties for the STUDENT_ID item.

g) What are the values for the Data Type and Required properties?

h) How did the Data Block Wizard know to set them this way?

2.1.2 REENTERING THE WIZARDS

Select File | Close from the Main Menu to close the R_WIZARD.fmb form.

Select File | New | Form from the Main Menu to create a new form.

Use the wizards to quickly create a form based on the SECTION table. In the Data Block Wizard, include all of the SECTION table's columns in the block except CREATED_BY and MODIFIED_BY. Enforce data integrity should be **unchecked**.

44 Lab 2.1: The Data Block and Layout Wizards

LAB 2.1

In the Layout Wizard, display all of the items. Do not make any changes to Prompt, Width, or Height on the items page. Choose Form as the layout style. Display one record and do not include a scrollbar. Click the Finish button when you are done so that control returns to the Form Builder. Use the Name property in the Property Palette to rename the canvas SECTION and its frame SECTION. Select File | Save from the Main Menu and save the form to \guest\forms\exercises as R_SECTION.fmb.

a) Try to reenter the Data Block Wizard for the SECTION block using the Object Navigator. How did you do it?

b) How does the reentered Data Block Wizard look different from the wizard you are used to using?

c) Which tab page should you select to add CREATED_BY and MODIFIED_BY to the block? How did you add them?

Click the Finish button.

d) Have CREATED_BY and MODIFIED_BY been added to the block? How about the canvas?

Select the SECTION canvas in the Object Navigator and right-click to open the Layout Wizard.

e) Does it look like you have reentered the Layout Wizard? How can you tell that you haven't?

f) Which object should you select in the Object Navigator to reenter the Layout Wizard for the items in the SECTION block? Why? Hint: This is a physical object that helps you group and arrange items.

Position the reentered wizard in the upper right-hand corner of the screen so that you get a partial (or whole, depending on the size of your monitor) view of the Layout Editor.

Select the Data Block tab page in the wizard and move CREATED_BY and MODIFIED_BY to the Displayed Items list. Click the Apply button.

g) Between which items did they appear on the canvas? How can you use the wizard to position them so that they appear after CAPACITY and before CREATED_DATE?

Change the style of the form to Tabular and display five records. Click the Apply button to view the changes.

h) What steps did you take to accomplish this? List the tab page names as well as a brief description of what you did on each page.



After you apply the changes and exit the wizard, the layout will have changed and will not look very pleasing. Ignore this. The purpose of this Exercise is to experiment with reentering the wizard. Look and feel is not important here.

LAB 2.1 EXERCISE ANSWERS

2.1.1 ANSWERS

- a) What are the four ways you discovered to open the Data Block Wizard?

Answer: There are actually at least six ways to access the Data Block Wizard. They are as follows:

- 1) From the Main Menu, select `Tools | Data Block Wizard`.
- 2) In the Object Navigator, double-click the `Data Blocks` node. (Note: This will only work for the first block of each module.)
- 3) In the Object Navigator, select the `Data Blocks` node and click the `Create` button. The `Create` button has a green plus sign as its icon.
- 4) In the Object Navigator, select the `Data Blocks` node and right-click. Select `Data Block Wizard`.
- 5) In the Layout Editor, click the `Data Block Wizard` button. The `Data Block Wizard` button is situated in the middle of the Layout Editor's horizontal toolbar. It has a gray cylinder and a magic wand as its icon.
- 6) On the Form Builder Welcome dialog, in the section labeled `Designing`, select `Use the Data Block Wizard`, and click the `OK` button.

It is certainly not necessary to memorize these six ways now. However, as you become more familiar with the Form Builder, you will become anxious to speed your programming by learning different ways to accomplish the same tasks. As a beginner, it is wise to use the toolbars because they are visual and intuitive. Right-clicking is also very easy because you will be presented with a descriptive, context-sensitive list of choices.

- b) What is the name of this trigger?

Answer: The name of this trigger is `KEY-DELREC`.

Double-click the icon to the right of the trigger to open the `PL/SQL` Editor and view its code.

- c) What function will this trigger code perform?

Answer: It will prevent the user from deleting `STUDENT` records if `ENROLLMENT` records exist.

- d) Where did the Data Block Wizard get the information needed to write this trigger?

Answer: It queried the database's data dictionary to see if there were any integrity constraints defined for the table and its columns.

- e) What did you do to make the Data Block Wizard create this trigger?

Answer: You should have checked Enforce data integrity on the Data Block Wizard's table page.

- f) Have any triggers been created for other objects in this block? Fully expand the block and its items to find out.

Answer: Yes, WHEN-VALIDATE-ITEM triggers have been created for many of the items in the block.

As you have learned through these Exercises, checking Enforce data integrity results in the Form Builder writing a series of triggers in the form to enforce the integrity constraints that are stored at the database level. Had there been check constraints against the `STUDENT` table, triggers to enforce them would have been written as well. By checking Enforce data integrity, you are helping to insure that the data going to the database is valid.

■ FOR EXAMPLE:

Re-open the `R_WIZARD.fmb` form if you have already closed it.

Double-click the icon next to the `WHEN-VALIDATE-ITEM` trigger for the `STUDENT.ZIP` item. Note that the code has two sections. If you study the code for the first section, you can see that it is based on the `STU_ZIP_FK` foreign-key constraint. The trigger code is analogous to the constraint in that it will compare the `ZIP` value that a user has entered in the form with the `ZIP` values in the `ZIPCODE` table. If it does not find the value in the `ZIPCODE` table, the trigger will return an error. This corresponds to what you know about foreign-key constraints; a `ZIP` value cannot be inserted or updated into the `STUDENT` table unless that value already exists in the `ZIPCODE` table.

Well, you say, the database would have done this anyway: *that's* what the constraint is for. True, but the database would not have checked until the user had tried to commit the record. The user would have tabbed out of the `ZIP` item and would have had to navigate back to fix it. But, it is unlikely that the user would know which item to fix because the database would not have returned an intelligible error message. By putting the validation logic at the item level within the application itself, the user is alerted to his mistake immediately and is given a message he can understand.

- g)** What are the values for the Data Type and Required properties?

Answer: The Data Type is Number and Required is set to Yes.

- h)** How did the Data Block Wizard know to set them this way?

Answer: It inherited information from the data dictionary tables in the database. The Data Type of STUDENT_ID in the STUDENT table is Number and it is a not-null column.

The Data Block and Layout Wizards set many properties for items based on what they read in the data dictionary tables. The values for Prompt, Height, and Width are other examples of properties set by what is defined for the columns in the data dictionary tables. Keep in mind, however, that it is often necessary to adjust these properties after the wizard has set them.

■ FOR EXAMPLE:

By default, the wizard uses the column names in the database to create the prompts for the items. In the sample *STUDENT* schema, most of the column names are in near-plain English, so in most cases, there really isn't any need to change the default prompts that the wizard assigns. In many database systems, column and table names will not be in plain English, or they may be prefixed with some kind of system codes like *ST_STUD_NAM*. In cases like these, it is wise to use the items page of the Layout Wizard to change the prompts. You can certainly change the prompts later, but if you do it on the items page, the Layout Wizard will create the default layout with these prompts in mind. That is, it will adjust the layout automatically depending on the lengths of the prompts. If you choose to set the prompts manually in the Layout Editor, which is certainly possible, you may have to manually adjust more than just the prompt name to keep the layout neat and organized. This could cost you some time and considerable effort.

The length of an item is expressed in points. Forms makes each item approximately five points wide for each character. So, if a column in the database is set to be *VARCHAR(5)*, Forms will make its corresponding item approximately 25 points wide. These values are set for each module in the Coordinate System property. To view the Coordinate System settings, take the following steps:

- 1) Open *R_WIZARD.fmb* in the Form Builder.
- 2) In the Object Navigator, select the *R_WIZARD* module and view its properties.

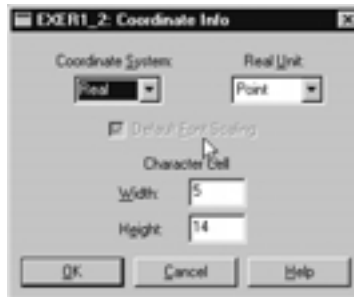


Figure 2.11 ■ The Coordinate Info dialog displaying the coordinate system information for the form.

- 3) Scroll down in the Property Palette window until you see the property named Coordinate System.
- 4) Select the Coordinate System property and click the More button.
- 5) You will see a window like the one in Figure 2.11.

You can reset the coordinate values here if you wish. The Coordinate System pop-up list lets you choose between Real and Character coordinates. Real gives you a finer grain of control over the positioning of items, while Character positions items by character cell. What you select here depends on how you plan to deploy the application. The Forms help system provides a small chart that explains which Coordinate System setting to choose for some of the more common deployment strategies.

If you choose the Real coordinate system you can choose which real unit you'd like to use and the size of each character cell. The Character Cell values determine the size of each character cell, expressed in the Real Unit you have chosen.

All of this is quite a mouthful and not worth worrying about now. It is certainly possible that at some point, depending on the deployment strategy, you will have to reset these values. For now you should understand the concepts behind the different coordinate systems, but stick with the default values.

2.1.2 ANSWERS

- a) Try to reenter the Data Block Wizard for the SECTION block using the Object Navigator. How did you do it?

Answer: You should have selected the SECTION block, then right-clicked and selected the Data Block Wizard.

[go to contents](#)

- b) How does the reentered Data Block Wizard look different from the wizard you are used to using?

Answer: It has tabbed pages instead of standard wizard pages.

The point of these two questions is to illustrate that it is necessary to have the proper block or one of its items selected in the Object Navigator to reenter the Data Block Wizard. If you had selected the `BLOCKS` node or some other object, the Form Builder would have assumed that you were trying to start a new block, not edit an existing one.

- c) Which tab page should you select to add `CREATED_BY` and `MODIFIED_BY` to the block? How did you add them?

Answer: The Table tab. They were added by moving them from the Available Columns list to the Database Items list.

The reentered wizard's interface behaves in the same way as the regular Data Block Wizard.

Click the `Finish` button.

- d) Have `CREATED_BY` and `MODIFIED_BY` been added to the block? How about the canvas?

Answer: Yes, they have been added to the block; no, they have not been added to the canvas.

They will not be added to the canvas or its frame until you reenter the Layout Wizard.

Interestingly, the Data Block Wizard does not seem to allow you to *remove* items from a block. You can go through the motions of removing an item in the Data Block Wizard, but when you click the `Apply` button, the item is not actually removed from the block. Luckily, it is easy to remove items simply by selecting them in the Object Navigator or Layout Editor and then deleting them manually by clicking the `DELETE` key.

Select the `SECTION` canvas in the Object Navigator and right-click to open the Layout Wizard.

- e) Does it look like you have reentered the Layout Wizard? How can you tell that you haven't?

Answer: No it doesn't. The pages aren't tabbed. It looks like the regular Layout Wizard.

- f) Which object should you select in the Object Navigator to reenter the Layout Wizard for the items in the SECTION block? Why? Hint: This is a physical object that helps you group and arrange items.

Answer: You should select the SECTION frame.

When the Layout Wizard creates and positions items on canvases, it also lays them out within frames. As described before, these frames are graphical objects that belong to canvases. They make it easier to control the layout of multiple groups of items.

■ FOR EXAMPLE:

In Chapter 4, “Master Detail Forms,” you will learn how to create forms, which will allow you, among other things, to have two groups of items on a single canvas. One of those groups will be laid out in form style, while the other will be laid out in tabular style. Each group of items will be in a frame. Having two frames will let you reenter the Layout Wizard for each group (frame) individually.

Select the Data Block tab page in the Wizard and move `CREATED_BY` and `MODIFIED_BY` to the Displayed Items list. Click the Apply button.

- g) Between which items did they appear on the canvas? How can you use the wizard to position them so that they appear after `CAPACITY` and before `CREATED_DATE`?

Answer: It depends on which item was selected in the Displayed Items list when you moved `CREATED_BY` and `MODIFIED_BY` over. You can drag the items up and down in the Displayed Items list to position them.

Change the style of the form to Tabular and display five records. Click the Apply button to view the changes.

- h) What steps did you take to accomplish this? List the tab page names as well as a brief description of what you did on each page.

Answer: On the style page, you must select the radio button labeled Tabular. On the rows page, you should set Records Displayed to 5 and check the check box labeled Display Scrollbar.

Reentering the wizard can be fast and efficient for adding items to the Displayed Items list or changing the layout style from Form to Tabular, or vice versa. However, it can sometimes be more trouble than it is worth. In the example, it stretched the layout far beyond the width of the canvas. As your proficiency with the Form Builder improves, you may find it easier to make adjustments manually in the Layout Editor rather than reentering the Layout Wizard.

LAB 2.1 SELF-REVIEW QUESTIONS

In order to test your progress, you should be able to answer the following questions:

- 1) Which of the following is true of all of the settings in the Layout Wizard?
 - a) They are final
 - b) They are subclassed to the block
 - c) They are based on frames and their widths
 - d) They contribute to the first cut of a canvas

- 2) Which of the following is true of the Data Block Wizard?
 - a) It associates blocks with tables
 - b) It creates display items for non-null columns
 - c) It has canvas properties
 - d) Both b & c

- 3) Where do the prompts set in the Layout Wizard become editable?
 - a) In the database
 - b) By reentering the wizard
 - c) In the Object Navigator
 - d) b & c

- 4) The wizards never do which of the following?
 - a) Write triggers
 - b) Access the database
 - c) Change module names
 - d) Create frames

Quiz answers appear in Appendix A, Section 2.1.

LAB 2.2

ORACLE FORMS FILES

LAB
2.2

LAB OBJECTIVES

After this Lab, you will be able to:

- Differentiate Between Source and Executable Files
- Compile Binary Files into Executable Files
- Run Executable Files

In the previous three Labs, you learned how Forms works, what its mandatory objects are, and how to create those objects quickly and easily using wizards. With this knowledge, you have been able to create some simple applications. Now you will learn how Forms stores the applications you create, how it readies those applications for deployment, and how it runs those readied files.

BINARY AND EXECUTABLE FILES

When you work in the Form Builder, you are creating binary files, and when you run an application, you are executing executable files. Forms stores its binary files with an `.fmb` extension and its executable files with an `.fmx` extension. You were already exposed to `.fmb` files when you were asked to open them in the Form Builder.

The `.fmx` files are the compiled version of `.fmb` files. To run a form for testing or deployment, it must first be compiled into an `.fmx` file. The `.fmx` files are the files that ultimately make it to your users.

Oracle Developer, and therefore Oracle Forms, is a multi-platform product. It can run on Windows systems as well as UNIX systems. It is possible to develop applications on one platform and then run them on another. This is because the `.fmb` files are platform-independent. They can be created on one platform and then edited or compiled on another.

The `.fmx` files, on the other hand, are platform-dependent. An `.fmx` file compiled on Windows NT cannot run on UNIX, and vice versa.

COMPILING

LAB 2.2

It is possible to compile `.fmb` files into executable `.fmx` files directly in the Form Builder. This creates an `.fmx` file and then allows you to continue working in the Form Builder. You may choose to do this if you'd simply like to check your form for errors. If there are any errors, the compilation will fail and the Form Builder will alert you to the existence and location of these errors.

However, you do not have to be in the Form Builder to compile a form. Installed with the Form Builder and the rest of the application components is a utility called the Form Compiler. If you are using a Windows platform, you may have seen it when you navigated through the `Start` menu to start the Form Builder.

You will get an opportunity to practice compiling using the Form Builder and Form Compiler in this Lab's Exercises.

RUNNING

It is common to want to test your forms as you are building them. You finish the wizard, make a few changes to the layout, add a couple of triggers, and then you want to test the form and see how it looks and operates. You can do this by running the form directly from the Form Builder.

You do not have to be in the Form Builder to run a form. Along with the Form Builder and Form Compiler, another utility called the Forms Runtime is available.

In the Exercises, you will get the opportunity to run forms from the Form Builder, as well as from the Forms Runtime.

LAB 2.2 EXERCISES

2.2.1 DIFFERENTIATE BETWEEN SOURCE AND EXECUTABLE FILES

a) What are the differences between `.fmb` files and `.fmx` files?

b) Which file do you work with in the Form Builder?

c) Do both files get distributed to application users? If not, which one does?

2.2.2 COMPILE BINARY FILES INTO EXECUTABLE FILES

Open `EX02_02.fmb` in the Form Builder. Select `File | Administration | Compile File` from the Main Menu.

a) How can you tell that a form compiled successfully?

b) What can you assume was created and written to the filesystem?

Expand the Triggers node for the module and double-click the icon for the `WHEN-NEW-FORM-INSTANCE` trigger. Delete the semi-colon after the word `Null`. Compile the file again using the steps for Question a.

c) Did the form compile successfully? How do you know?

Replace the semi-colon and compile the form a third time.

Select `Program` from the Main Menu and view the compilation options you see.

56 *Lab 2.2: Oracle Forms Files*

d) These options are for compilation, but they do not create `.fmx` files. What do you think they compile?

**LAB
2.2**

From the Windows `Start` menu, locate and run the Form Compiler. You will find it just below the Form Builder in the menu system. Once the Form Compiler has opened, use its `Browse` button to locate the file `EX02_02`.

e) Which should you compile, the `.fmx` file or `.fmb` file?

Enter your User ID, Password, and Database alias and click the `OK` button.

f) What happened?

g) Why can you develop your applications on a Windows NT workstation but then deploy them to a UNIX server?

h) What steps would you take to accomplish 2.2.2.g? Assume that the UNIX machine is running Solaris 2.6.

2.2.3 RUN EXECUTABLE FILES

Open EX02_02.fmb in the Form Builder.

a) Which button on the Object Navigator should you use to run the form?

b) What *has* to be happening by default when you run a module from the Form Builder?

From the Windows `Start` menu, locate and run the Forms Runtime. You will find it just below the Form Compiler in the menu system. Once the Forms Runtime is open, use the `Browse` button to locate the file EX02_02.

c) Which should you run, the .fmx file or the .fmb file?

A running form can be in one of three modes: Normal, Enter Query, or Fetch mode.

d) Which button on the default toolbar can you use to put the form into Enter Query mode?

e) In which mode should the form be to insert records?

LAB 2.2 EXERCISE ANSWERS

LAB
2.2

2.2.1 ANSWERS

- a) What are the differences between `.fmb` files and `.fmx` files?

Answer: .fmb files are binary files; .fmx files are executable files. .fmb files are platform-independent, while .fmx files are platform-dependent.

- b) Which file do you work with in the Form Builder?

Answer: You work with .fmb files in the Form Builder.

That is, you work with `.fmb`'s in the Form Builder when you are creating forms modules. Later you will learn how to create menus and libraries in the Form Builder, in which case you will work with `.mmb` and `.pll` files.

- c) Do both files get distributed to application users? If not, which one does?

Answer: No, both do not. Only .fmx files get distributed to application users.

2.2.2 ANSWERS

- a) How can you tell that a form compiled successfully?

Answer: The message Module Built Successfully will appear in the Form Builder's hint line.

You can also tell that it has compiled successfully when no error messages are reported.

- b) What can you assume was created and written to the filesystem?

Answer: You can assume that an .fmx file was created since the result of a successful compilation is an executable.

When the Form Builder creates `.fmx` files, it names them after the `.fmb` files. So, if you look in the filesystem, you will now find a file called `EX02_02.fmx`.

Compile the file again using the steps for Question a.

- c) Did the form compile successfully? How do you know?

Answer: No, a Compilation Errors window opened.

Note that the message in the window is very descriptive. It tells you where in the form it encountered the error and supplies an error number.

Answer: Under the Program selection from the Main Menu, there are options for Compile and Compile Selection.

- d) These options are for compilation, but they do not create `.fmx` files. What do you think they compile?

Answer: They compile the PL/SQL in the form, but do not create an `.fmx` file.

As your forms get more complicated, you will add more PL/SQL objects such as triggers, program units, and PL/SQL libraries. You can use the `Compile` and `Compile Selection` commands to compile these PL/SQL objects.

- e) Which should you compile, the `.fmx` file or `.fmb` file?

Answer: The `.fmb` file. The compiled version of the `.fmb` is saved as an `.fmx` file.

Enter your User ID, Password and Database alias and click the OK button.

- f) What happened?

Answer: Next to nothing. There was a pause, then the Form Compiler closed.

If the module compiles successfully, the Form Compiler will not give you any confirmation of success; it will simply close. You can confirm the compilation by looking for the `.fmx` file in the filesystem.

If there are any errors during compilation, they will be displayed in a Forms Compilation Error window and written to a text file. This text file will have the same name as the `.fmb` file and will have an `.err` extension. So, in this case, if there had been an error, you would have seen it listed in an error window and would have been able to find an `EX02_02.err` file in the filesystem.

- g) Why can you develop your applications on a Windows NT workstation but then deploy them to a UNIX server?

Answer: Because `.fmb` files are platform-independent and therefore portable from platform to platform.

The current release of Oracle Developer is supported on Windows 95, 98, and NT, as well as a host of UNIX and other platforms. For a complete list of supported platforms, visit www.oracle.com.

It is very common for application developers to develop their applications on a Windows workstation and then port them to a UNIX environment for deployment.

This is especially true when Oracle Forms will be deployed in a Web environment with a browser or applet viewer interface. In these cases, the applications are developed on Windows workstations and then transferred to powerful middle-tier servers. If these servers are not Windows NT servers, then the application will have to be recompiled for the appropriate operating system.

- h)** What steps would you take to accomplish 2.2.2.g? Assume that the UNIX machine is running Solaris 2.6.

Answer: To do this, you would take the following steps:

- 1) Develop a module using a Windows NT workstation.
- 2) Save the `.fmb` file.
- 3) FTP or somehow copy the `.fmb` file from the NT workstation to the Solaris machine.
- 4) Compile the `.fmb` on the UNIX machine using the Solaris version of the Form Compiler.
- 5) You will now have an `.fmx` that is runnable on Solaris.

Cross-platform development is a large topic, though it is not covered within the scope of this book. However, it is worth noting that it is one of the features that has made Oracle Forms such a popular product.

2.2.3 ANSWERS

Open `EX02_02.fmb` in the Form Builder.

- a)** Which button on the Object Navigator should you use to run the form?

Answer: From the Object Navigator's vertical toolbar or the Layout Editor's horizontal toolbar, click the Run Form Client/Server button. This button has a traffic light as its icon.

- b)** What has to be happening by default when you run a module from the Form Builder?

Answer: By default, the Form Builder has to be compiling the `.fmb` file and creating an `.fmx` file.

Remember, in the Form Builder you are working with `.fmb` files, which are binary files and therefore not executable. Only executables, `.fmx` files, can be run.

When you click Run, Forms automatically compiles the form before running it. If there are any errors during compilation, they will be displayed to you in a `Compilation Errors` window. If the form can run despite these errors, it will. If it can't, you will have to fix the errors in the Form Builder before continuing.

- c) Which should you run, the `.fmx` file or `.fmb` file?

Answer: You should run the `.fmx` file, which is the executable.

A running form can be in one of three modes: Normal, Enter Query, or Fetch mode.

- d) Which button on the default toolbar can you use to put the form into Enter Query mode?

Answer: The Enter Query button, which has a question mark in front of a cylinder as its icon.

When a form first opens, the default behavior is for it to be in Normal mode, which means it is capable of accepting new records or updating existing ones. By putting the form in Enter Query mode, it is set to accept a query by example.

■ FOR EXAMPLE:

Run the `EX02_02` form from the Form Builder. Click the `Enter Query` button. Type `101` into the `INSTRUCTOR ID` item. Click the `Execute Query` button. Note that only the record for Instructor 101 was returned to the form. You can add more than one parameter in Enter Query mode and you can also use wildcards. So, what would you do to get all of the instructors whose Zip Code is 10025 and whose `FIRST NAME` begins with "T?" Put the form into Enter Query mode, enter `T%` in the `FIRST NAME` item and `10025` in the `ZIP` item, and click the `Execute Query` button.

There are certain restrictions that apply to writing code in Enter Query mode, which you will learn in later Chapters as you begin to write more complicated logic and utilize built-ins.

- e) In which mode should the form be to insert records?

Answer: The form should be in Normal mode.

62 Lab 2.2: Oracle Forms Files

You can always get out of Enter Query mode and back to Normal mode by clicking the `Cancel Query` button on the toolbar.



Shortcut keys in the Form Builder can be very helpful for running and compiling forms. Use `CTRL-R` to run the form. Use `CTRL-T` to compile the form without running it.

LAB 2.2

LAB 2.2 SELF-REVIEW QUESTIONS

In order to test your progress, you should be able to answer the following questions.

- 1) Which of the following can be used to compile binary files?
 - a) ___ The Forms Runtime
 - b) ___ The Form Builder
 - c) ___ The Form Compiler
 - d) ___ b & c

- 2) Which of the following is not true about `.fmx` files?
 - a) ___ Distributed to users
 - b) ___ Platform-dependent
 - c) ___ Binary files
 - d) ___ Runnable from the Layout Editor

- 3) To run an `.fmx` client/server, the user would need which of the following?
 - a) ___ The Forms Runtime
 - b) ___ Access to a database
 - c) ___ Module files for the button icons
 - d) ___ The `.fmx` to be compiled for their operating system
 - e) ___ a, b, & d

- 4) Which of the following can be used to compile `.fmb`'s?
 - a) ___ The Form Compiler
 - b) ___ The Run button on the Object Navigator
 - c) ___ Program/Compile/All from the Form Builder's Main Menu
 - d) ___ `CTRL+T` in the Form Builder

Quiz answers appear in Appendix A, Section 2.2.

CHAPTER 2

TEST YOUR THINKING

Use the wizards to create several forms based on the following criteria. It is important to complete these now as you will be required to enhance some of these forms in later Chapters. Save all of the forms to your `\guest\forms\exercises` directory.

- 1) Create a form based on the `STUDENT` table. Include all of the columns as items in the block. Do not enforce data integrity. Display all of the items in the frame on the canvas except for the audit columns. Select `Form` for the layout style and display only one record. Give the frame a suitable name. Save the form as `R_STUDENT.fmb`.
- 2) Create a form based on the `INSTRUCTOR` table. Include all of the columns as items in the block. Do not enforce data integrity. Display all of the items in the frame on the canvas except for the audit columns. Select `Form` for the layout style and display only one record. Give the frame a suitable name. Save the form as `R_INSTRUCTOR.fmb`.
- 3) Create a form based on the `ENROLLMENT` table. Include all of the columns as items in the block. Do not enforce data integrity. Display all of the items in the frame on the canvas except for the audit columns. Select `Tabular` for the layout style and display five records. Include a scrollbar and give the frame a suitable name. Save the form as `R_ENROLLMENT.fmb`.
- 4) Create a form based on the `GRADE` table. Include all of the columns as items in the block. Do not enforce data integrity. Display all of the items in the frame on the canvas except for the audit columns. Select `Tabular` for the layout style and display five records. Include a scrollbar and give the frame a suitable name.

Reenter the wizard and adjust some of the item widths so that all of them fit neatly on the canvas. Shorten some of the prompt values to make this easier. Try to make the frame smaller using the Layout Editor.

Save the form as `R_GRADE.fmb`.