Oracle® Application Express

Advanced Tutorials

Release 3.1.2

E10497-02

August 2008

ORACLE®

Oracle Application Express Advanced Tutorials, Release 3.1.2

E10497-02

Primary Authors: Drue Baker, Anne Romano, Terri Winters

Contributors: Carl Backstrom, Amitabh Chhibber, Sharon Kennedy, Jason Straub, and Rekha Vallam

Contents

4 How to Create a Drill Down Report

5 How to Control Form Layout

6 How to Work with Check Boxes

7 How to Implement a Web Service

8 How to Create a Stacked Bar Chart

9 How to Upload and Download Files in an Application

10 How to Incorporate JavaScript into an Application

11 How to Build an Access Control Page

12 How to Review a Packaged Application

13 How to Create a Master Detail PDF Report

14 How to Build and Deploy an Issue Tracking Application

A DDLs and Scripts

Preface

Oracle Application Express Advanced Tutorials contains tutorials with step-by-step instructions that explain how to create a variety of application components and entire applications using the Oracle Application Express development environment.

Topics in this section include:

Audience

Oracle Application Express Advanced Tutorials is intended for application developers who wish to learn how to build database-centric Web applications using Oracle Application Express. To use this guide, you need to have a general understanding of relational database concepts, the operating system environment under which you are running the Oracle Application Express, and Application Builder.

What is Application Builder? A component of Oracle Application Express, Application Builder is a powerful tool that enables you to quickly assemble an HTML interface (or application) on top of database objects such as tables and procedures. Prior to completing these tutorials, please review *Oracle Database 2 Day + Application Express Developer's Guide*.

See Also: "About these Tutorials" on page 1-1

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Related Documents

For more information, see these Oracle resources:

- *Oracle Application Express Installation Guide*
- *Oracle Application Express Release Notes*
- *Oracle Database 2 Day + Application Express Developer's Guide*
- *Oracle Application Express Application Builder User's Guide*
- *Oracle Application Express SQL Workshop and Utilities Guide*
- *Oracle Application Express API Reference*
- *Oracle Application Express Migration Guide*
- *Oracle Application Express Administration Guide*
- *Oracle Database Concepts*
- *Oracle Database Advanced Application Developer's Guide*
- *Oracle Database Administrator's Guide*
- *Oracle Database SQL Language Reference*
- *SQL*Plus User's Guide and Reference*

For information about Oracle error messages, see *Oracle Database Error Messages*. Oracle error message documentation is available only in HTML. If you have access to the Oracle Database Documentation Library, you can browse the error messages by range. Once you find the specific range, use your browser's "find in page" feature to locate the specific message. When connected to the Internet, you can search for a specific error message using the error message search feature of the Oracle online documentation.

Many books in the documentation set use the sample schemas of the seed database, which is installed by default when you install Oracle. Refer to *Oracle Database Sample Schemas* for information on how these schemas were created and how you can use them yourself

Printed documentation is available for sale in the Oracle Store at

`http://oraclestore.oracle.com/`

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

http://www.oracle.com/technology/membership/

If you already have a user name and password for OTN, then you can go directly to the documentation section of the OTN Web site at

http://www.oracle.com/technology/documentation/

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

About these Tutorials

Oracle Application Express Advanced Tutorials contains a series of tutorials that explain how to use Oracle Application Express to create applications and application components. The goal of this book is to help you understand how to use Oracle Application Express through hands-on experience.

This section contains the following topics:

What this Book Is Not

Oracle Application Express Advanced Tutorials contains 12 tutorials with step-by-step instructions. The objective of these tutorials is to demonstrate how to build a particular type of application or application component using the Oracle Application Express development environment.

Where appropriate, this book describes concepts relevant to understanding or completing a task. However, this book is not intended to be a complete discussion of Oracle Application Express concepts. For this type of information, see Oracle Application Express online Help or *Oracle Application Express Application Builder User's Guide*.

If you are new to Oracle Application Express, please review the *Oracle Database 2 Day + Application Express Developer's Guide*. This guide introduces you to application development using Oracle Application Express. It leads your through the process of setting up your development environment and walks you through building an initial application.

Tutorial Topics

This document contains the following tutorials:

Title	Description
How to Create a Tabular Form	Illustrates how to create a tabular form within a new application and how to change one of the updatable columns from a text field to a select list.

Title	Description
How to Create a Parameterized Report	Illustrates how to create a report based on a SQL query that restricts the query to the value of a form item within the application.
How to Create a Drill Down Report	Describes how to create a report on a table. You then modify the report to contains drill down links to details in another report.
How to Control Form Layout	Explains how to create a data input form and then change the form layout by editing the region and item attributes.
How to Work with Check Boxes	Illustrates the different ways in which you can create and process check boxes within an application.
How to Implement a Web Service	Explains how to call a Web service from within an application.
How to Create a Stacked Bar Chart	Explains how to create a stacked bar chart within an application.
How to Upload and Download Files in an Application	Illustrates how to create a form and report with links for file upload and download.
How to Incorporate JavaScript into an Application	Describes some usage scenarios for JavaScript and includes details about how to implement them in your application.
How to Build an Access Control Page	Explains how to build an Access Control Administration to restrict access to an application.
How to Review a Packaged Application	Explores the *OEHR Sample Objects* packaged application. By reviewing the supporting objects behind this application, you can learn how the Supporting Object Utility works so that you can create your own packaged applications.
How to Create a Master Detail PDF Report	Explains how to create a master detail form, define a report query and RTF template, and then create a button to expose the new report.
How to Build and Deploy an Issue Tracking Application	Provides step-by-step instructions on how to create and deploy an application that tracks the assignment, status, and progress of issues related to a project.

About Loading Sample Objects

In Oracle Application Express, users log in to a workspace. Think of each workspace as a shared work area that separates your objects, data, and applications into a virtual private database.

Before you can start these exercises, you need to create the appropriate sample objects within your workspace. These sample objects are copies of the objects that are typically installed in two schemas:

- Human Resources (HR)

  The HR schema contains information about the employees and the facilities where they work. Each employee has an identification number, email address, job identification code, salary, and manager. Employees are assigned to a department and each department is associated with one location that has a full address. including the street name, postal code, city, state or province, and country code.

- Order Entry (OE)

  The OE schema tracks product inventories and sales of a company's products, including the product identification number, the product name, the associated

associates product category, product descriptions, the weight group (for shipping purposes), the warranty periods, the supplier, the status availability, and a minimum price.

To create the objects locally in your workspace, you need to import the *OEHR Sample Objects* application.

This section contains the following topics:

- Downloading OEHR Sample Objects
- Importing and Installing OEHR Sample Objects
- Viewing Database Objects

> **Tip:** In order to successfully import the objects associated with the *OEHR Sample Objects* application, your Oracle database must include Oracle Spatial. If your database instance does not include Oracle Spatial, you can install it using Database Configuration Assistant. To learn more, see the *Oracle Database Installation Guide* for your operating environment.

Downloading OEHR Sample Objects

To import the *OEHR Sample Objects* application, you first need to download it from the Oracle Technology Network (OTN):

1. In your Web browser go to:

   `http://www.oracle.com/technology/products/database/application_express/packaged_apps/oehr_sample_objects.zip`

2. Locate the *OEHR Sample Objects* application.
3. Download the `oehr_sample_objects.zip` file to your computer.
4. Unzip and extract the `oehr_sample_objects_installer.sql` file:
   - Microsoft Windows - Double-click the `oehr_sample_objects.zip` file
   - UNIX or Linux - Enter the following command:

     ```
     $ unzip oehr_sample_objects.zip
     ```

Importing and Installing OEHR Sample Objects

After you download the *OEHR Sample Objects* application, you need to import it into Oracle Application Express. During the import process, specify that you also want to install both the application and the supporting objects. Installing the application creates the objects and sample data needed to complete the exercises in *Oracle Application Express Advanced Tutorials*.

To import and install the *OEHR Sample Objects* application:

1. Log in to Oracle Application Express. See "Logging In To Oracle Application Express" in *Oracle Application Express Application Builder User's Guide*.
2. On the Workspace home page, click **Application Builder**.

   The Application Builder home page appears.
3. Click the **Import** button.
4. For Specify File, specify the following:

a. Import file - Click **Browse** and go to the `oehr_sample_objects_installer.sql` file.

b. File Type - Select **Application, Page, or Component Export**.

c. Verify that File Character Set is correct.

d. Click **Next**.

Now that you have imported the file, you want to install it.

5. To install an imported file, click **Next**.

   The Install Application Wizard appears.

6. In the Install Application Wizard, specify the following:

   a. Parsing Schema - Select a schema.

   b. Build Status - Select **Run and Build Application**.

   c. Install As Application - Select **Auto Assign New Application ID**.

   d. Click **Install**.

7. For Supporting Objects, select **Yes** and click **Next**.
8. Confirm your selections by clicking **Install**.
9. Click the **Home** breadcrumb link at the top of the page.

   The Application Builder home page appears.

See Also: "Viewing Database Objects" on page 1-5

Checking Available Space in Your Workspace

If you experience problems installing the *OEHR Sample Objects* application, verify the available space in your workspace. You many need to request additional storage space.

If you are a workspace administrator, you can:

1. Determine if you need additional storage space. See "Viewing the Workspace Overview Report" in *Oracle Application Express Application Builder User's Guide*.
2. Request additional storage space. See "Requesting Additional Storage" in *Oracle Application Express Application Builder User's Guide*.

Deleting the OEHR Sample Objects Application

Deleting the *OEHR Sample Objects* application and selecting to deinstall the supporting objects completely removes all associated objects and sample data.

To delete the *OEHR Sample Objects* application:

1. Log in to Oracle Application Express.
2. On the Workspace home page, click **Application Builder**.

   The Application Builder home page appears.

3. Select the *OEHR Sample Objects* application.

   The Application home page appears.

4. On the Tasks list, click **Delete this Application**.

   The Deinstall page appears.

5. To remove all associated objects and sample data, select **Remove Application Definition** and **Deinstall Supporting Objects**.
6. Click **Deinstall**.

Viewing Database Objects

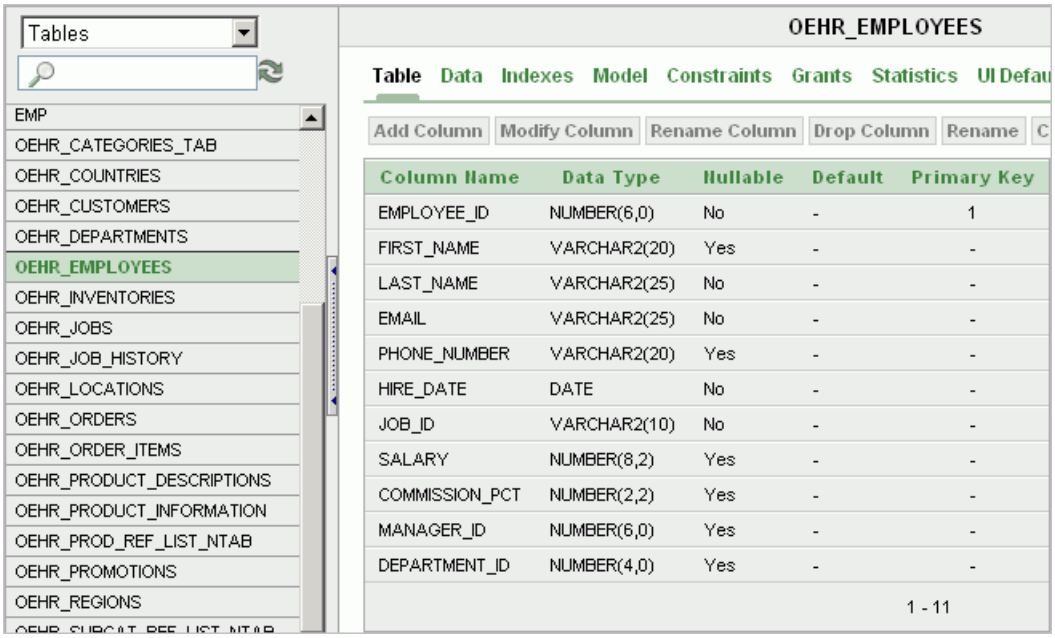
Now, take a look at the objects you just created by going to Object Browser. Object Browser enables you to browse, create, and edit objects in your database.

To view the objects:

1. On the Workspace home page, click **SQL Workshop**.
2. Click **Object Browser**.

   As shown in Figure 1–1 on page 1-5, Object Browser appears.

Figure 1–1 Object Browser



Object Browser is divided into two sections:

- Object Selection pane displays on the left side of the Object Browser page and lists database objects of a selected type within the current schema.

  The list of objects that appears depends upon the available objects in the current schema. Note that any object having a red bar adjacent to it is invalid.

- Detail pane displays to the right of the page and displays detailed information about the selected object.

3. From the Object Selection list, select **Tables**.
4. In the Object Selection pane, click **OEHR_EMPLOYEES** from the list.

   The Detail pane shows details about the table.
5. Click the **Data** tab in the row at the top of the Details pane.

   The data in the OEHR_EMPLOYEES table appears. Note that other tabs show additional details about the object you select.

6. To search for an object name, enter a case insensitive term in the Search field.
7. To view all objects, leave the Search field blank.

> **See also:** "Managing Database Objects with Object Browser" in *Oracle Application Express Application Builder User's Guide*

About Application Authentication

As you create new pages, you can view them by running the page individually or by running an entire application. When you run a page or application, the Application Express engine dynamically renders it into viewable HTML based on data stored in the database.

By default, all the applications you create in these tutorials use Application Express Authentication. Application Express Authentication is a built-in authentication scheme that uses the same internal user accounts you use to log in to a workspace.

The first time you run a page in an application, you are prompted to enter a user name and password. To continue, simply enter your workspace user name and password and then click Login.

When you create your own applications, you can choose from a number of preconfigured authentication schemes or build your own.

> **See Also:** "Establishing User Identity Through Authentication" in *Oracle Application Express Application Builder User's Guide*.

2

How to Create a Tabular Form

A tabular form enables users to update multiple rows in a table at once from a single page. You can use the Tabular Form Wizard to create a tabular form that contains a built-in multiple row update process. This built-in process performs optimistic locking behind the scenes to maintain the data integrity.

This tutorial explains how to create a tabular form within a new application and then how to change one of the updatable columns from a text field to a select list. Before you begin, you need to import and install the *OEHR Sample Objects* application in order to access the necessary sample database objects. See "About Loading Sample Objects" on page 1-2.

This section contains the following topics:

Creating an Application

First, you need to create an application using the Create Application Wizard.

To create an application using the Create Application Wizard:

1. On the Workspace home page, click the **Application Builder** icon.

   The Application Builder home page appears.
2. Click **Create**.
3. Select **Create Application** and click **Next**.
4. For Name:
   a. Name - Enter `Tabular Form`.
   b. Application - Accept the default.
   c. Create Application - Select **From scratch**.
   d. Schema - Select the schema where you installed the OEHR sample objects.
   e. Click **Next**.

      Next, you need to add a page. You have the option of adding a blank page, a report, a form, a tabular form, or a report and form. For this exercise, you create an application containing a blank page. Then, you create a tabular form.
5. Add a blank page:

a. Under Select Page Type, select **Blank** and click **Add Page**.

   The new page appears in the list at the top of the page.

b. Click **Next**.

6. For Tabs, accept the default, **One Level of Tabs,** and click **Next**.
7. For Copy Shared Components from Another Application, accept the default, **No**, and click **Next**.
8. For Attributes, accept the defaults for Authentication Scheme, Language, and User Language Preference Derived From and click **Next**.
9. For User Interface, select **Theme 2** and click **Next**.

   A theme is collection of templates that define the layout and style of an application. You can change a theme at any time.

10. Review your selections and click **Create**.

    The Application home page appears.

    See Also: "Managing Themes" in *Oracle Application Express Application Builder User's Guide*

Creating a Tabular Form Using a Wizard

The Tabular Form Wizard creates a form to perform update, insert, and delete operations on multiple rows in a database table. Additionally, the wizard creates a multiple row update process that checks for MD5 checksum values before doing the update to prevent lost updates. In the following exercise you create a tabular form on the `OEHR_EMPLOYEES` table.

To create a tabular form using the Tabular Form Wizard:

1. On the Application home page, click **Create Page**.
2. For the page type, select **Form** and click **Next**.
3. Select **Tabular Form** and click **Next**.
4. For Table/View Owner:
   a. Table/View Owner - Accept the default.
   b. Allowed Operations - Accept the default, **Update, Insert, and Delete**.
   c. Click **Next**.
5. For Table/View Name, select **OEHR_EMPLOYEES** and click **Next**.
6. For Displayed Columns:
   a. For Select Columns, press **Ctrl** and select the following columns:

      ```
      FIRST_NAME, LAST_NAME, HIRE_DATE, SALARY, DEPARTMENT_ID
      ```

      Note: This exercise limits the number of columns to optimize the display on-screen. For a real form, you would probably want to include additional columns.

   b. Click **Next**.

7. For Primary Key, accept the default, **EMPLOYEE_ID (Number)** and click **Next**.
8. For Source Type, accept the default, **Existing trigger**, and click **Next**.
9. For Updatable Columns, select all columns and click **Next**.
10. For Page and Region Attributes:
    a. Page - Accept the default.
    b. Page Name - Enter `Tabular Form`.
    c. Region Title - Accept the default, `Tabular Form`.
    d. Region Template and Report Template - Accept the defaults.
    e. Breadcrumb - Accept the default.
    f. Click **Next**.
11. For Tab, accept the default, **Do not use tabs,** and click **Next**.
12. For Button Labels, specify the following:
    a. Submit button - Enter `Apply Changes`.
    b. Cancel, Delete, and Add Row buttons - Accept the default label text.
    c. Click **Next**.
13. For Branching, accept the defaults and click **Next**.

    Branching tells the Web browser what page to display when the current page is submitted for processing. In this case, you want the user to remain on the current page.
14. Confirm your selections and click **Finish**.

Next, run the page to view your new form.

To run the page:

1. Click the **Run Page** icon as shown in Figure 2–1.

Figure 2–1 Run Page Icon

2. If prompted to enter a user name and password, enter your workspace user name and password and click **Login**. See "About Application Authentication" on page 1-6.

   The tabular form appears as shown in Figure 2–2 on page 2-4.

Figure 2–2 Tabular Form

As shown in Figure 2–2, note that the tabular form contains four buttons. Cancel, Delete, and Apply Changes display in the upper right corner and Add Row displays in the bottom right corner. Additionally, a check box appears to the left of each row to enable the user to select the current row. Users can also select all rows at once by selecting the check box to the left of the column headings. The same check box is also used in conjunction with the Delete button to identify the rows to be deleted.

Note that the overall form layout (that is, the color scheme, button placement, region header placement, and so on) are controlled by templates in the currently selected theme.

> **See Also:** "Managing Themes" in *Oracle Application Express Application Builder User's Guide*

Changing an Updatable Column to a Select List

When the Tabular Form Wizard creates a tabular form, updatable columns are displayed, by default, as text fields. In the next exercise, you change the default display of the Department Id column to a select list. To accomplish this, you create a named list of values (LOV) and then edit the column attributes.

Topics in this section include:

- Create a Named List of Values
- Edit the Column to Display as a Select List

> **See Also:** "Creating Lists of Values" in *Oracle Application Express Application Builder User's Guide*

Create a Named List of Values

To create a named LOV for the Department Id:

1. Click **Edit Page 2** on the Developer toolbar as shown in Figure 2–3.

Figure 2–3 Developer Toolbar

The Page Definition for page 2 appears.

2. Under List of Values, click the **Create** icon.

Figure 2–4 Create Icon

The Create List of Values Wizard appears.

3. For Source, select **From Scratch** and click **Next**.
4. For Name and Type:
   a. Name - Enter `DEPTID`.
   b. Type - Select **Dynamic**.
   c. Click **Next**.
5. For Query or Static Values, replace the existing text with this:

```
SELECT DISTINCT department_id a, department_id b FROM oehr_employees
```

6. Click **Create List of Values**.

   The Page Definition for page 2 appears. Note that the LOV does not yet appear on the Page Definitions.

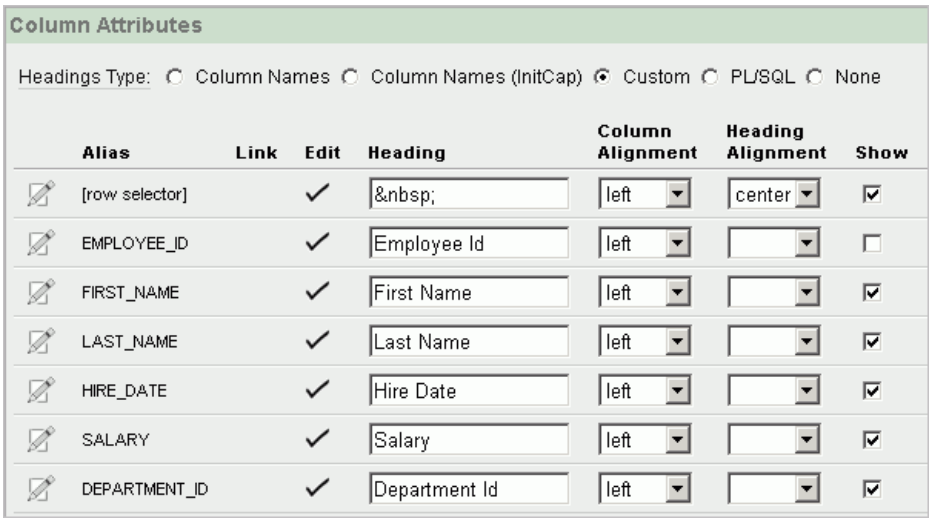
Edit the Column to Display as a Select List

To edit the column to display as a select list:

1. Under Regions, click the **Report** link.

   The Report Attributes page appears as shown in Figure 2–5.

Figure 2–5 Column Attributes on the Report Attributes Page



2. Under Column Attributes, click the **Edit** icon next to the DEPARTMENT_ID column as shown in Figure 2–5. The Edit icon resembles a small page with a pencil on top of it.

   The Column Attributes page appears.

   Next, change the default display of this column to a select list.

3. Scroll down to Tabular Form Element. From Display As, select **Select List (named LOV)**.

4. Scroll down to Lists of Values. From Named LOV, select **DEPTID**.

5. Scroll up to the top of the page and click **Apply Changes**.

6. Click the **Run Page** icon in the upper right corner of the page.

   As shown in Figure 2–6 on page 2-7, notice the Department Id column now displays as a select list.

Figure 2–6 Tabular Form with Department Id Column Changed to a Select List

Note: Do not modify the select list of a SQL statement of a tabular form after it has been generated. Doing so can result in a checksum error when altering the data in the form and applying updates.

Consider the following example:

```
SELECT first_name FROM oehr_employees;
```

Note that this should not be altered to:

```
SELECT lower(first_name) FROM oehr_employees
```

3

How to Create a Parameterized Report

In an Oracle Application Express application, a report is the formatted result of a SQL query. You can generate reports in three ways:

- Running a built-in wizard
- Defining a report region based on a SQL query
- Creating a report region based on a PL/SQL function returning a SQL query

This tutorial illustrates how to create a report in which the results depend on the form input, otherwise known as a parameterized report. In this exercise, you create a report region based on a SQL query that references the value of form items within the application.

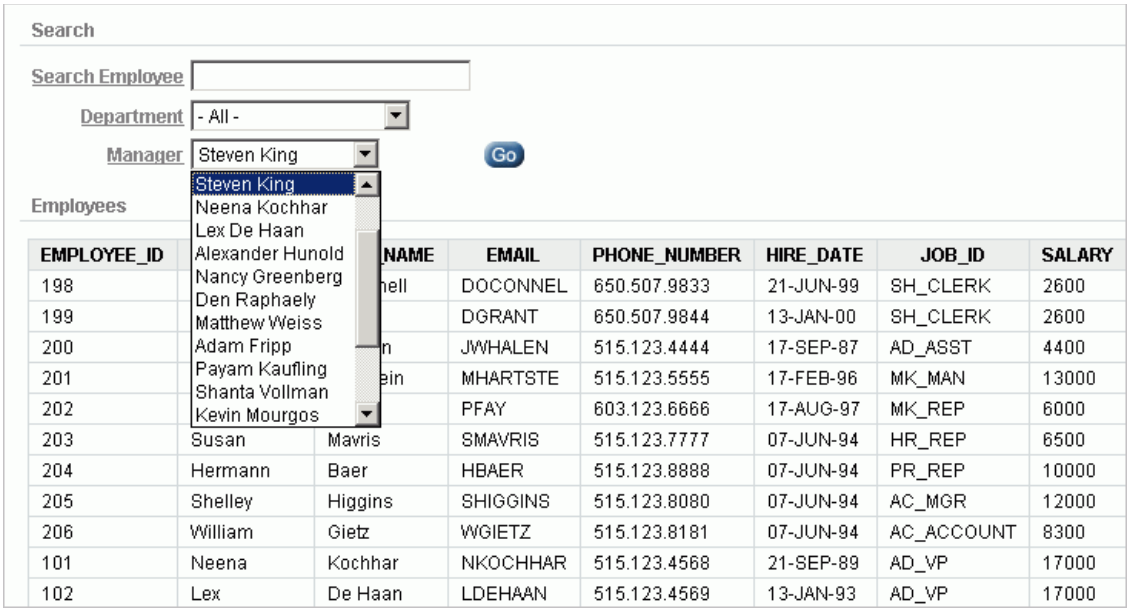
Before you begin, you need to import and install the *OEHR Sample Objects* application in order to access the necessary sample database objects. See "About Loading Sample Objects" on page 1-2.

This section contains the following topics:

Sample Report Utilizing a Form Input

Figure 3–1 on page 3-2 is an example of a form in which the report results are based on user input. In this example, the user populates the form by entering an employee ID in the Search Employee field, or by making a selection from two select lists. The easiest way to create this type of report in Application Builder is to define a report region based on a SQL query.

Figure 3–1 Sample Report



EMPLOYEE_ID		NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
198		hell	DOCONNEL	650.507.9833	21-JUN-99	SH_CLERK	2600
199			DGRANT	650.507.9844	13-JAN-00	SH_CLERK	2600
200		n	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400
201		ein	MHARTSTE	515.123.5555	17-FEB-96	MK_MAN	13000
202			PFAY	603.123.6666	17-AUG-97	MK_REP	6000
203	Susan	Mavris	SMAVRIS	515.123.7777	07-JUN-94	HR_REP	6500
204	Hermann	Baer	HBAER	515.123.8888	07-JUN-94	PR_REP	10000
205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94	AC_MGR	12000
206	William	Gietz	WGIETZ	515.123.8181	07-JUN-94	AC_ACCOUNT	8300
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000

Creating an Application

First, you need to create an application using the Create Application Wizard.

To create an application using the Create Application Wizard:

1. On the Workspace home page, click the **Application Builder** icon.

   The Application Builder home page appears.
2. Click **Create**.
3. Select **Create Application** and click **Next**.
4. For Name, specify the following:
   a. Name - Enter `Parameterized Report`.
   b. Application - Accept the default.
   c. Create Application - Select **From scratch**.
   d. Schema - Select the schema where you installed the OEHR sample objects.
   e. Click **Next**.

      Next, add a blank page.
5. Under Add Page, specify the following:
   a. Select Page Type - Select **Blank**.
   b. Page Name - Enter `Employees`.
   c. Click **Add Page**.

      The new page appears in the list at the top of the page.
   d. Click **Next**.
6. For Tabs, accept the default, **One Level of Tabs,** and click **Next**.

7. For Copy Shared Components from Another Application, accept the default, **No**, and click **Next**.
8. For Attributes, accept the defaults for Authentication Scheme, Language, and User Language Preference Derived From and click **Next**.
9. For User Interface, select **Theme 18** and click **Next**.

   A theme is collection of templates that define the layout and style of an application. You can change a theme at any time.
10. Review your selections and click **Create**.

    The Application home page appears.

    See Also: "Managing Themes" in *Oracle Application Express Application Builder User's Guide*

Creating Regions

Next, you need to create regions. A region is an area on a page that serves as a container for content. For this exercise you need to create two regions: a Search region to contain search criteria items and a Query region that displays the resulting report.

Topics in this section include:

- Create a Search Region
- Create a Query Region

Create a Search Region

To create a search region:

1. On the Application home page, click **1 - Employees**.

   The Page Definition for page 1 appears.
2. Under Regions, click the **Create** icon as shown in Figure 3–2.

Figure 3–2 Create Icon

3. For Region:
   a. Identify the type of region to add to this page - Accept the default, **HTML**, and click **Next**.
   b. Select the type of HTML region container you wish to create - Accept the default, **HTML**, and click **Next**.
4. For Display Attributes, specify the following:
   a. For Title - Enter `Search`.
   b. Accept the remaining default values.
   c. Click **Next**.
5. Click **Create Region**.

The Page Definition for page 1 appears. A confirmation message displays at the top of the page: Region created.

Create a Query Region

Next, you need to create a report region based on a SQL query.

> **See Also:** "Understanding Regions" in *Oracle Application Express Application Builder User's Guide*

To create a report region based on a SQL query:

1. Under Regions, click the **Create** icon.
2. For Region, select **Report** and click **Next**.
3. For Report Implementation, select **SQL Report** and click **Next**.
4. For Display Attributes:
   a. Title - Enter the following, making sure to include the trailing period:

      ```
      Employees &P1_TEXT.
      ```

      `&P1_TEXT` is a substitution string that will determine region title. You create this item in the next section.

   b. Accept the remaining default values and click **Next**.
5. Enter the following SQL query:

```
SELECT
   "OEHR_EMPLOYEES"."EMPLOYEE_ID" "EMPLOYEE_ID",
   "OEHR_EMPLOYEES"."FIRST_NAME" "FIRST_NAME",
   "OEHR_EMPLOYEES"."LAST_NAME" "LAST_NAME",
   "OEHR_EMPLOYEES"."EMAIL" "EMAIL",
   "OEHR_EMPLOYEES"."PHONE_NUMBER" "PHONE_NUMBER",
   "OEHR_EMPLOYEES"."HIRE_DATE" "HIRE_DATE",
   "OEHR_EMPLOYEES"."JOB_ID" "JOB_ID",
   "OEHR_EMPLOYEES"."SALARY" "SALARY",
   "OEHR_EMPLOYEES"."COMMISSION_PCT" "COMMISSION_PCT",
   "OEHR_EMPLOYEES"."MANAGER_ID" "MANAGER_ID",
   "OEHR_EMPLOYEES"."DEPARTMENT_ID" "DEPARTMENT_ID"
FROM
   "#OWNER#"."OEHR_EMPLOYEES" "OEHR_EMPLOYEES"
WHERE
     (lower(first_name) like '%' || lower(:P1_NAME) || '%' OR
      lower(last_name) like '%' || lower(:P1_NAME) || '%')
 AND department_id = decode(:P1_DEPT,'%null%',department_id,:P1_DEPT)
 AND manager_id = decode(:P1_MGR,'%null%',manager_id,:P1_MGR)
```

The `WHERE` clause forces both the search criteria and value from the database to be lower case. This makes the resulting search case insensitive for first and last names.

6. Click **Create Region**.

   The Page Definition for page 1 appears. A confirmation message displays at the top of the page.

Adding Form Items

An item is part of an HTML region. An item can be a text field, text area, password, select list, check box, and so on. The previous SQL query references the following items: `P1_NAME`, `P1_DEPT`, `P1_MGR`, and `P1_TEXT`. Next, you need to create these items.

Topics in this section include:

- Create a Search Employee Text Field
- Create a Hidden Text Field
- Create Department and Manager Select Lists

> **See Also:** "Understanding Page-Level Items" in *Oracle Application Express Application Builder User's Guide*.

Create a Search Employee Text Field

To create the text field `P1_NAME`:

1. Under Items, click the **Create** icon.
2. For Item Type, select **Text** and click **Next**.
3. For Text Control Display Type, select **Text Field** and click **Next**.
4. For Display Position and Name:
   a. For Item Name - Enter `P1_NAME`.
   b. For Sequence - Accept the default.
   c. For Region - Select **Search**.
   d. Click **Next**.
5. For Item Attributes:
   a. Label - Enter `Search Employee`.
   b. Accept the defaults.
   c. Click **Next**.
6. Accept the defaults and click **Next**.
7. Click **Create Item**.

Create a Hidden Text Field

Next, you will create a hidden text field named `P1_TEXT`. The value of `P1_TEXT` is used as the basis for the region title that displays.

To create the hidden text field P1_TEXT:

1. Under Items, click the **Create** icon.
2. For Item Type, select **Hidden** and click **Next**.
3. For Display Position and Name:
   a. For Item Name - Enter `P1_TEXT`.
   b. For Sequence - Accept the default.
   c. For Region - Select **Employees &P1_TEXT**.

d. Click **Next**.

4. Accept the remaining defaults and click **Next**.
5. Click **Create Item**.

Create Department and Manager Select Lists

Next, you need to create two items named `P1_DEPT` and `P1_MGR`. These items will display as select lists and be based on two named (or shared) lists of values.

A list of values (LOV) is a static or dynamic set of values used to display a page item. To create these items, you first define two dynamic LOVs and then create the items `P1_DEPT` and `P1_MGR`.

Topics in this section include:

- Create a Named LOV for Department
- Create a Named LOV for Manager
- Create an Item Named P1_DEPT
- Create an Item Named P1_MGR

> **See Also:** "Creating Lists of Values" in *Oracle Application Express Application Builder User's Guide*

Create a Named LOV for Department

To create a named LOV for department:

1. Under Lists of Values, click the **Create** icon.
2. For Source, accept the default, **From Scratch**, and click **Next**.
3. For Name and Type:
   a. Name - Enter `DEPARTMENT`.
   b. Type - Select **Dynamic**.
   c. Click **Next**.
4. For Query or Static Values, replace the existing text with:

```
SELECT department_name,department_id FROM oehr_departments
```

5. Click **Create List of Values**.

   The Page Definition appears.

Create a Named LOV for Manager

To create a named LOV for manager:

1. Under Lists of Values, click the **Create** icon.
2. For Source, accept the default, **From Scratch**, and click **Next**.
3. For Name and Type:
   a. Name - Enter `MANAGER`.
   b. Type - Select **Dynamic**.
   c. Click **Next**.
4. For Query or Static Values, replace the existing text with:

```
SELECT y.first_name || ' ' || y.last_name d, y.employee_id r
FROM oehr_employees y
WHERE y.employee_id IN ( SELECT x.manager_id FROM oehr_employees x)
```

5. Click **Create List of Values**.

   The Page Definition appears.

Create an Item Named P1_DEPT

To create the item named `P1_DEPT`:

1. Under Items, click the **Create** icon.
2. For Item Type, select **Select List** and click **Next**.
3. For Select List Control Type, select **Select List**, and click **Next**.
4. For Display Position and Name:
   a. For Item Name - Enter `P1_DEPT`.
   b. For Sequence - Accept the default.
   c. For Region - Select **Search**.
   d. Click **Next**.
5. For List of Values:
   a. Named LOV - Select **DEPARTMENT**.
   b. Null Text - Enter the following:

      ```
      - All -
      ```

   c. Accept the remaining defaults and click **Next**.
6. For Item Attributes:
   a. Label - Enter `Department`.
   b. Accept the remaining defaults and click **Next**.
7. Click **Create Item**.

Create an Item Named P1_MGR

To create the item named `P1_MGR`:

1. Under Items, click the **Create** icon.
2. For Item Type, select **Select List** and click **Next**.
3. For Select List Control Type, select **Select List** and click **Next**.
4. For Display Position and Name:
   a. For Item Name - Enter `P1_MGR`.
   b. For Sequence - Accept the default.
   c. For Region - Select **Search**.
   d. Click **Next**.
5. For List of Values:
   a. Named LOV - Select **MANAGER**.
   b. Null Text - Enter the following:

```
- All -
```

c. Accept the remaining defaults and click **Next**.

6. For Item Attributes:
   a. Label - Enter `Manager`.
   b. Accept the remaining defaults and click **Next**.
7. Click **Create Item**.

Adding a Button to Submit the Page

Why do you need to submit the page? Once the user enters search criteria, the page needs to be submitted so that the query will be rerun against that criteria. To submit the page, you add a button.

To add a button to submit the page:

1. Under Buttons, click the **Create** icon.
2. For Button Region, select **Search** and click **Next**.
3. For Button Position, select **Create a button displayed among this region's items** and click **Next**.
4. In Button Name, enter `P1_GO`.
5. Accept the remaining defaults and click **Create Button**.

   The Page Definition for page 1 appears.

> **See Also:** "Creating Buttons" in *Oracle Application Express Application Builder User's Guide*

Adding an Onload Process

Next, you create a process that sets the value for the hidden item `P1_TEXT`. The value of `P1_TEXT` determines the region title that displays. By adding this process, the region title displays as:

`Employees in` *`department_name`*

If you choose to not add this process, the region header simply displays as `Employees`.

To add an onload process:

1. Under Page Processing, Processes, click the **Create** icon.
2. For Process Type, select **PL/SQL** and click **Next**.
3. For Process Attributes:
   a. Name - Enter `get region title info`.
   b. Accept the remaining defaults and click **Next**.
4. For Process, enter the following SQL query:

```
DECLARE
   l_dept varchar2(100);
   l_mgr varchar2(100);
BEGIN
:P1_TEXT := null;
```

```
IF :P1_DEPT != '%null%'
   THEN SELECT department_name
          INTO l_dept
          FROM oehr_departments
         WHERE department_id = :P1_DEPT;
        :P1_TEXT := :P1_TEXT || ' in Department ' || l_dept;
END IF;

IF :P1_MGR != '%null%'
   THEN SELECT first_name || ' ' || last_name
          INTO l_mgr
          FROM oehr_employees
         WHERE employee_id = :P1_MGR;
        :P1_TEXT := :P1_TEXT ||' reporting to ' || l_mgr;
END IF;
END;
```

5. Click **Create Process**.

The Page Definition for page 1 appears.

Running the Page

To run the page:

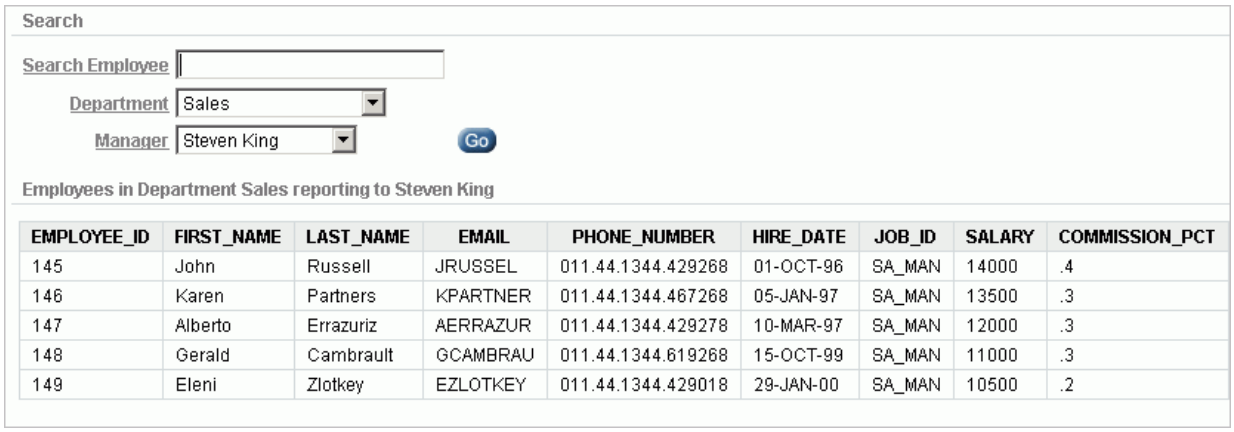
1. Click the **Run Page** icon in the upper right corner as shown in Figure 3–3.

Figure 3–3 Run Page Icon

2. If prompted to enter a user name and password, enter your workspace user name and password and click Login. See "About Application Authentication" on page 1-6.
3. When the Employees page appears, make a selection from the Department or Manager lists and click **Go**.

An Employees report appears as shown in Figure 3–4 on page 3-9.

Figure 3–4 Form Results Being Populated from Select Lists



EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT
145	John	Russell	JRUSSEL	011.44.1344.429268	01-OCT-96	SA_MAN	14000	.4
146	Karen	Partners	KPARTNER	011.44.1344.467268	05-JAN-97	SA_MAN	13500	.3
147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	10-MAR-97	SA_MAN	12000	.3
148	Gerald	Cambrault	GCAMBRAU	011.44.1344.619268	15-OCT-99	SA_MAN	11000	.3
149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29-JAN-00	SA_MAN	10500	.2

4

How to Create a Drill Down Report

A drill down report is a type of report that displays summary data with links to related detail data in other reports.

In this tutorial, you create a report on the `OEHR_ORDERS` table that contains links to drill down to additional data in the `OEHR_ORDER_ITEMS` table. Additionally, you also learn how to change the format of a column by editing column attributes.

Before you begin, you need to import and install the *OEHR Sample Objects* application in order to access the necessary sample database objects. See "About Loading Sample Objects" on page 1-2.

This section contains the following topics:

Creating a New Application

First, create a new application.

To create an application:

1. On the Workspace home page, click the **Application Builder** icon.

   The Application Builder home page appears.
2. Click **Create**.
3. Select **Create Application** and click **Next**.
4. For Name, specify the following:
   a. Name - Enter `Drilldown Reports`.
   b. Application - Accept the default.
   c. For Create Application - Accept the default, **From scratch**.
   d. For Schema - Select the schema where you installed the OEHR sample objects.
   e. Click **Next**.

      Next, you need to add pages. You have the option of adding a blank page, a report, a form, a tabular form, or a report and form. For this exercise, you add two blank pages.

5. First, add the first blank page. Under Add Page:
   a. Select Page Type - Accept the default, **Blank**.
   b. Page Name - Enter `Orders`.
   c. Click **Add Page**.
6. Add the second blank page. Under Add Page:
   a. Select Page Type - Select **Blank**.
   b. Subordinate to Page - Accept the default.
   c. Page Name - Enter `Order Items`.
   d. Click **Add Page**.

   The two new pages appear at the top of the page.
7. Click **Next**.
8. For Tabs, accept the default, **One Level of Tabs,** and click **Next**.
9. For Copy Shared Components from Another Application, accept the default, **No**, and click **Next**.
10. For Attributes, accept the defaults for Authentication Scheme, Language, and User Language Preference Derived From and click **Next**.
11. For User Interface, select **Theme 2** and click **Next**.
12. Review your selections and click **Create**.

    The Application home page appears. Note that your application contains three pages:
    - 1 - Orders
    - 2 - Order Items
    - 101 - Login

Creating Reports for OEHR_ORDERS and OEHR_ORDER_ITEMS

Next, you need to create reports for the `OEHR_ORDERS` and the `OEHR_ORDER_ITEMS` tables.

Topics in this section include:

- Create a Report for OEHR_ORDERS
- Create a Report for OEHR_ORDER_ITEMS

Create a Report for OEHR_ORDERS

To create a report on the `OEHR_ORDERS` table:

1. On the Application home page, click **Create Page**.
2. For page type, select **Report** and click **Next**.
3. Select **Wizard Report** and click **Next**.
4. For Page Attributes:
   a. Page Number - Enter **1**.
   b. Page Title - Enter `Orders`.

c. Region Title - Enter `Orders`.

d. For Region Template, accept the default.

e. For Breadcrumb, accept the default.

f. Click **Next**.

5. For Tables and Columns:

a. For Table/View Owner, select the default.

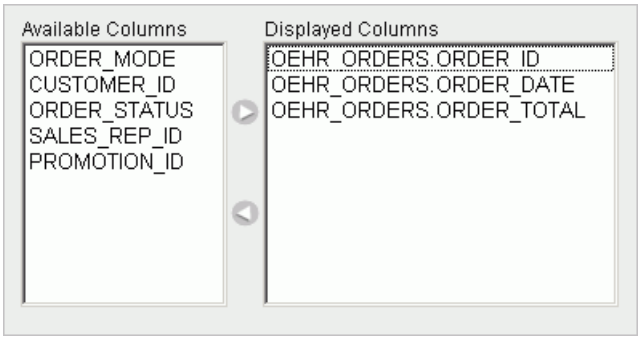
b. For Table/View, select **OEHR_ORDERS**.

The columns in the `OEHR_ORDERS` table appear.

c. From the Available Columns list, press **Ctrl** to select and move the following columns to the Displayed Columns list as shown in Figure 4–1:

`ORDER_ID, ORDER_TOTAL, ORDER_DATE`

Figure 4–1 Selected Columns



Next, create a join with the `OEHR_CUSTOMERS` table to display the customer name. First, select the table.

d. From the Table/View list, select **OEHR_CUSTOMERS**.

The columns in the `OEHR_CUSTOMERS` table appear.

e. From the Available Columns list, select **CUST_LAST_NAME** and move it to the Displayed Columns list.

f. Click **Next**.

6. For Join Conditions, accept the defaults and click **Next**.

7. For Report Options, accept the defaults and click **Next**.

8. Click **Create Report Page**.

9. Run the page by clicking the **Run Page** icon. If prompted for a user name and password, enter your workspace user name and password and click **Login**. See "About Application Authentication" on page 1-6.

As shown in Figure 4–2, a report on the `ORDERS` table appears.

Figure 4–2 Report on OEHR_ORDERS Table

Order Id	Order Date	Cust Last Name	Order Total
2408	29-JUN-99 07.59.31.333617 AM	Alexander	309
2389	04-JUN-00 04.49.43.546954 PM	Aykroyd	17620
1108	21-JUL-06 12.00.00.000000 AM	Bogart	0
2400	10-JUL-99 12.34.29.559387 AM	Boyer	69286.4
2376	07-JUN-99 05.18.08.883310 AM	Brown	11006.2
2444	27-JUL-99 01.22.27.462632 PM	Cage	77727.2

Note the report displays four columns: Order Id, Order Date, Cust Last Name, and Order Total. Also notice the format of the Order Date and Order Total Columns. Also note that your data might appear in a different order. You can ignore this difference and continue with the steps.

Next, you will change the format of these two columns.

Change the Format of Order Date Column

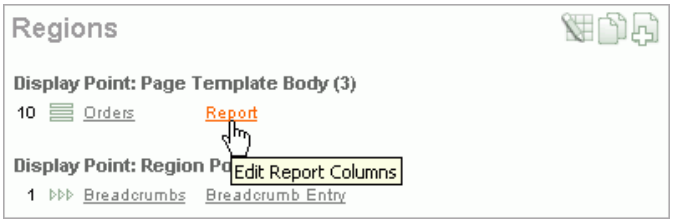
To change the format of the Order Date column:

1. Click **Edit Page 1** on the Developer toolbar.

   The Page Definition for page 1 appears.

2. Under Regions, click the **Report** link as shown in Figure 4–3.

Figure 4–3 Report Link



   The Report Attributes page appears.

3. Click the **Edit** icon next to ORDER_DATE. The Edit icon resembles a small page with a pencil on top of it.

   The Column Attributes page appears.

4. Locate the section Column Formatting.

5. From Number/Date Format, select the date format DD-MON-YYYY (for example, 12-JAN-2004).

6. Click **Apply Changes**.

   The Report Attributes page appears.

   Next, change the format of the Order Total Column.

Change the Format of Order Total Column

To change the format of the Order Total column:

1. Click the **Edit** icon next to ORDER_TOTAL.

   The Column Attributes page appears.

2. Locate the section Column Formatting.
3. From Number/Date Format, select the number format that includes a dollar sign (for example, $5, 234.10).
4. Click **Apply Changes**.
5. Run the page by clicking the **Run Page** icon in the upper right corner as shown in Figure 4–4.

Figure 4–4 Run Page Icon

As shown in Figure 4–5, the revised report on the `OEHR_ORDERS` table appears.

Figure 4–5 Report on OEHR_ORDERS Table with New Column Formats

Order Id	Order Date	Cust Last Name	Order Total
2408	29-JUN-1999	Alexander	$309.00
2389	04-JUN-2000	Aykroyd	$17,620.00
1108	21-JUL-2006	Bogart	$0.00
2400	10-JUL-1999	Boyer	$69,286.40
2376	07-JUN-1999	Brown	$11,006.20
2444	27-JUL-1999	Cage	$77,727.20
2421	12-MAR-1999	Cage	$72,836.00

6. Click **Application** on the Developer toolbar to return to the Application home page.

Create a Report for OEHR_ORDER_ITEMS

To create a report on the `OEHR_ORDER_ITEMS` table:

1. On the Application home page, click **Create Page**.
2. For Page, select **Report** and click **Next**.
3. On Create Page, select **Wizard Report** and click **Next**.
4. For Page Attributes:
   a. Page Number - Enter `2`.
   b. Page Title - Enter `Order Items`.
   c. Region Title - Enter `Order Items`.
   d. For Region Template, accept the default.
   e. For Breadcrumb, accept the default.
   f. Click **Next**.
5. For Tables and Columns:
   a. Table/View Owner - Accept the default.
   b. Table/View - Select **OEHR_ORDER_ITEMS**.

      The columns in the `OEHR_ORDER_ITEMS` table appear.
   c. From the Available Columns list, press **Ctrl** and move the following columns to the Displayed Columns list:

```
ORDER_ITEM_ID, ORDER_ID, UNIT_PRICE, QUANTITY
```

Next, create a join with the OEHR_PRODUCT_INFORMATION table to display the product name.

d. For Show Only Related Tables, select **No**.

Next, select the table.

e. From the Table/View list, select **OEHR_PRODUCT_INFORMATION**.

The columns in the OEHR_PRODUCT_INFORMATION table appear.

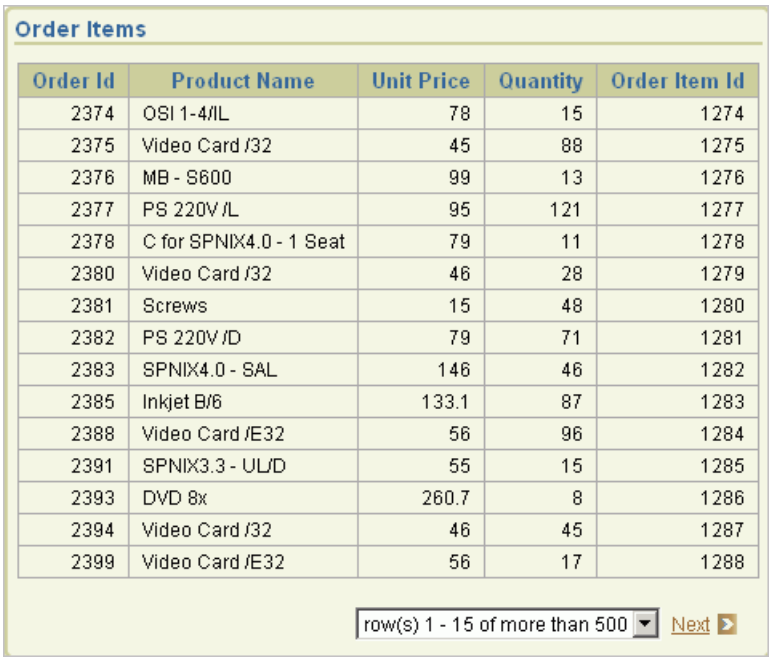
f. From the Available Columns list, select **PRODUCT_NAME** and move it to the Displayed Columns list.

g. Click **Next**.

6. For Join Conditions, accept the defaults and click **Next**.
7. For Report Options, accept the defaults and click **Next**.
8. Click **Create Report Page**.
9. Click **Run Page**.

As shown in Figure 4–6, a report on the OEHR_ORDER_ITEMS table appears.

Figure 4–6 Report on OEHR_ORDER_ITEMS Table



Order Items

Order Id	Product Name	Unit Price	Quantity	Order Item Id
2374	OSI 1-4/IL	78	15	1274
2375	Video Card /32	45	88	1275
2376	MB - S600	99	13	1276
2377	PS 220V /L	95	121	1277
2378	C for SPNIX4.0 - 1 Seat	79	11	1278
2380	Video Card /32	46	28	1279
2381	Screws	15	48	1280
2382	PS 220V /D	79	71	1281
2383	SPNIX4.0 - SAL	146	46	1282
2385	Inkjet B/6	133.1	87	1283
2388	Video Card /E32	56	96	1284
2391	SPNIX3.3 - UL/D	55	15	1285
2393	DVD 8x	260.7	8	1286
2394	Video Card /32	46	45	1287
2399	Video Card /E32	56	17	1288

row(s) 1 - 15 of more than 500 Next

Customizing the ORDER_ITEMS Report

Next, you need to customize the Order Items page. In this exercise, you add an item to hold the value of the ORDER_ID, add a WHERE clause that constrains the report by the value of ORDER_ID item, and modify the Region Title to note which order is being viewed.

Topics in this section include:

- Add an Item to Hold the Value of ORDER_ID

- Add a Where Clause to Restrict the Report
- Modify the Region Title

Add an Item to Hold the Value of ORDER_ID

To create an item to hold the value of ORDER_ID:

1. Click **Edit Page 2** on the Developer toolbar.

   The Page Definition appears.

2. Under Items, click the **Create** icon as shown in Figure 4–7.

Figure 4–7 Create Icon

3. For Item Type, select **Hidden** and click **Next**.
4. For Display Position and Name:
   a. Item Name - Enter `P2_ORDER_ID`.
   b. Sequence - Accept the default.
   c. Region - Select **Order Items**.
   d. Click **Next**.
5. Click **Create Item**.

Add a Where Clause to Restrict the Report

Next, you add a `WHERE` clause that constrains the report by the value of ORDER_ID item.

To add a `WHERE` to the ORDER_ITEMS report:

1. Under Regions, select **Order Items** as shown in Figure 4–8.

Figure 4–8 Order Items

2. Click the **Query Definition** tab.
3. Click **Modify Join Conditions**.
4. On the Modify Join Conditions page:
   a. For the first Column, select `OEHR_ORDER_ITEMS.ORDER_ID`. Note that you may need to click **Search** to view available columns.

b. In the second Column field, replace the existing text with the following condition:

```
:P2_ORDER_ID
```

5. Click **Apply Changes**.

Modify the Region Title

To modify the region title of the ORDER_ITEMS report:

1. Under Regions, click **Order Items**.
2. In Title, replace the existing text with the following:

```
Order Items for Order # &P2_ORDER_ID.
```

3. Click **Apply Changes**.

Linking the ORDERS Report to the ORDER_ITEMS Report

Lastly, you link the OEHR_ORDERS report to the OEHR_ORDER_ITEMS report. To accomplish this, you must edit the attributes of the `ORDER_ID` column on the OEHR_ORDERS report and create a link. The link will populate the P2_ORDER_ID hidden item on page 2 with the clicked ORDER_ID.

To create a link from the `ORDER_ID` column on the OEHR_ORDERS report to the OEHR_ORDER_ITEMS report:

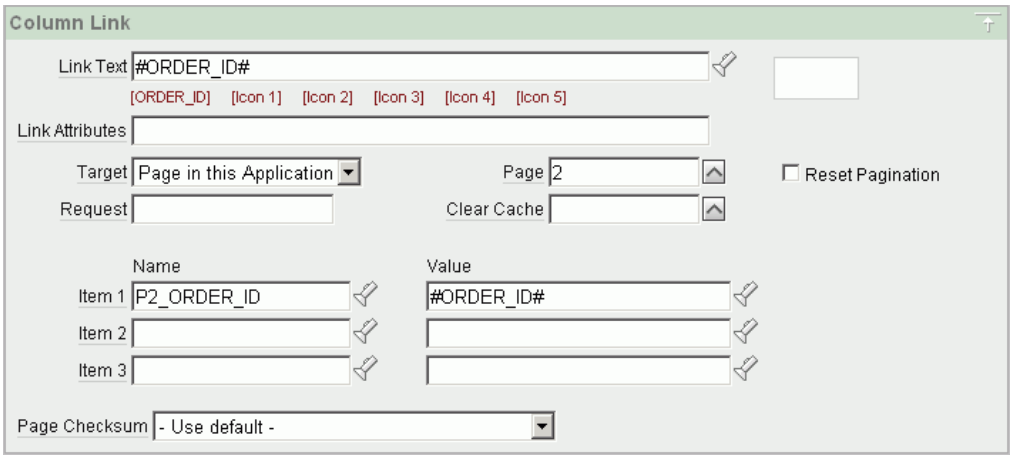
1. On the Page Definition, enter `1` in the Page field in the Navigation bar and click **Go**.
2. Under Regions, click **Orders**.
3. Click the **Report Attributes** tab.
4. Click the **Edit** icon next to ORDER_ID.
5. Scroll down to Column Link.
   a. In the Page field, select **2 Order Items**.

      Next, populate the P2_ORDER_ID hidden item on page 2 with the clicked ORDER_ID.

   b. From Item 1 Name, select **P2_ORDER_ID**.
   c. From Item 1 Value, select **#ORDER_ID#**.
   d. For Link Text, select **#ORDER_ID#**.

   Your Column Link attributes should resemble Figure 4–9 on page 4-9.

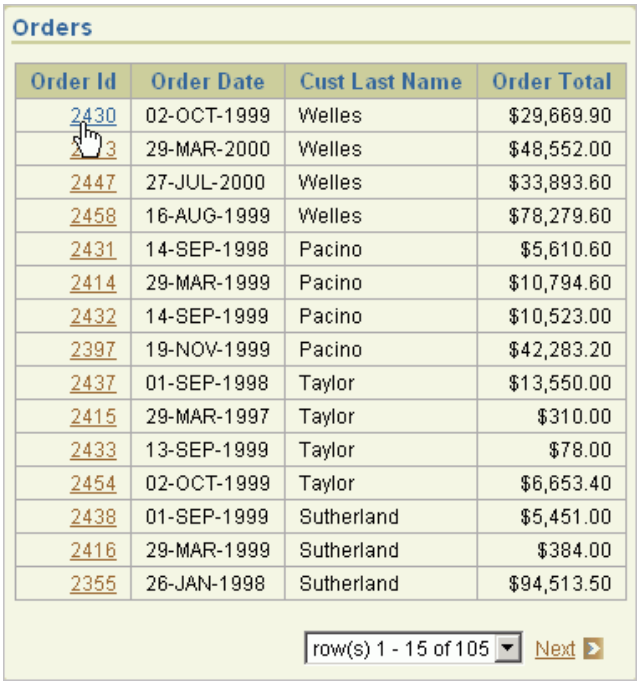
Figure 4–9 Column Link Attributes for the ORDER_ID Column



6. Scroll to the top of the page and click **Apply Changes**.
7. Click the **Run Page** icon in the upper right corner of the page.

   As shown in Figure 4–10, you can link to page 2 by clicking an Order Id.

Figure 4–10 OEHR_ORDERS Report with Link to Page 2



Orders

Order Id	Order Date	Cust Last Name	Order Total
2430	02-OCT-1999	Welles	$29,669.90
2403	29-MAR-2000	Welles	$48,552.00
2447	27-JUL-2000	Welles	$33,893.60
2458	16-AUG-1999	Welles	$78,279.60
2431	14-SEP-1998	Pacino	$5,610.60
2414	29-MAR-1999	Pacino	$10,794.60
2432	14-SEP-1999	Pacino	$10,523.00
2397	19-NOV-1999	Pacino	$42,283.20
2437	01-SEP-1998	Taylor	$13,550.00
2415	29-MAR-1997	Taylor	$310.00
2433	13-SEP-1999	Taylor	$78.00
2454	02-OCT-1999	Taylor	$6,653.40
2438	01-SEP-1999	Sutherland	$5,451.00
2416	29-MAR-1999	Sutherland	$384.00
2355	26-JAN-1998	Sutherland	$94,513.50

5

How to Control Form Layout

Data and form elements in an Oracle Application Express application are placed on a page using containers called regions. There are several attributes that control the placement and positioning of regions on pages. In turn, you control the placement and style of form elements (called items) inside of regions using item attributes.

In this tutorial, you create the underlying data objects for a data input form by running a script. Then, you create a data input form and learn how to change the form layout by altering region and item attributes.

This section contains the following topics:

Creating a Table and Data Input Form

The first step in creating a data input form is to create the underlying data objects. In this exercise, you create a table named `HT_EMP` and then use a wizard to create a new page.

Topics in this section include:

Create the HT_EMP Table

First, you create a new table by running a script in SQL Scripts.

To create the HT_EMP table and the appropriate associated objects:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.

   The SQL Scripts home page appears.

2. Click **Create**.

   The Script Editor appears.

3. In Script Name, enter HT_EMP.

4. In the Script Editor, enter the following data definition language (DDL):

```
CREATE TABLE ht_emp (
   emp_id                 NUMBER       primary key,
   emp_first_name         VARCHAR2(30) not null,
   emp_middle_initial     VARCHAR2(1),
   emp_last_name          VARCHAR2(45) not null,
   emp_part_or_full_time  VARCHAR2(1)  not null check (emp_part_or_full_time in
('P','F')),
   emp_salary             NUMBER,
   emp_dept               VARCHAR2(20) check (emp_dept in
('SALES','ACCOUNTING',

'MANUFACTURING','HR')),
   emp_hiredate           DATE,
   emp_manager            NUMBER       references ht_emp,
   emp_special_info       VARCHAR2(2000),
   emp_telecommute        VARCHAR2(1) check (emp_telecommute in ('Y')),
   rec_create_date        DATE         not null,
   rec_update_date        date)
/

INSERT INTO ht_emp
   (emp_id, emp_first_name, emp_middle_initial, emp_last_name, emp_part_or_
full_time, emp_salary, emp_dept, emp_hiredate, emp_manager, emp_special_info,
emp_telecommute, rec_create_date)
VALUES
   (1,'Scott','R','Tiger','F',
    100000,'SALES',sysdate,null,'cell phone number is xxx.xxx.xxxx
home phone is yyy.yyy.yyyy','Y',
    SYSDATE)
/

CREATE SEQUENCE ht_emp_seq
   start with 2
/

CREATE OR REPLACE TRIGGER bi_ht_emp
      BEFORE INSERT ON ht_emp
      FOR EACH ROW
   BEGIN
      SELECT ht_emp_seq.nextval
        INTO :new.emp_id
        FROM DUAL;
      :new.rec_create_date := SYSDATE;
   END;
/
```

```
CREATE OR REPLACE TRIGGER bu_ht_emp
      BEFORE UPDATE ON ht_emp
      FOR EACH ROW
   BEGIN
      :new.rec_update_date := SYSDATE;
   END;
/
```

5. Click **Save**.

   The script appears in the SQL Scripts Repository.

6. Run the HT_EMP script:

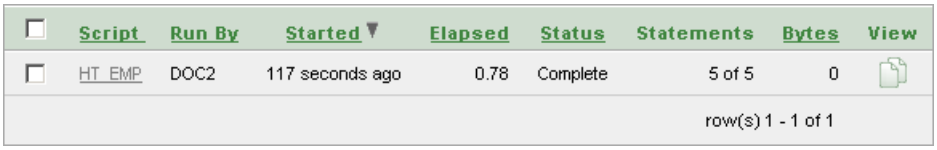
   a. Click the **HT_EMP** script.

   b. Click **Run**.

   c. On the Run Script page, click **Run** again.

7. From the View list, select **Details** and click **Go**.

Figure 5–1 Details View in the Manage Script Results Page



8. Click the **View Results** icon to access the Results page.

   Red text indicates errors while executing the file.

> **See Also:** "Using SQL Scripts" in *Oracle Application Express Application Builder User's Guide*.

Create a New Application

Next, create a new application.

To create an application:

1. Return to the Workspace home page. Click the **Home** breadcrumb link at the top of the page.
2. On the Workspace home page, click the **Application Builder** icon.

   The Application Builder home page appears.

3. Click **Create**.
4. Select **Create Application** and click **Next**.
5. For Name, specify the following:

   a. Name - Enter `Form Layout`.

   b. Application - Accept the default.

   c. Create Application - Accept the default, **From scratch**.

   d. Schema - Select the schema where you installed the OEHR sample objects.

   e. Click **Next**.

Next, you need to add pages. You have the option of adding a blank page, a report, a form, a tabular form, or a report and form. For this exercise, you add two blank pages.

6. Add a blank page:
   a. Under Select Page Type, accept the default, **Blank**.
   b. Click **Add Page**.

      The blank page appears at the top of the page.
7. Click **Next**.
8. For Tabs, accept the default, **One Level of Tabs,** and then click **Next**.
9. For Copy Shared Components from Another Application, accept the default, **No**, and click **Next**.
10. For Attributes, accept the defaults for Authentication Scheme, Language, and User Language Preference Derived From and then click **Next**.
11. For User Interface, select **Theme 12** and then click **Next**.
12. Review your selections and click **Create**.

    The Application home page appears. Note that your application contains two pages.

Create a New Page Containing an Input Form

Next, create a new form using the Form on a Table or View Wizard.

To create a data input form:

1. On the Application home page, click **Create Page**.
2. For Page, select **Form** and click **Next**.
3. On Create Page, select **Form on a Table or View** and click **Next**.
4. For Table/View Owner, accept the default and click **Next**.
5. For Table/View Name, select the **HT_EMP** table and click **Next**.
6. For Page and Region Attributes:
   a. Page Number - Enter `2`.
   b. Page Name - Enter `Form Layout`.
   c. Region Title - Enter `Form Layout`.
   d. Region Template - Accept the default.
   e. Breadcrumb - Accept the default.
   f. Click **Next**.
7. For Tab, accept the default, **Do not use tabs,** and click **Next**.
8. For Primary Key, accept the default and click **Next**.

   Note that the wizard reads the primary key from the database definition.
9. For Source Type, accept the default **Existing Trigger** and click **Next**.
10. For Select Columns, select all the columns. Press **SHIFT**, select the first column and then the last one. Then, click **Next**.

11. For Process Options, accept the defaults and click **Next**.
12. For Branching, enter 2 (the page you are creating) in both fields and click **Next**.

    Since this page is just for demonstration, you will not be utilizing branching.
13. Click **Finish**.
14. Click the **Edit Page** icon.

    The Page Definition for page 2 appears.
15. Delete the following validation:
    - Under Page Processing, Validations, select **P2_REC_CREATE_DATE not null**.
    - Click **Delete**.

    You are removing this validation because the value of this column is set using a trigger. The item will have no value in the form for a new record. This validation was automatically created because the underlying database column is not null.

Run the Page

Once you create a new input form, the next step is to run the page.

To run the page from the Page Definition:

1. Click the **Run Page** icon in the upper right corner as shown in Figure 5–2.

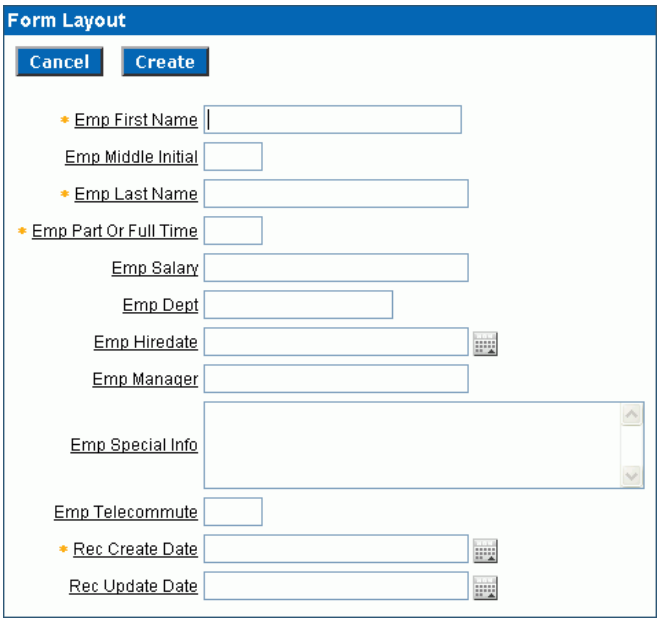
Figure 5–2 Run Page Icon

2. If prompted for a user name and password, enter your workspace user name and password and click **Login**. See "About Application Authentication" on page 1-6.

   The application appears.

As shown in Figure 5–3 on page 5-6, the new form appears. Note that the form contains basic employee details and includes select lists, text areas, and display only items.

Figure 5–3 Employee Info Form



By default, the Primary Key column does not display since it is assumed that the primary key is system generated. In reality, the primary key is included in the page, but appears as a hidden item.

Notice that the page defaults with one item for each row and labels display to the left of the items. The item labels default to the column names with initial capitalization and with the underscores (_) replaced with spaces. You can override this default behavior by configuring user interface defaults for the table.

> **See Also:** "Managing User Interface Defaults" in the *Oracle Application Express Application Builder User's Guide*.

Also notice that items based on date columns default to include a date picker. Lastly, notice that the Emp Special Info item was created as a text area because of the size of the base column. This item type is a better choice for large text items since it allows the input text to wrap.

Changing the Appearance of a Page by Altering Region Attributes

A region is an area on a page that serves as a container for content. You can alter the appearance of a page by changing the region attributes.

Topics in this section include:

- Edit the Region Title
- Change the Display Point and Template
- Change the Region Attributes Back to the Original Selections

Edit the Region Title

To edit the region title and other region level attributes:

1. Click **Edit Page 2** on the Developer toolbar.
2. Under Regions, click **Form Layout**.

The Region Definition appears.

3. Change the Title to `Employee Info`.
4. Click **Apply Changes** at the top of the page.
5. From the Page Definition, click the **Run Page** icon in the upper right corner.

   Note the new region title.

Change the Display Point and Template

To change other region attributes:

1. Return to the Page Definition. Click **Edit Page 2** on the Developer toolbar.
2. Under Regions, click **Employee Info**.

   The Region Definition appears.
3. Scroll down to User Interface.

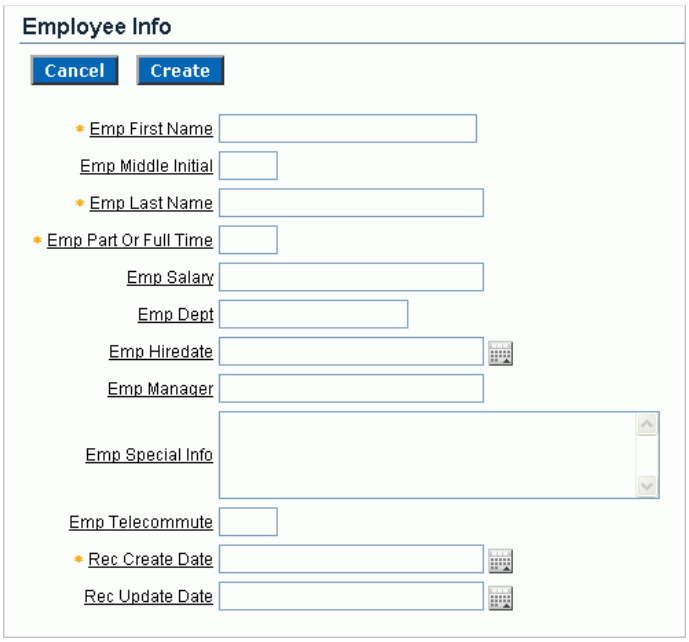
   You can control the position of a region within a page template by changing the page template.
4. From Display Point, select **Page Template Region Position 3**.

   In this instance, your selection moves the region to the right side of the page. The region display points are determined by the page level template. If you do not select a page level template, Oracle Application Express uses the default page level template defined in the current theme. You can view specific positions by selecting the flashlight icon to the right of the Display Point list.

   Next, temporarily change the region template.
5. From Template, select **Borderless Region**.
6. Click **Apply Changes** at the top of the page.
7. From the Page Definition, click the **Run Page** icon in the upper right corner.

   The form appears as shown in Figure 5–4.

Figure 5–4 Employee Info Form with New Display Point and Template



Note the changes in the appearance of the form. The form title displays in a dark color on a white background and has a rule beneath it. Also, note that there is no longer a border around the form.

Change the Region Attributes Back to the Original Selections

To return to the region template and display point to the original selections:

1. Click **Edit Page 2** on the Developer toolbar.
2. Under Regions, select **Employee Info**.
3. Scroll down to User Interface.
4. From the Template list, select **Form Region**.
5. From the Display Point List, select **Page Template Body (3. Items above region content)**.
6. Click **Apply Changes** at the top of the page.

Understanding How Item Attributes Affect Page Layout

An item is part of an HTML form. An item can be a text field, text area, password, select list, check box, and so on. Item attributes control the display of items on a page. Item attributes determine where a label displays, how large an item will be as well as whether the item displays next to or below a previous item. You can change multiple item labels at once on the Page Items summary page.

Topics in this section include:

- Edit Item Attributes
- Fix Item Alignment
- Change a Region to Display-only

See Also: "Creating Items" in *Oracle Application Express Application Builder User's Guide*.

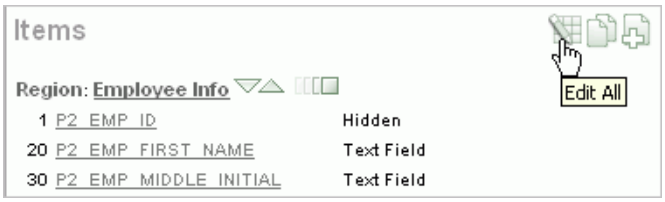
Edit Item Attributes

To edit all item attributes:

1. On the Page Definition for page 2, locate the Items section.
2. Under Items, click the **Edit All** icon.

   The Edit All icon resembles a small grid with a pencil on top of it as shown in Figure 5–5.

Figure 5–5 Edit All Icon



   The Page Items summary page appears.

   You change how a page appears by editing the item attributes. Common item attribute changes include:

   - Changing item labels by editing the Prompt field.
   - Placing more than one item in certain rows to group like items together. For example, you could group all items that make up an employee's name.
   - Changing the item width. Many items display better when they have a width that is less than the maximum. To change the item width, enter a new value in the Width field.
   - Reordering the items. The initial order of items is based on the order of the columns in the table on which the region is based. To reorder items, enter a new value in the Sequence field.

3. To see the how item attributes affect page layout, make the following changes:

   a. Change the values in the Prompt, New Line, and Width fields to match those in Table 5–1:

Table 5–1 New Prompt, New Line, and Width Field Values

Prompt Field	New Line	Width
Emp ID	Yes	30
First Name	Yes	15
Middle Initial	No	2
Last Name	No	15
Part or Full Time	Yes	2
Salary	Yes	10
Department	Yes	15
Hire Date	Yes	10

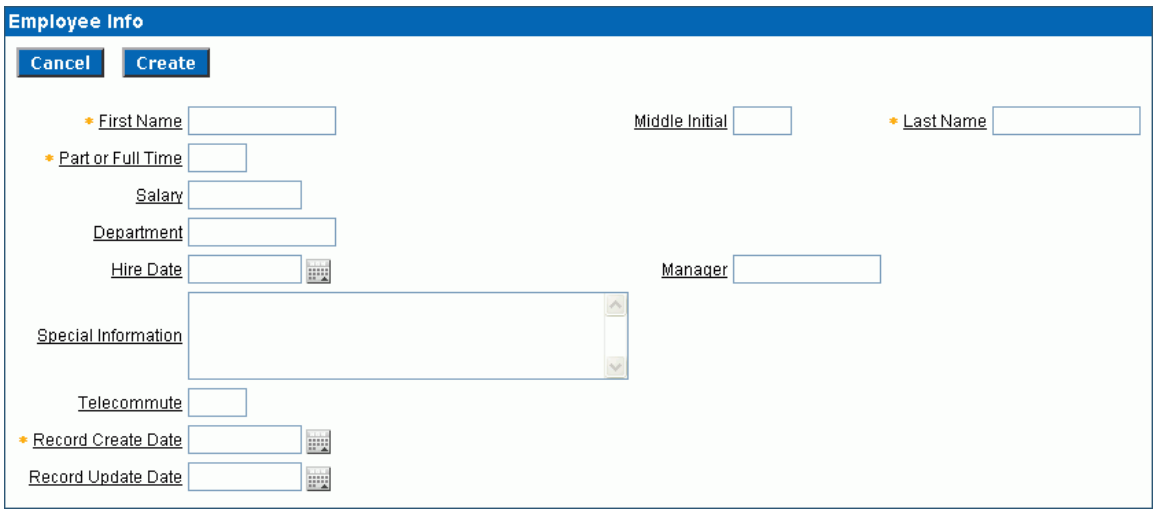
Table 5–1 (Cont.) New Prompt, New Line, and Width Field Values

Prompt Field	New Line	Width
Manager	No	15
Special Information	Yes	60
Telecommute	Yes	2
Record Create Date	Yes	10
Record Update Date	Yes	10

b. Click **Apply Changes**.

c. Click the **Run Page** icon in the upper right corner of the page.

Figure 5–6 Employee Info Form After Editing the Prompt, New Line, Width Attributes



As shown in Figure 5–6, note that some items are pushed too far to the right because of the width of the Special Information item. Oracle Application Express lays out regions as tables, and the width of each column is determined by the largest display width of the items in that column.

Fix Item Alignment

There are several approaches to fixing item alignment:

- For the items Middle Initial, Last Name and Manager items, set New Field to equal No.

  This places the items directly after the items they follow, but in the same column. This approach is best used for positioning embedded buttons next to items. Note that this setting can make text items appear squashed.

- Change the Column Span field of the Special Information item.

  For example, setting the Column Span for the Special Information item to 5 would enable multiple items to display above and below it. This change causes five items to display above Special Information (First Name, Middle Initial, and Last Name).

  Be aware, however, that using Column Span to fix the display of the name does not result in a consistent layout. The Manager item would still be in the same

column as Middle Initial. Because the Manager item is larger than Middle Initial, Last Name would still be pushed too far to the right. To fix this, you could change the Column Span of the Manager item to 3 so it displays above Special Information.

- Reset the column width in the middle of the region by adding an item of type **Stop and Start HTML Table.** This forces the close of an HTML table using the `</table>` tag and starts a new HTML table. Inserting a Stop and Start HTML Table item just after the Last Name item results in an even layout. Note that a Stop and Start HTML Table item only displays its label. You can prevent the label from displaying at all by setting it to null. To do this, you simply remove the defaulted label.

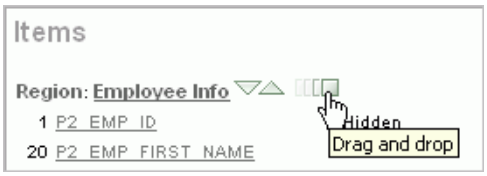
Add a Stop and Start HTML Table Item

You can use the Drag and Drop Layout page to interactively reorder items within a given region, change select item attributes, create new items, or delete existing items. For this exercise, you will reorder two items.

To add a Stop and Start HTML Table item:

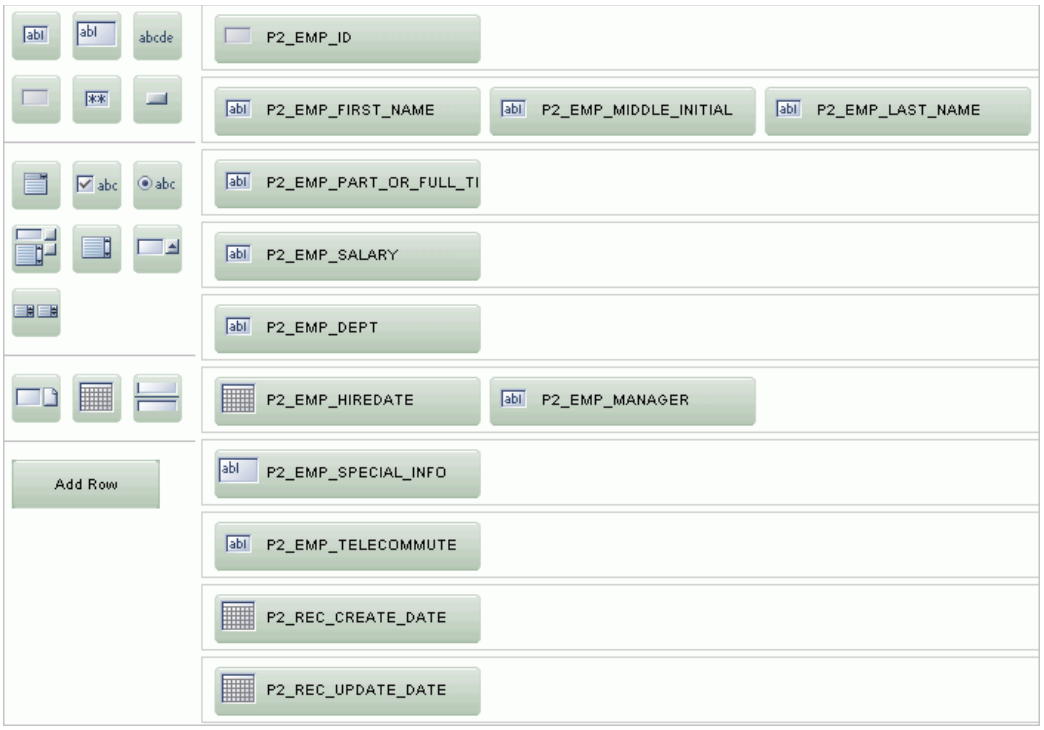
1. Click **Edit Page 2** on the Developer toolbar.
2. Under Items, click the **Drag and drop** icon as shown in Figure 5–7.

Figure 5–7 Drag and Drop Icon



The Drag and Drop Layout page appears. This page is divided into the Item palette on the left and the Layout region on the right.

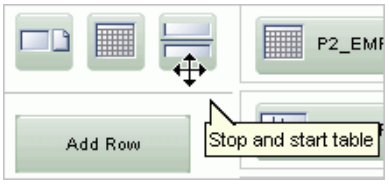
Figure 5–8 Drag and Drop Layout Page



3. In the Item palette, select the **Stop and start table** icon and drop it after the row containing the `P2_EMP_ID` item:

   a. Move your mouse over the icons in the Item palette and locate the Stop and start table. Note that when you position the cursor over an item type, a tooltip appears.

Figure 5–9 Stop and Start Table Item Type

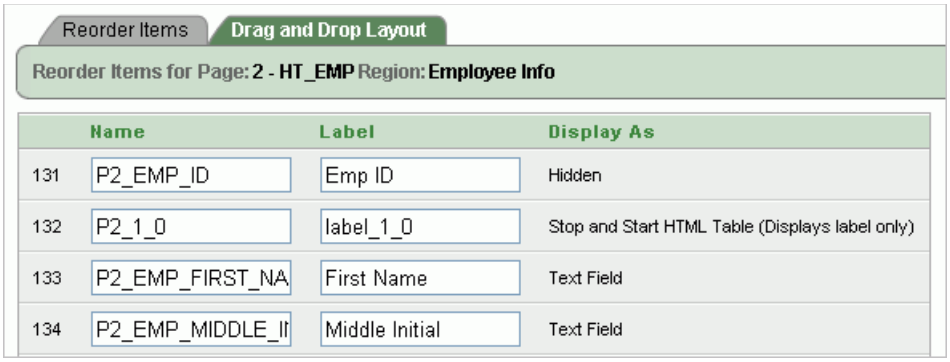


   b. Click the Stop and start table icon and drag it beneath of `P2_EMP_ID`.

   c. Click **Next**.

      A report appears. Note that the new Stop and Start HTML Table appears beneath `P2_EMP_ID`.

Figure 5–10 Revised Drag and Drop Layout Page



4. Click **Apply Changes**.

Edit the Special Information Item

Items are laid out in HTML tables. Next, you need to edit the Column Span attribute for the Special Information item. The Column Span attribute defines the value to be used for the `COLSPAN` attribute in the table cell.

To edit the Special Information item:

1. Under Items, click the **Reorder Region Items** icon.

   The Reorder Region Items icon resembles a light green down and up arrow as shown in Figure 5–11.

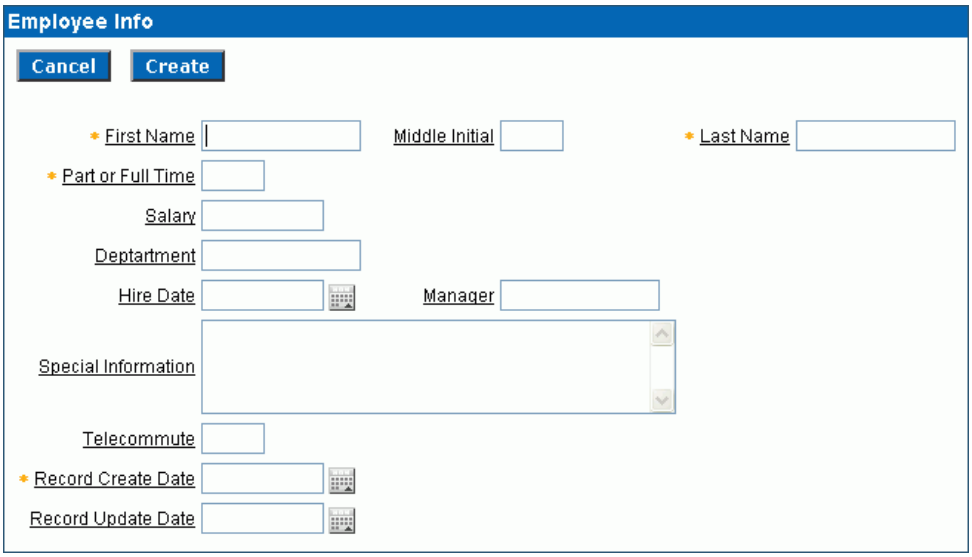
Figure 5–11 Reorder Region Items Icon

   The Reorder Region Items page appears in a separate window.

   Items on a page are laid out in tables. You can edit the position of an item by selecting values for the New Line, New Field, Column Span, and Label Alignment attributes. You can change the order in which items display by clicking the up and down arrows in the far right column. Also notice that a graphical representation of how the items display appears at the bottom of the page.

2. For Special Information, change the Column Span to **3** and click **Apply Changes**.
3. Click the **Run Page** icon in the upper right corner. The form appears as shown in Figure 5–12.

Figure 5–12 Employee Info Form with Corrected Item Alignment



Change a Region to Display-only

There are two columns in the HT_EMP table for auditing, Record Create Date and Record Update Date. Because the values of these columns are set with triggers, these columns should not be updatable by users. This exercise demonstrates how to make items display-only and then how to add them to their own region.

Topics in this section include:

- Change Items to Display-only
- Create a New Region
- Move Audit Items to the New Region
- Change the Region Template to Hide/Show Region

Change Items to Display-only

To make the item P2_REC_CREATE_DATE display-only:

1. Go to the Page Definition for page 2. Click **Edit Page 2** on the Developer toolbar.
2. Under Items, select the item **P2_REC_CREATE_DATE**.
3. From the Display As list in the Name section, select **Text Field (Disabled, saves state)**.
4. Click **Apply Changes**.

To make the item P2_REC_UPDATE_DATE display-only:

1. Go to the Page Definition for page 2.
2. Under Items, select the item **P2_REC_UPDATE_DATE**.
3. From the Display As list in the Name section, select **Text Field (Disabled, saves state)**.
4. Click **Apply Changes**.

   Next, create a new region and move the display-only items to the new region.

Create a New Region

To create a new region:

1. Go to the Page Definition for page 2.
2. Under Regions, click the **Create** icon.
3. For Region:
   a. Identify the type of region to add to this page - Accept the default, **HTML**, and click **Next**.
   b. Select the type of HTML region container you wish to create - Accept the default, **HTML**, and click **Next**.
4. For Display Attributes, enter `Audit Information` in the Title field, accept the remaining defaults, and click **Next**.
5. Click **Create Region**.

Move Audit Items to the New Region

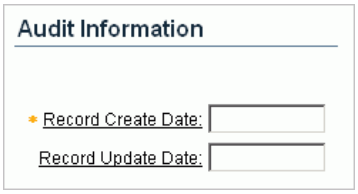
To move the items to the new region:

1. Go to the Page Definition for page 2.
2. Under Items, click the **Edit All** icon. The Edit All icon resembles a small grid with a pencil on top of it.

   The Page Items summary page appears.
3. For `P2_REC_CREATE_DATE` and `P2_REC_UPDATE_DATE`:
   a. In the Prompt column, add a colon to the end of the label name.
   b. In the Region column, select **Audit Information**.
   c. Click **Apply Changes**.
4. Click the **Run Page** icon in the upper right corner.

   Figure 5–13 on page 5-15 demonstrates how these items would display in a running page.

Figure 5–13 Audit Information Region



Change the Region Template to Hide/Show Region

The Hide/Show Region template enables the user to click a plus (+) sign to expand the contents of the region or click a minus (-) sign to hide the contents. By changing the region template to Hide/Show Region, users can decide whether they want to see the Audit Information region.

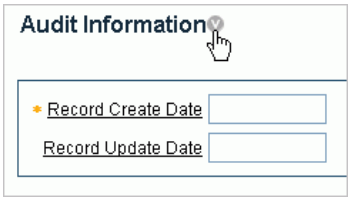
To change the region template to Hide/Show Region:

1. Go to the Page Definition for page 2.
2. Under Regions, click **Audit Information**.

3. Under User Interface, select **Hide and Show Region** from the Template list.
4. Click **Apply Changes**.
5. Click the **Run Page** icon in the upper right corner.

   Figure 5–14 demonstrates how Audit Information displays as a Hide/Show region. You can hide the region by clicking the icon to the right of the region title.

Figure 5–14 Audit Information as a Hide/Show Region



Adding a Region Header and Footer

Regions can have headers and footers. Headers and footers typically contain text or HTML that displays at either the top or bottom of the region.

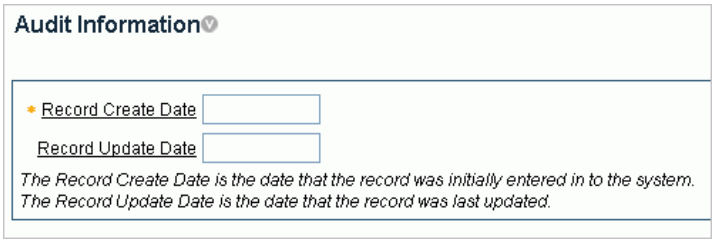
To add a region footer:

1. Go to the Page Definition for page 2. Click **Edit Page 2** on the Developer toolbar.
2. Under Regions, select **Audit Information**.
3. Scroll down to Header and Footer.
4. Enter the following in Region Footer:

```
<i>The Record Create Date is the date that the record was initially entered in
to the system. <br/>The Record Update Date is the date that the
record was last updated.</i>
```

5. Click **Apply Changes**.
6. Click the **Run Page** icon in the upper right corner.
7. Click the arrow next to Audit Information to display the two date fields and new footer text as shown in Figure 5–15.

Figure 5–15 Audit Information Region with Footer



As shown in Figure 5–15 on page 5-16, the text of the footer is wrapped with the italic HTML tag and there is an imbedded break. Without the manual break, the text would take up the entire width of the region (as defined by region template).

Making a Region Conditional

To make a region conditional, you create a display condition for the Audit Information region so that it only displays if the Employee ID is not null. Since the Employee ID is set by a trigger, it only exists for records retrieved from the database. You can control the display of the Audit Information region by using a built-in condition that checks for the presence of a value for the item containing the Employee ID (that is, P2_EMP_ID)

To display the Audit Information region conditionally:

1. Go to the Page Definition for page 2.
2. Under Regions, select **Audit Information**.

   The Region Definition appears.
3. Scroll down to Conditional Display.
4. Under Conditional Display:
   a. From Condition Type, select **Value of Item in Expression 1 is NOT NULL**.
   b. In Expression 1, enter:

      ```
      P2_EMP_ID
      ```
5. Click **Apply Changes**.

Adding a Region to Contain Hint Text

You have the option of displaying regions in columns as well as in rows. This exercise explains how to create another region to display explanatory text (hints) for the user.

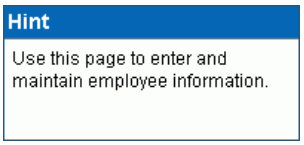
To create a region to display hint text:

1. Go to the Page Definition for page 2.
2. Under Regions, click the **Create** icon.
3. For Region:
   a. Identify the type of region to add to this page - Accept the default, **HTML**, and click **Next**.
   b. Select the type of HTML region container you wish to create - Accept the default, **HTML**, and click **Next**.
4. For Display Attributes:
   a. Title - Enter `Hint`.
   b. Region Template - Select **Sidebar Region**.
   c. Display Point - Select **Page Template Region Position 3**.
   d. Sequence - Accept the default.
   e. Column - Select **3**.
   f. Click **Next**.
5. In Enter HTML Text Region Source, enter the following:

   ```
   Use this page to enter and<br/>
   maintain employee information.
   ```
6. Click **Create Region**.

7. Click the **Run Page** icon. Figure 5–16 shows the new Hint region on the page.

Figure 5–16 Hint Region



Changing Item Types

This exercise describes how to change item types to make data entry easier for the user. To change an item type, go to the Item attributes page and select another Display As option.

Topics in this section include:

- Change an Item to a Radio Group
- Change an Item to a Select List
- Change an Item to a Check Box

> **Tip:** For simplicity, this tutorial has you alter items by editing item attributes. As a best practice, however, you can also create named LOVs and reference them.

> **See Also:** "Creating Lists of Values" in *Oracle Application Express Application Builder User's Guide*

Change an Item to a Radio Group

Because the Part or Full-time item only has two valid choices, this item is a good candidate for either a check box or a radio group.

To change the Part or Full-time item to a radio group:

1. Go to the Page Definition for page 2. Click **Edit Page 2** on the Developer toolbar.
2. Under Items, select **P2_EMP_PART_OR_FULL_TIME**.

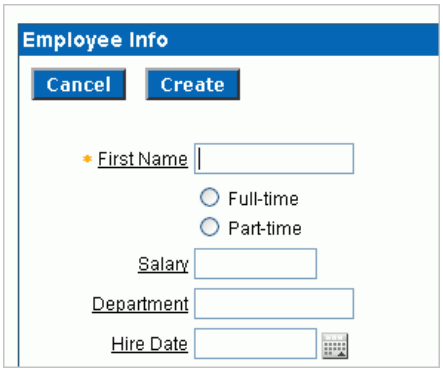
   The Edit Page Item page appears.
3. Locate the Name section. From the Display As list, select **Radiogroup**.
4. Locate the Label section. Specify the following:
   a. Label - Remove the text in the Label field.
   b. Template - Select **No Label**.
5. Under List of Values, create a static list of values. Specify the following:
   a. Named LOV - Select **Select Named LOV**.
   b. List of values definition - Enter:

   ```
   STATIC:Full-time;F,Part-time;P
   ```

   This definition will display as two radio buttons with the labels **Full-time** and **Part-time**, but the value being inserted into the database will be either F or P.
6. At the top of the page, click **Apply Changes**.

7. Click the **Run Page** icon in the upper right corner. The modified form appears as shown in Figure 5–17.

Figure 5–17 Part or Full-time item Changed to a Radio Group



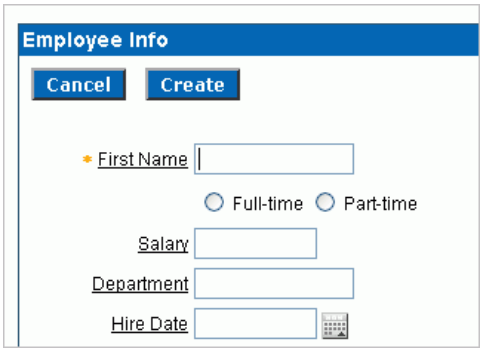
Notice that Full-time and Part-time displays as a radio group that is stacked in one column. You can have these buttons display side by side.

To display the Full-time and Part-time radio buttons side by side:

1. Go to the Page Definition for page 2.
2. Under Items, select **P2_EMP_PART_OR_FULL_TIME**.
3. Scroll down to List of Values.
4. In Number of Columns, enter **2**.
5. At the top of the page, click **Apply Changes**.
6. Click the **Run Page** icon.

   By changing this setting to match the number of valid values (that is, Full-time and Part-time), the values display side by side as shown in Figure 5–18.

Figure 5–18 Part or Full-time Item Displayed Side by Side



Change an Item to a Select List

In the DDL you used to create the `HT_EMP` table, the Department is validated by a check constraint. You can implement the Department as a radio group, a select list, or a Popup LOV.

To change Department to a select list:

1. Go to the Page Definition for page 2. Click **Edit Page 2** on the Developer toolbar

2. Under Items, select **P2_EMP_DEPT**.
3. From the Display As list in the Name section, select **Select List**.

   The other Select List choices are for either redirecting the user to another page or URL based on the selection, or submitting the current page which is used when other information needs to be retrieved based upon the selection in the Select List.
4. Scroll down to List of Values.
5. Under List of Values, create a static list of values. Specify the following:
   a. Named LOV - Select **Select Named LOV**
   b. List of values definition - Enter:

      ```
      STATIC:SALES,ACCOUNTING,MANUFACTURING,HR
      ```

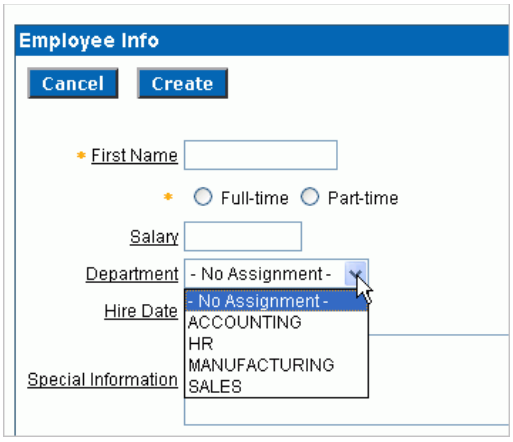
   c. From Display Null, select **Yes**.
   d. In Null display value, enter:

      ```
      - No Assignment -
      ```

      The last two selections take into account that the `EMP_DEPT` column can contain nulls. As a best practice, whenever you implement a select list and have a column that can be null, you should set Display Null to **Yes**. Failure to do so results in the item defaulting to the first item in the select list.
6. At the top of the page, click **Apply Changes**.
7. Click the **Run Page** icon.

   The revised form appears as shown in Figure 5–19.

Figure 5–19 Department Changed to a Select List



Change an Item to a Check Box

The item Telecommute is ideal for a check box. When you change the Display Type, you can also move it up on the page and place it next to the Full-time and Part-time radio group.

To change Telcommute to a check box:

1. Go to the Page Definition for page 2. Click **Edit Page 2** on the Developer toolbar.
2. Under Items, select **P2_EMP_TELECOMMUTE**.

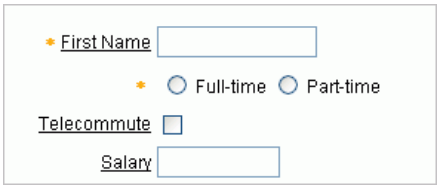
3. From the Display As list in the Name section, select **Checkbox**.
4. Under Displayed, specify the following:
   a. Sequence - Enter `136`.
   b. Begin on New Line - Select **Yes**.
5. Scroll down to List of Values.
6. To have the label precede the check box, specify the following:
   a. Named LOV - Select **Select Named LOV**.
   b. List of values definition - Enter:

      ```
      STATIC:;Y
      ```

      This List of values definition displays the check box after the label, but will not display a value associated with the check box. If the check box is checked, the value passed to the database will be Y.
7. At the top of the page, click **Apply Changes**.
8. Click the **Run Page** icon.

   Note that the check box appears for Telecommute as shown in Figure 5–20.

Figure 5–20 Telecommute Field Changed to Check Box



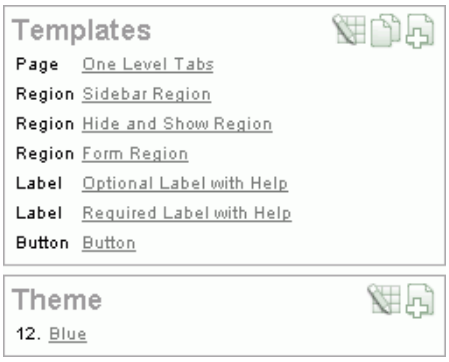
About Label Templates

You can control the look of an item label by using a label template. You can view the templates associated with the current theme on the Page Definition.

To view templates associated with the current theme:

1. Go to the Page Definition for page 2. Click **Edit Page 2** on the Developer toolbar.
2. Locate the Shared Components area and note the Theme section as shown in Figure 5–21 on page 5-21.

Figure 5–21 Templates and Theme



The current theme is Blue. In the Templates section, note that this theme includes two Label templates: Optional Label with Help and Required Label with Help.

The Required with Help label template prepends a yellow asterisk to the left of the item label. You can change the appearance of an item by selecting another template.

To change to a different label template:

1. Go to the Page Definition for page 2.
2. Under Items, select an item.
3. Scroll down to Label and make a selection from the Template list.
4. Click **Apply Changes**.
5. Run the page.

Changing Buttons

The wizard that created the form in this tutorial also created buttons. These buttons display conditionally based upon whether the page is being used to create a new record (that is, `P2_EMP_ID` equals null), or the page is being used to update an existing record. These buttons were created as HTML buttons and positioned at the top of the region.

You can also position buttons at the bottom of the region, to the left or right of the page title, above the region, below the region, or in any button position defined in the region template.

To change a button position:

1. Go to the Page Definition for page 2.
2. Under Buttons, click the **Edit All** icon in the Buttons section. The Edit All icon resembles a small grid with a pencil on top of it.
3. Make a new selection from the Position column.
4. Click **Apply Changes**.
5. Run the page.

Buttons can also have templates associated with them to refine how they look.

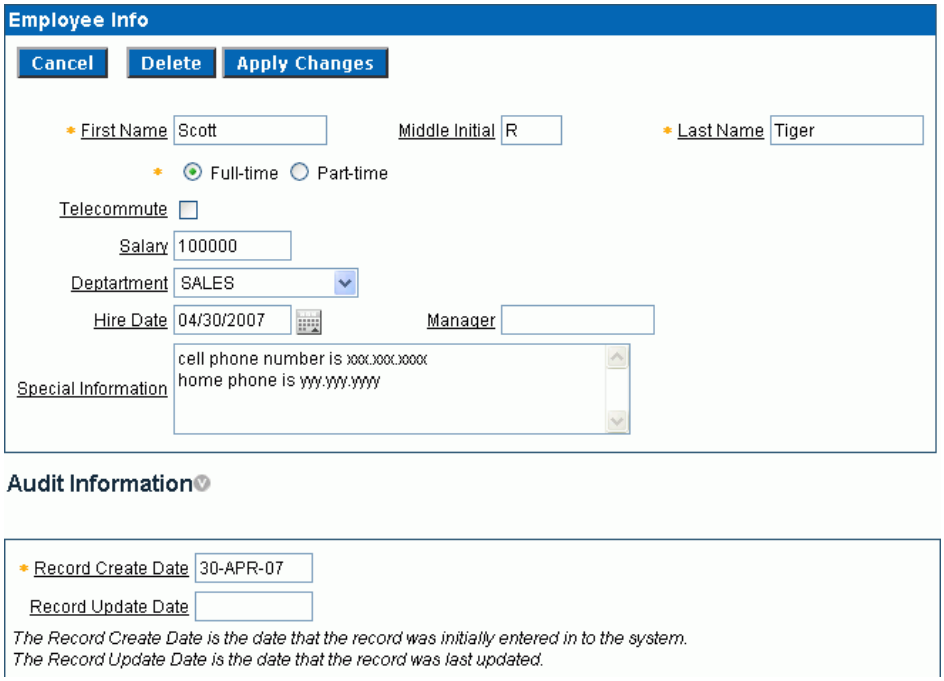
Running the Page for Update

You can run the page and provide it with an Employee ID to retrieve. Typically, this would be done with a link from a report page; but for this example, run the page and add `P2_EMP_ID:1` to the end of the URL as shown in the following example:

```
http://apex.oracle.com/pls/otn/f?p=9659:2:1284393467360777225::::P2_EMP_ID:1
```

This will pass the value `1` to the item `P2_EMP_ID`. If you run the page, note that the Delete and Apply Changes buttons now display as shown in Figure 5–22. The Create button appeared previously because the page was expecting a new record to be created. Also note that a value now appears in the Record Create Date field.

Figure 5–22 Revised Update Form



See Also: "Understanding URL Syntax" in *Oracle Application Express Application Builder User's Guide*.

Making Data Bold

One way to make the information in a region easier to read is to make the labels (or the data) more pronounced. You can accomplish this by changing the color, specifying another font, or using bold. To make a label bold, you could bold the data manually, or create a new label template. In the latter approach, you would create a new label template that would wrap the HTML tag for bold around the label and then associate that template with the items in the Audit Information region.

To make data bold manually:

1. Go to the Page Definition for page 2.
2. Under Items, select an item name.
3. Scroll down to Element.
4. In HTML Form Element Attributes, enter:

   ```
   class="fielddatabold"
   ```

   This example references a class in the Cascading Style Sheet associated with this application.
5. Click **Apply Changes**.
6. Run the page.

6

How to Work with Check Boxes

In Oracle Application Express, you can create check boxes as items, or you can create check boxes in reports. Check boxes on a form work similarly to a list of values. When you define an item to be a check box, you need to provide the check box value in the List of Values section of the Item Attributes page. You define check boxes on a report using the supplied function, `APEX_ITEM.CHECKBOX`.

This tutorial illustrates different ways in which you can create check boxes and explains how to reference and process the values of checked boxes. Before you begin, you need to import and install the *OEHR Sample Objects* application in order to access the necessary sample database objects. See "About Loading Sample Objects" on page 1-2.

This section contains the following topics:

Creating an Application

First, you need to create an application using the Create Application Wizard.

To create an application using the Create Application Wizard:

1. On the Workspace home page, click the **Application Builder** icon.

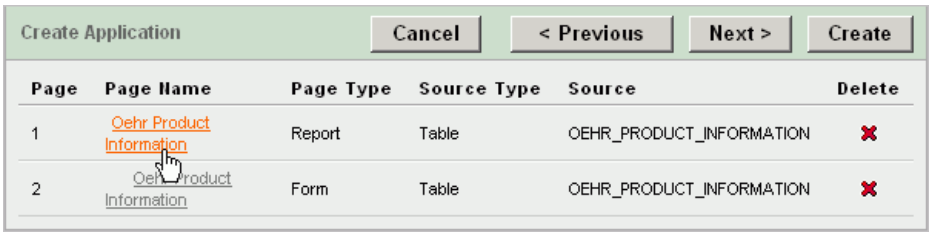
   The Application Builder home page appears.
2. Click **Create**.
3. Select **Create Application** and click **Next**.
4. For Name:
   a. Name - Enter `Check Boxes`.
   b. Application - Accept the default.
   c. Create Application - Select **From scratch**.
   d. Schema - Select the schema where you installed the OEHR sample objects.
   e. Click **Next**.

      Next, you need to add a page. For this exercise, you add a report and form.

5. To add a report and form:
   a. Select Page Type -Select **Report and Form**.
   b. Table Name - Select **OEHR_PRODUCT_INFORMATION**.
   c. Click **Add Page**.

      Two new pages appear in the list at the top of the page. Note that each page has the same page name. Next, edit the page names to make them more meaningful.
6. To edit the name of page 1:
   a. Click **OEHR_PRODUCT_INFORMATION** next to page 1 at the top of the page as shown in Figure 6–1.

Figure 6–1 Page Name in the Create Application Wizard



   b. In Page Name, replace the existing text with `Product Report`.
   c. Click **Apply Changes**.
7. To edit the name of page 2:
   a. Click **OEHR_PRODUCT_INFORMATION** next to page 2 at the top of the page as shown in Figure 6–1.
   b. In Page Name, replace the existing text with `Update Form`.
   c. Click **Apply Changes**.
8. Click **Next**.
9. For Tabs, accept the default, **One Level of Tabs,** and click **Next**.
10. For Copy Shared Components from Another Application, accept the default, **No**, and click **Next**.
11. For Attributes, accept the defaults for Authentication Scheme, Language, and User Language Preference Derived From and click **Next**.
12. For User Interface, select **Theme 2** and click **Next**.
13. Review your selections and click **Create**.

    The Application home page appears.

Run the Application

Next, review the application by running it.

To run the application:

1. Click **Run Application** as shown in Figure 6–2.

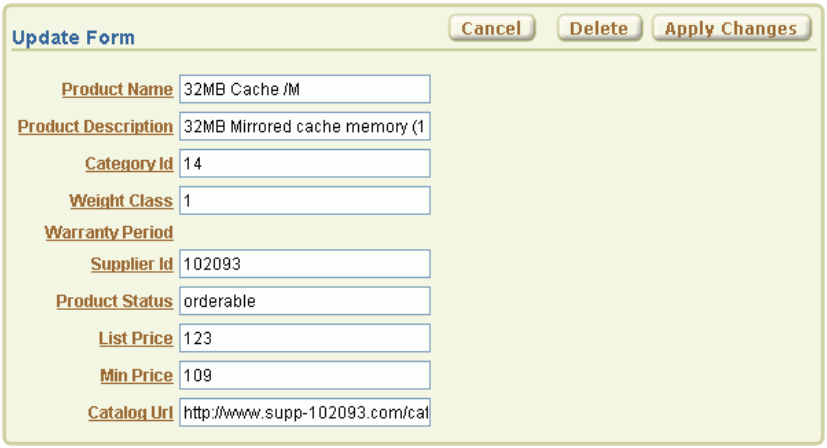
Figure 6–2 Run Application Icon

2. If prompted to enter a user name and password, enter your workspace user name and password and click **Login**. See "About Application Authentication" on page 1-6.

   The application appears. Note that the report contains ten columns displaying product information. Users can link to an update form by clicking the Edit icon in the far left column.

3. Click the **Edit** icon next to a specific product. As shown in Figure 6–3, an update form appears.

Figure 6–3 Update Form



Editing the Update Form

Page 2 of your application is an update form. In this exercise, you modify this form by hiding the Warranty Period field and creating a new check box.

Topics in this section include:

- Hide the Warranty Period Field
- Add a New Checkbox

Hide the Warranty Period Field

First, hide the Warranty Period field by changing the Display As attribute.

To hide the Warranty Period field:

1. Click **Edit Page 2** on the Developer toolbar.

   The Page Definition for page 2 appears.

2. Scroll down to the Items section.

3. Under Items, select **P2_WARRANTY_PERIOD**.

4. From Display As in the Name section, select **Hidden**.

5. Click **Apply Changes**.

Add a New Checkbox

In this exercise you create a check box that automatically sets the minimum product price to 75% of the list price.

Topics in this section include:

- Add a New Item
- Create a Process
- Edit the Item to Display as a Check Box

> **Tip:** For simplicity, this tutorial has you create a checkbox by editing item attributes. As a best practice, however, you can also create a named LOV and reference it.

> **See Also:** "Creating Lists of Values" in *Oracle Application Express Application Builder User's Guide*

Add a New Item

First, you add a new item. Initially, you create this item to display as a radio group and later change it to a check box.

To add an item that displays as a radio group:

1. On the Page Definition for page 2, scroll down to Items.
2. Under Items, click the **Create** icon as shown in Figure 6–4.

Figure 6–4 Create Icon

3. For Item Type, select **Radio** and click **Next**.
4. For Radio Group Control Type, select **Radio group** and click **Next**.
5. For Display Position and Name:
   a. Item Name - Enter `P2_SET_MIN_PRICE`.
   b. Sequence - Enter `9.5`.

      Note that this sequence positions the item below the P2_MIN_PRICE item (the Minimum Price field).
   c. Region - Select **Update Form**.
   d. Click **Next**.
6. For List of Values:
   a. Named LOV - Select **Select Named LOV**.
   b. Display Null Option - Select **No**.
   c. List of Values Query - Enter:

```
STATIC:Yes;Y,No;N
```

d. Click **Next**.

7. For Item Attributes:
   a. Label - Replace the existing text with `Set Minimum Price`.
   b. Accept the remaining defaults.
   c. Click **Next**.
8. For Source:
   a. Item Source - Select **SQL Query**.
   b. Item Source Value - Enter:

```
SELECT 'Y' FROM DUAL WHERE :P2_LIST_PRICE*0.75=:P2_MIN_PRICE
```

   c. Accept the remaining defaults and click **Create Item**.

Create a Process

Next, you create a page process that sets the minimum price at a 25% discount of the list price.

To create a page process:

1. On the Page Definition for page 2, locate the Page Processing area.
2. Under Processes, click the **Create** icon.
3. For Process Type, select **PL/SQL** and click **Next**.
4. For Process Attributes:
   a. Name - Enter `Update Min Price`.
   b. Sequence - Accept the default.
   c. Point - Select **OnSubmit - After Computations and Validataions**.
   d. Click **Next**.
5. For Process:
   a. Enter the following:

```
UPDATE oehr_product_information
SET MIN_PRICE=(:P2_LIST_PRICE*0.75)
WHERE PRODUCT_ID=:P2_PRODUCT_ID;
```

   b. Click **Next**.
6. For Messages:
   a. Success Message - Enter:

```
Product successfully updated.
```

   b. Failure Message - Enter:

```
Unable to update this product. Contact your system administrator.
```

   c. Click **Next**.
7. For Process Conditions:
   a. Condition Type - Select **Value of Item in Expression 1 = Expression 2**.

b. Expression 1 - Enter:

```
P2_SET_MIN_PRICE
```

c. Expression 2 - Enter `Y`.

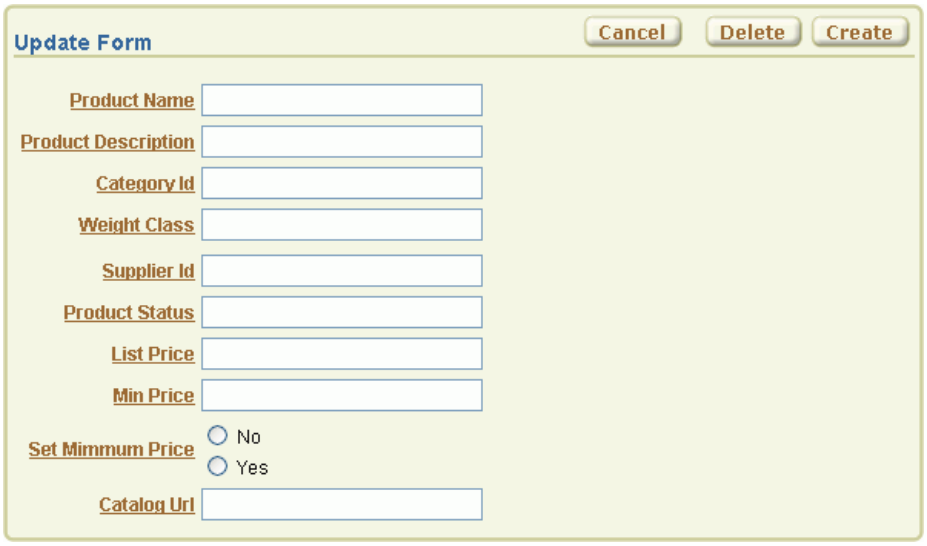
d. Click **Create Process**.

Run the Page To run the page:

1. Click the **Run Page** icon in the upper right corner.
2. If prompted to enter a user name and password, enter your workspace user name and password and click **Login**. See "About Application Authentication" on page 1-6.

   The revised form appears as shown in Figure 6–5. Note that the Warranty Period field no longer displays and a new Set Minimum Price radio group appears.

Figure 6–5 Update Form with Set Minimum Price Radio Group



Edit the Item to Display as a Check Box

Next, change the Set Minimum Price radio group (P2_SET_MIN_PRICE) to display as a check box.

To edit P2_SET_MIN_PRICE:

1. Click **Edit Page 2** on the Developer toolbar.

   The Page Definition for Page 2 appears.
2. Under Items, click **P2_SET_MIN_PRICE**.
3. From Display As, select **Checkbox**.
4. Scroll down to Label. In Label, delete the existing text, `Set Minimum Price`.
5. Scroll down to Default. In Default Value, enter `N`.
6. Under Lists of Values:

   a. Number of Columns - Enter `1`.

   b. List of values definition - Enter:

```
STATIC: <b> Set Minimum Price</b><br/> (25% Discount on List Price);Y
```

7. Click **Apply Changes** at the top of the page.

Run the Page To run the page, click the **Run Page** icon in the upper right corner. The revised form appears as shown in Figure 6–6 on page 6-7. Note the new Set Minimum Price check box.

Figure 6–6 Update Form with Set Minimum Price Check Box

Change the Report Display

You can alter how a report displays by editing report attributes. In the exercise, you change the number of columns that display on page 1 and then change the format of two columns to include a currency symbol.

To edit report attributes for page 1:

1. Click **Application** on the Developer toolbar.

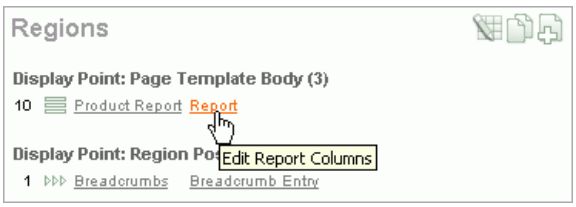
   The Application home page appears.

2. Click **1 - Product Report**.

   The Page Definition for page 1 appears.

3. Under Regions, click the **Report** link as shown in Figure 6–7 on page 6-7.

Figure 6–7 Report Link



The Report Attributes page appears. You can use this page to precisely control the report layout. First, change the number of columns that display.

4. Deselect the Show check box for the following columns:
   - Weight Class
   - Warranty Period
   - Supplier ID

   Next, edit List Price and Min Price columns to include a currency symbol.
5. Edit the List Price column:
   a. Click the **Edit** icon next to List Price.
   b. From Number / Date Format, select **$5,234.10**.
   c. Click the **Next** (>) icon at the top of the page.

      Clicking the Next icon submits your changes and then displays attributes for the next column, Min Price.
6. Edit the Min Price column:
   a. From Number / Date Format, select **$5,234.10**.

      Note that you select a format by selecting an example. However, the value that actually displays field is the Oracle number format.
   b. Click **Apply Changes**.
7. Click the **Run Page** icon in the upper right corner.

   The revised report appears. Notice the Weight Class, Warranty Period, and Supplier ID no longer appear and the List Price and Min Price columns include a currency symbol.

Create Multi Value Check Boxes to Filter Content

In the next exercise, you change the Search field (P1_REPORT_SEARCH) on the Product Report page to a multi value check box. These check boxes enable users to filter the report by product category (obsolete, orderable, planned, under development).

Topics in this section include:

- Change the Search Field to a Multi Value Check Box
- Edit the Report Region Definition
- Change the Default Check Box Behavior
- Change the Check Boxes to Display in Bold

Change the Search Field to a Multi Value Check Box

To change the search field to a check box:

1. Click **Edit Page 1** on the Developer toolbar.

   The Page Definition for page 1 appears.
2. Under Items, click **P1_REPORT_SEARCH**.
3. From Display As, select **Checkbox**.
4. Scroll down to Label. For Label, delete the existing text and replace with `Product Status`.

5. Scroll down to Source. In Source Value or Expression, enter:

```
obsolete:orderable:planned:under development
```

6. Scroll down to List of Values. Specify the following:
   a. Named LOV - Accept the default.
   b. Number of Columns - Enter `4`.
   c. List of values definition - Enter:

```
SELECT DISTINCT product_status display_value, product_status return_value
FROM oehr_product_information
ORDER BY 1
```

> **Note:** Note that to create a multi value check box, the List of Values query must return more than one row.

7. Click **Apply Changes** at the top of the page.

   The Page Definition for page 1 appears.

Edit the Report Region Definition

To edit the report region definition:

1. Under Regions, click **Product Report**.

   The Region Definition appears.

2. Scroll down to Source.

3. In Source modify the WHERE clause to read as follows:

```
...
WHERE  instr(':'||:P1_REPORT_SEARCH||':',product_status)> 0
```

4. Click **Apply Changes** at the top of the page.

   The Page Definition for page 1 appears.

5. Click **Apply Changes** at the top of the page.

   The Page Definition for page 1 appears.

Change the Default Check Box Behavior

Although the Product Status check boxes correctly filter the content on page 1, if you deselect all the check boxes, notice the report returns all products. This behavior results from the fact that if a check box has a NULL value (that is, it is deselected), then it defaults to the default value Y. The default value of Y, in turn, enables the check box.

You can alter this behavior by adding a computation that remembers the state of the check box.

To add a computation that tracks the state of the check box:

1. Under Page Processing, Computations, click the **Create** icon.

   The Create Page Computation Wizard appears.

2. For Item Location, select **Item on This Page** and click **Next**.

3. For Item, specify the following:
   a. Compute Item - Select **P1_REPORT_SEARCH**.
   b. Sequence - Accept the default.
   c. Computation Point - Select **After Submit**.
   d. Computation Type - Select **Static Assignment**.
   e. Click **Next**.
4. In Computation:
   a. Enter:

      ```
      none(bogus_value)
      ```

   b. Click **Next**.
5. For Condition:
   a. From Condition Type, select **Value of Item in Expression 1 Is NULL**.
   b. In Expression 1, enter:

      ```
      P1_REPORT_SEARCH
      ```

6. Click **Create**.

   The Page Definition for page 1 appears.
7. Click the **Run Page** icon in the upper right corner. Note that the Product Status check boxes display at the top of the page.

Change the Check Boxes to Display in Bold

Next, you edit the check box display values (or labels) so that they appear as bold text.

To edit check box display values (or labels) to appear in bold:

1. Go to the Page Definition for page 1.
2. Under Items, click **P1_REPORT_SEARCH**.
3. Scroll down to Element.
4. In Form Element Option Attributes, enter:

   ```
   class="fielddatabold"
   ```

   Form Element Option Attributes are used exclusively for check boxes and radio buttons and control the way the Application Express engine renders individual options.
5. Click **Apply Changes**.

   The Page Definition for page 1 appears.

Adding Check Boxes to Each Row in the Report

In the next exercise, you add a delete check box to each row in the Product Report. To accomplish this, you must edit the report query and make a call to the `APEX_ITEM` package.

`APEX_ITEM` is a supplied package for generating certain items dynamically. In this instance, you use `APEX_ITEM.CHECKBOX` to generate check boxes in the Product

report. When the page is submitted, the values of the check boxes are stored in global package arrays. You can reference these values using the PL/SQL variables APEX_APPLICATION.G_F01 to APEX_APPLICATION.G_F50 based on the p_idx parameter value that was passed in.

Topics in this section include:

- Call APEX_ITEM.CHECKBOX
- Add a Button to Submit Check Box Array Values
- Add a Process

> **See Also:** "APEX_ITEM" in *Oracle Application Express Application Builder User's Guide*

Call APEX_ITEM.CHECKBOX

To edit the query to call APEX_ITEM.CHECKBOX:

1. Go to the Page Definition for page 1.
2. Under Regions, click **Product Report**.
3. Scroll down to Source.
4. In Region Source, add the new line appearing in bold face to the query.

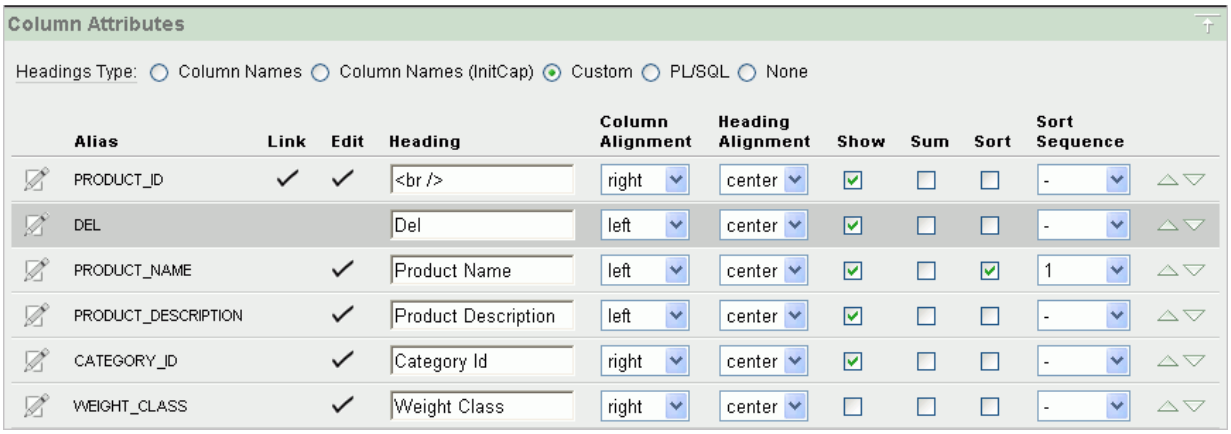
```
SELECT
"product_id",
apex_item.checkbox(1,product_id) del,
"product_name",
"product_description",
"category_id",
"weight_class",
"warranty_period",
"supplier_id",
"product_status",
"list_price",
"min_price",
"catalog_url"
FROM   "oehr_product_information"
WHERE  instr(':'||:p1_report_search||':',product_status)> 0
```

   APEX_ITEM is an Oracle Application Express supplied package that you can use to generate certain items dynamically. Note that the value passed in for p_idx in the above example is 1. You reference the check box values using the global variable APEX_APPLICATION.G_F01 later on.

   Oracle Application Express automatically adds new columns to the end of the column list. Next, you need to move the DEL column.

5. Scroll to the top of the page and select the **Report Attributes** tab.
6. Under Column Attributes, locate the Del column.
7. Click the **Up** arrow on the far right until the DEL column is directly below PRODUCT_ID. (See Figure 6–8 on page 6-12).

Figure 6–8 Report Column Attributes Page



8. Click **Apply Changes**.

   The Page Definition for page 1 appears.

Add a Button to Submit Check Box Array Values

To add a button to submit the check box array values:

1. Go to the Page Definition for page 1.
2. Under Buttons, click the **Create** icon.
3. For Button Region, select **Product Report (1)** and click **Next**.
4. For Position, select **Create a button in a region position** and click **Next**.
5. For Button Attributes:
   a. Button Name - Enter `DELETE_PRODUCTS`.
   b. Label - Enter `Delete Products`.
   c. Accept the remaining defaults and click **Next**.
6. In Button Template, accept the default selection and click **Next**.
7. For Display Properties:
   a. Position - Select **Top of Region**.
   b. Accept the remaining defaults and click **Next**.
8. For Branching, select **1 Product Report** and click **Create Button**.

Add a Process

To add a process that executes when the user clicks the Delete Products button:

1. Under Page Processing, Processes, click the **Create** icon.
2. For Process Type, select **PL/SQL** and click **Next**.
3. For Process Attributes:
   a. Name - Enter `Delete Products`.
   b. Sequence - Accept the default.
   c. For Point - Select **On Submit - After Computations and Validations**.

d. Click **Next**.

4. Enter the following PL/SQL process and then click **Next**:

```
FOR i in 1..APEX_APPLICATION.G_F01.count
LOOP
   DELETE FROM oehr_product_information
   WHERE product_id = APEX_APPLICATION.G_F01(i);
END LOOP;
```

`APEX_ITEM` is an Oracle Application Express supplied package that you can use to generate certain items dynamically. When a page is submitted, the values of each column are stored in global package arrays, which you can reference using the PL/SQL variable `APEX_APPLICATION.G_F01` to `APEX_APPLICATION.G_F50`. In this exercise, the value passed in for `product_id` is 1, so you reference the column values using the global variable `APEX_APPLICATION.G_F01`.

5. On Messages:

   a. In Success Message, enter:

   ```
   Product(s) deleted.
   ```

   b. In Failure Message, enter:

   ```
   Unable to delete product(s).
   ```

6. Click **Create Process**.

7. Run the page.

   Notice that the Delete Products button appears above the report as shown in Figure 6–9. To remove a product from the report, select the **Del** check box and then click **Delete Products**.

Figure 6–9 Product Report with Delete Products Check Box

	Del	Product Name	Product Description	Category Id	Product Status	List Price	Min Price	Catalog Url
		LCD Monitor 11/PM	Liquid Cristal Display 11 inch passive monitor. The virtually-flat, high-resolution screen delivers outstanding image quality with reduced glare.	11	under development	259	$208.00	http://www.www.supp-102067.com/cat/hw/p1726.html
		Chemicals - RCP	Cleaning Chemicals - 3500 roller clean pads	39	orderable	80	$66.00	http://www.supp-103094.com/cat/off/p1729.html
		PS 220V /UK	220V Power supply type - United Kingdom	19	orderable	89	$76.00	http://www.supp-102080.com/cat/hw/p1733.html
		Cable RS232 10/AM	10 ft RS232 cable with M/M and 9M/25M adapters	19	orderable	6	$5.00	http://www.supp-102055.com/cat/hw/p1734.html

7

How to Implement a Web Service

Web services enable applications to interact with one another over the Web in a platform-neutral, language independent environment. In a typical Web services scenario, a business application sends a request to a service at a given URL by using the HTTP protocol. The service receives the request, processes it, and returns a response. You can incorporate calls to external Web services in applications developed in Oracle Application Express.

Web services in Oracle Application Express are based on SOAP (Simple Object Access Protocol). SOAP is a World Wide Web Consortium (W3C) standard protocol for sending and receiving requests and responses across the Internet. SOAP messages can be sent back and forth between a service provider and a service user in SOAP envelopes.

Web services are called from within an Oracle Application Express application by:

- Using the Universal Description, Discovery, and Integration (UDDI) registry
- Manually providing the WSDL URL

This tutorial illustrates the later method.

Topics in this section include:

Note: The SOAP 1.1 specification is a W3C note. (The W3C XML Protocol Working Group has been formed to create a standard that will supersede SOAP.)

For information about Simple Object Access Protocol (SOAP) 1.1 see:

`http://www.w3.org/TR/SOAP/`

See Also: "Implementing Web Services" in *Oracle Application Express Application Builder User's Guide*

About Creating Web Service References

To utilize Web services in Oracle Application Express, you create a Web service reference using a wizard. When you create the Web reference, you can follow one of these methods:

- You supply the URL to a WSDL document. The wizard then analyzes the WSDL and collects all the necessary information to create a valid SOAP message.

  The wizard provides a step where you can locate a WSDL using the Universal Description, Discovery, and Integration (UDDI) registry. A UDDI registry is a directory where businesses register their Web services. You search for the WSDL by entering either a service or business name.

- You supply the relevant information on how to interact with the Web reference, including the SOAP request envelope, and create the Web reference manually.

This tutorial describes the second method, creating a Web service reference manually.

Creating a New Application

First, create a new application.

To create an application:

1. On the Workspace home page, click the **Application Builder** icon.
2. On the Application Builder home page, click **Create**.
3. For Method, select **Create Application**, and click **Next**.
4. For Name:
   a. For Name - Enter `Web Services`.
   b. Accept the remaining defaults and click **Next**.
5. Add a blank page:
   a. Under Select Page Type, accept the default, **Blank**.
   b. In Page Name, enter `Web Services` and then click **Add Page**.

      The new page appears in the list at the top of the page.
   c. Click **Next**.
6. For Tabs, accept the default, **One Level of Tabs**, and click **Next**.
7. For Shared Components, accept the default, **No**, and click **Next**.
8. For Attributes, accept the default for Authentication Scheme, Language, and User Language Preference Derived From and click **Next**.
9. For User Interface, select **Theme 2** and click **Next**.
10. Review your selections and click **Create**.

The Application home page appears.

Specifying an Application Proxy Server Address

If your environment requires a proxy server to access the Internet, you must specify a proxy server address on the Application Attributes page before you can create a Web service reference.

To specify a proxy address:

1. On the Application home page, click **Shared Components**.
2. Under Application, click **Definition**.
3. Under Name, enter the proxy server in the Proxy Server field.
4. Click **Apply Changes**.

   The Application home page appears.

Creating a Web Service Reference from a WSDL

In this exercise, you create a Web service by supplying the location of a WSDL document to a Web service. You then create a form and report for displaying movie theaters and locations.

> **Note:** The following exercise is dependent upon the availability of the specified Web service ultimately invoked. If the Web service is unavailable, you may experience difficulties completing this exercise.

To create a new Web reference by supplying the WSDL location:

1. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
2. Under Logic, select **Web Service References**.
3. Click **Create**
4. When prompted whether to search a UDDI registry to find a WSDL, select **No** and click **Next**.
   a. In the WSDL Location field enter:

      ```
      http://www.ignyte.com/webservices/ignyte.whatsshowing.webservice/moviefunctions.asmx?wsdl
      ```

   b. Click **Next**.

      A summary page appears describing the selected Web service.
5. Click **Create Reference**.

   The Create Web Service Reference page appears. The Web service reference for MovieInformation is added to the Web Service References Repository.

Create a Form and Report

Next, you need to create a page that contains a form and report to use with your Web Service Reference.

To create a form and report after creating a Web Service Reference:

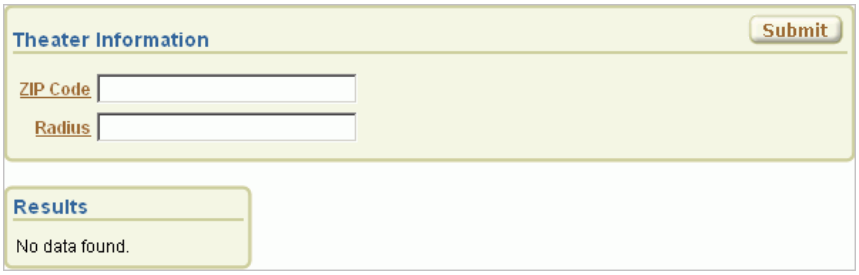
1. On the Create Web Service Reference success page, select **Create Form and Report on Web Service**.
2. For Choose Service and Operation:
   a. Web Service Reference - Select **MovieInformation**.
   b. Operation - Select **GetTheatersAndMovies**.

c. Click **Next**.

3. For Page and Region Attributes:
   a. Form Region Title - Change to `Theater Information`.
   b. Accept the other defaults and click **Next**.
4. For Input Items:
   a. For P2_ZIPCODE and P2_RADIUS, accept the default, **Yes**, in the Create column.
   b. For P2_ZIPCODE, change the Item Label default to `ZIP Code`.
   c. Click **Next**.
5. For Web Service Results:
   a. Temporary Result Set Name (Collection) - Accept the default.
   b. Result Tree to Report On - Select **Theater (tns:Theater)**.
   c. Click **Next**.
6. For Result Parameters, select all the parameters and click **Finish**.
7. Click **Run Page**.
8. If prompted to log in, enter the user name and password for your workspace and click **Login**.

   A form and report resembling Figure 7–1 on page 7-4 appear. Notice that the Theater Information Form at the top of the page contains a data entry field and a submit button, but the Results Report does not contain any data.

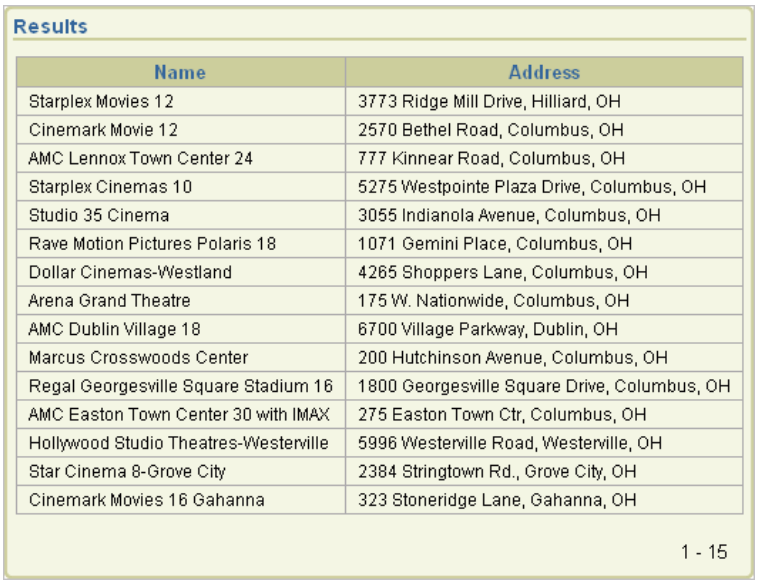
Figure 7–1 Theater Information Form and Report without Data



9. To test the form, enter `43221` in the ZIP Code field and `5` in the Radius field. Then click **Submit**.

   The report at the bottom of the page should resemble Figure 7–2. The report lists the names and addresses of movie theaters matching the entered ZIP code and radius.

Figure 7–2 Theater Information Report Showing Resulting Data



Results

Name	Address
Starplex Movies 12	3773 Ridge Mill Drive, Hilliard, OH
Cinemark Movie 12	2570 Bethel Road, Columbus, OH
AMC Lennox Town Center 24	777 Kinnear Road, Columbus, OH
Starplex Cinemas 10	5275 Westpointe Plaza Drive, Columbus, OH
Studio 35 Cinema	3055 Indianola Avenue, Columbus, OH
Rave Motion Pictures Polaris 18	1071 Gemini Place, Columbus, OH
Dollar Cinemas-Westland	4265 Shoppers Lane, Columbus, OH
Arena Grand Theatre	175 W. Nationwide, Columbus, OH
AMC Dublin Village 18	6700 Village Parkway, Dublin, OH
Marcus Crosswoods Center	200 Hutchinson Avenue, Columbus, OH
Regal Georgesville Square Stadium 16	1800 Georgesville Square Drive, Columbus, OH
AMC Easton Town Center 30 with IMAX	275 Easton Town Ctr, Columbus, OH
Hollywood Studio Theatres-Westerville	5996 Westerville Road, Westerville, OH
Star Cinema 8-Grove City	2384 Stringtown Rd., Grove City, OH
Cinemark Movies 16 Gahanna	323 Stoneridge Lane, Gahanna, OH

1 - 15

Creating a Web Service Reference Manually

In this exercise, you create a Web reference by supplying information about the Web service and using the manual facility. Manual Web references are created by visually inspecting the WSDL document as well as using a tool to determine the SOAP envelope for the Web service request.

Topics in this section include:

- Create a Web Service Reference Manually
- Test the Web Service
- Create a Page to Call the Manual Web Service
- Create a Submit Button
- Create Items for ZIP Code and Radius
- Create a Process to Call the Manually Created Web Reference
- Create a Report on the Web Service Result

Create a Web Service Reference Manually

To create a Web reference manually, you will copy code from the WSDL for a service called MovieInformation.

Please note the example settings provided in the following steps are based on the MovieInformation service at the time this document was released.

To create a manual Web reference:

1. On the Application home page, click **Shared Components**.
2. Under Logic, click **Web Service References**.
3. Click **Create**.
4. For Search UDDI, select **No** and click **Next**.
5. From the Tasks list on the right, click the **Create Web Reference Manually** link.

The Create/Edit Web Service page appears.

6. In the Name field, enter `Movie Info`.
7. Locate the endpoint of the MovieInformation service:
   a. Open the WSDL by going to:

      `http://www.ignyte.com/webservices/ignyte.whatsshowing.webservice/moviefunctions.asmx?wsdl`

   b. In the WSDL, find the `location` attribute of the `soap:address` element, which is a child of the `port` element. You can search for the following term within the code: soap:address location.

      At the time of this release, it was this attribute:

      ```
      http://www.ignyte.com/webservices/ignyte.whatsshowing.webservice/moviefunctions.asmx
      ```

8. In the URL field on the Create/Edit Web Service page, enter the endpoint of the MovieInformation service you located. For example:
   `http://www.ignyte.com/webservices/ignyte.whatsshowing.webservice/moviefunctions.asmx`
9. Locate the SOAP action for the `GetTheatersAndMovies` operation:
   a. If necessary, open the WSDL again. See Step 7a.
   b. In the WSDL, find the `soapAction` attribute of the `soap:operation` element, which is a child of the operation element that has a name attribute of `GetTheatersAndMovies`. You can search for the following term within the code: `soap:operation soapAction`.

      At the time of this release, it was this attribute:

      ```
      http://www.ignyte.com/whatsshowing/GetTheatersAndMovies
      ```

10. In the Action field on the Create/Edit Web Service page, enter the SOAP action you located. For example:
    `http://www.ignyte.com/whatsshowing/GetTheatersAndMovies`
11. In the SOAP Envelope field on the Create/Edit Web Reference page, enter the xml code representing the SOAP Request message. For example:

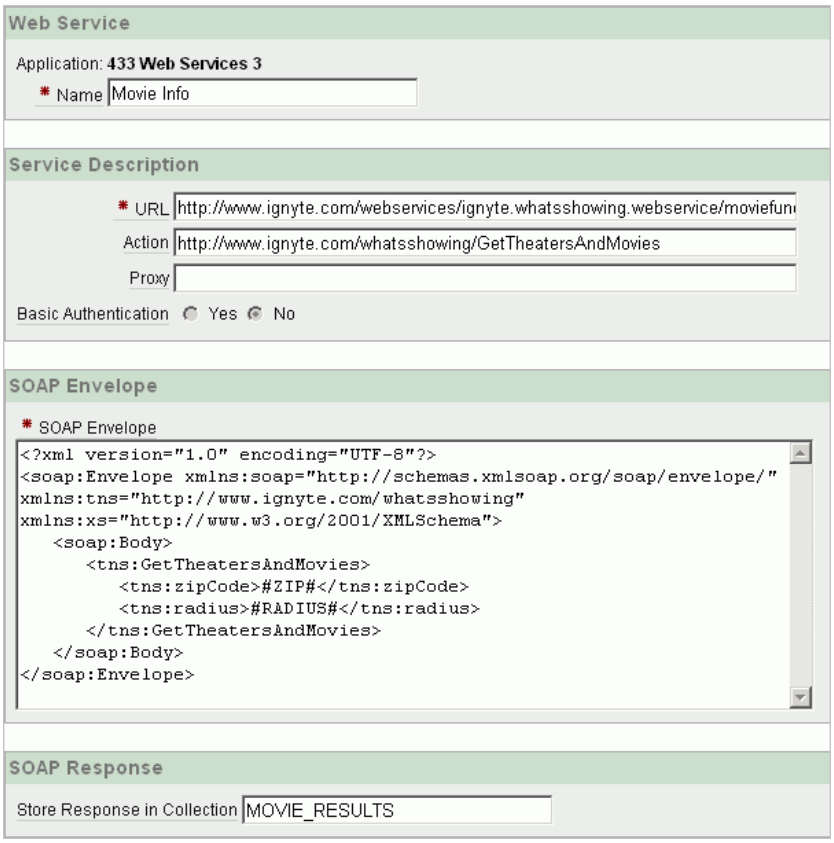
    ```
    <?xml version="1.0" encoding="UTF-8"?>
    <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:tns="http://www.ignyte.com/whatsshowing"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
       <soap:Body>
          <tns:GetTheatersAndMovies>
             <tns:zipCode>#ZIP#</tns:zipCode>
             <tns:radius>#RADIUS#</tns:radius>
          </tns:GetTheatersAndMovies>
       </soap:Body>
    </soap:Envelope>
    ```

    You can use a SOAP message generating tool, such as SOAPUI located at `http://soapui.org`, to construct a valid SOAP Request for a given Web service.

12. In the Store Response in Collection field, enter `MOVIE_RESULTS`. This is where the response from the Web service will be stored.

The Create/Edit Web Service page should resemble Figure 7–3.

Figure 7–3 Create/Edit Web Service Page



13. Click **Create**.

    The Web Service References page appears, showing Movie Info in the list.

Test the Web Service

To test the Web service:

1. On the Web Service References page, click the **Test** icon next to the Movie Info Web reference.

   The Web Services Testing page appears.

   Note View must be set to Details, otherwise the **Test** icon is not displayed.

2. In the SOAP Envelope field, replace #ZIP# with `43221` and #RADIUS# with `5`.
3. Click **Test**.
4. Review the Result field and note the following about the response:
   - The base node in the return envelope is called:

     `GetTheatersAndMoviesResponse`

   - The namespace for the message is:

     `http://www.ignyte.com/whatsshowing`

- The XPath to the interesting parts of the response under the result element is:

  ```
  /GetTheatersAndMoviesResponse/GetTheatersAndMoviesResult/Theater/Movies/Movie
  ```

- The interesting elements in the results are called:

  ```
  Name
  Rating
  RunningTime
  ShowTimes
  ```

Create a Page to Call the Manual Web Service

Next, you want to create a page to call the manual Web Service.

To create a page to call the manual Web service:

1. Click the **Application** breadcrumb link.
2. On the Application home page, click **Create Page**.
3. For Page, select **Blank Page** and click **Next**.
4. Accept the default for the Page Number and click **Next**.
5. In Name, enter `Find Movies` and click **Next**.
6. For Tabs, accept the default, **No**, and click **Next**.
7. Click **Finish**.
8. On the Success page, click **Edit Page**.
9. On the Page Definition, locate the Regions section.
10. Click the **Create** icon.
11. For Region, select **HTML** and click **Next**.
12. Select **HTML** as the HTML region container and click **Next**.
13. In the Title field, enter `Movie Information` and click **Next**.
14. Click **Create Region**.

Create a Submit Button

Next, you want to add a Submit button to the region to initiate a search from the page.

To create a Submit button:

1. On the Page Definition, click the **Create** icon in the Buttons section.
2. For Button Region, accept the default, **Movie Information**, and click **Next**.
3. For Button Position, accept the default, **Create a button in a region position**, and click **Next**.
4. For Button Attributes, enter `SUBMIT` in the Button Name and click **Next**.
5. For Button Template, accept the default, **Button**, and click **Next**.
6. For Display Properties, select **Region Template Position #CREATE#** from the Position list and click **Next**.

7. In the Branch to Page field, select **Find Movies** from the list. The page number appears in the field.
8. Click **Create Button**.

Create Items for ZIP Code and Radius

Next, you want to create two items where users can enter a search term.

To create the ZIP Code item:

1. On the Find Movies Page Definition, click the **Create** icon in the Items section.
2. For Item Type, select **Text** and click **Next**.
3. For Text Control Display Type, accept the default, **Text Field**, and click **Next**.
4. For Display Position and Name, specify the following:
   a. Item Name - Enter `ZIP`.

      The Movie Info Web Service Reference defines the zip code sent in the SOAP Request as #ZIP#. Therefore, this Item Name must be ZIP in order for it's value to be substituted into the SOAP Request sent to the Web Service.

   b. Region - Accept the default, **Movie Information**.
   c. Click **Next**.
5. In the Label field, replace the existing text with `ZIP Code` and click **Next**.
6. Click **Create Item**.

To create the Radius item:

1. On the Find Movies Page Definition, click the **Create** icon in the Items section.
2. For Item Type, select **Text** and click **Next**.
3. For Text Control Display Type, accept the default, **Text Field**, and click **Next**.
4. For Display Position and Name, specify the following:
   a. Item Name - Enter `RADIUS`.

      The Movie Info Web Service Reference defines the radius sent in the SOAP Request as #RADIUS#. Therefore, this Item Name must be RADIUS in order for it's value to be substituted into the SOAP Request sent to the Web Service.

   b. Region - Accept the default, **Movie Information**.
   c. Click **Next**.
5. In the Label field, enter `Radius` and click **Next**.
6. Click **Create Item**.

Create a Process to Call the Manually Created Web Reference

Next, you want to create a process that calls the manually created Web reference.

To create a process to call the manually created Web reference:

1. On the Find Movies Page Definition, click the **Create** icon in the Processes section.
2. For Process Type, select **Web Services** and click **Next**.
3. In the Name field, enter `Call Movie Info` and click **Next**.

4. From the Web Service Reference list, select **Movie Info** and click **Next**.
5. In the Success Message area, enter `Called Movie Info`.
6. In the Failure Message area, enter `Error calling Movie Info` and click **Next**.
7. From the When Button Pressed list, select **SUBMIT** and click **Create Process**.

Create a Report on the Web Service Result

Next, you want to add a report that displays the results of the called Web service.

To create a report on the Web service result:

1. On the Find Movies Page Definition, click the **Create** icon in the Regions section.
2. Select **Report** and click **Next**.
3. For Region, select **Report on collection containing Web service result** and click **Next**.
4. In the Title field, enter `Search Results` and click **Next**.
5. For Web Reference Type, select **Manually Created** and click **Next**.
6. For Web Reference Information, specify the following:
   a. Web Service Reference - Select **Movie Info** from the list.
   b. SOAP Style - Select **Document**.
   c. Message Format - Select **Literal**.

      Note that these two attributes can be determined by manually inspecting the WSDL document for the service.

   d. Result Node Path - Enter:

      ```
      /GetTheatersAndMoviesResponse/GetTheatersAndMoviesResult/Theater/Movies/Movie
      ```

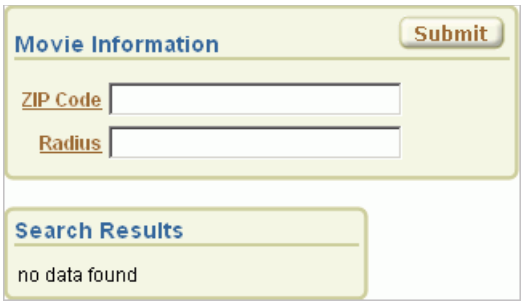
   e. Message Namespace - Enter:

      ```
      http://www.ignyte.com/whatsshowing
      ```

      Note that you reviewed both the Result Node Path and Message Namespace when testing the service.

   f. Click **Next**.
7. In the first four Parameter Names, enter `Name`, `Rating`, `RunningTime`, and `ShowTimes`, and click **Create SQL Report**.
8. To test the page:
   a. Click **Run Page**. If prompted to log in, enter your workspace user name and password. The Movie Information form appears, as shown in Figure 7–4.

Figure 7–4 Movie Information Form with No Data



b. In the ZIP Code and Radius fields, enter information and click **Submit**. The results appear in the Search Results area.

8

How to Create a Stacked Bar Chart

A stacked bar chart displays the results of multiple queries stacked on top of one another, either vertically or horizontally. Using a stacked bar chart is an effective way to present the absolute values of data points represented by the segments of each bar, as well as the total value represented by data points from each series stacked in a bar.

Although Application Builder includes built-in wizards for generating HTML, Scalable Vector Graphics (SVG), and Flash charts, only SVG and Flash charts support stacked bar charts.

This tutorial describes how to create a Flash stacked bar chart. Before you begin, you need to import and install the *OEHR Sample Objects* application in order to access the necessary sample database objects. See "About Loading Sample Objects" on page 1-2.

This section contains the following topics:

- About the Syntax for Creating Chart Queries
- Creating an Application
- Creating a New Page
- Adding Additional Series
- Updating the Sample Data
- Viewing the Chart
- Editing Chart Attributes

> **See Also:** "Creating Charts" in *Oracle Application Express Application Builder User's Guide*

About the Syntax for Creating Chart Queries

The syntax for the select statement of a chart is:

```
SELECT link, label, value
FROM   ...
```

Where:

- `link` is a URL. This URL will be called if the user clicks on the that point on the resulting chart.
- `label` is the text that displays in the bar.
- `value` is the numeric column that defines the bar size.

You must have all three items in your select statement. In the next example, the link is defined as null because there is no appropriate page to link to.

For example:

```
SELECT null link,
       last_name label,
       salary value
FROM   employees
WHERE  DEPARTMENT_ID = :P101_DEPARTMENT_ID
```

See Also: "Creating Charts" in *Oracle Application Express Application Builder User's Guide*

Creating an Application

First, you create an application using the Create Application Wizard.

To create an application using the Create Application Wizard:

1. On the Workspace home page, click the **Application Builder** icon.

   The Application Builder home page appears.
2. Click **Create**.
3. Select **Create Application** and click **Next**.
4. For Name:
   a. Name - Enter `Bar Chart`.
   b. Application - Accept the default.
   c. Create Application - Select **From scratch**.
   d. Schema - Select the schema where you installed the OEHR sample objects.
   e. Click **Next**.

      Next, you need to add a page. You have the option of adding a blank page, a report, a form, a tabular form, or a report and form. For this exercise, you add a blank page.
5. Add a blank page:
   a. Under Select Page Type, select **Blank** and click **Add Page**.

      The new page appears in the list at the top of the page.
   b. Click **Next**.
6. For Tabs, accept the default, **One Level of Tabs,** and click **Next**.
7. For Copy Shared Components from Another Application, accept the default, **No**, and click **Next**.
8. For Attributes, accept the defaults for Authentication Scheme, Language, and User Language Preference Derived From and click **Next**.
9. For User Interface, select **Theme 2** and click **Next**.
10. Review your selections and click **Create**.

    The Application home page appears.

Creating a New Page

To create your chart, you can either add a region to an existing page and define it as a stacked bar chart, or you can create a new page. In this exercise, you create a new page within the Bar Chart application you just created.

The chart will display the sum for sales by product category. It will contain sales for the twelve months prior to the current month. In the following exercise, you use a wizard to create the chart and the first query. Then, you add additional queries (or series) for other product categories to make it stacked.

See Also: "About the Syntax for Creating Chart Queries" on page 8-1

To create a new page:

1. On the Application home page, click **Create Page**.
2. For page, select **Chart** and click **Next**.
3. Select **Flash Chart** and click **Next**.
4. For Page Attributes:
   a. For Page Number, enter `2`.
   b. For Page Name, enter `Revenue by Category`.
   c. For Region Template, accept the default.
   d. For Region Name, enter `Revenue by Category`.
   e. For Breadcrumb, accept the default.
   f. Click **Next**.
5. For Tab Options, accept the default, **Do not use Tabs**, and then click **Next**.

   The Chart Preview appears. Use Chart Preview to configure chart attributes. Click **Update** to refresh the preview image.
6. On Chart Preview, specify the following:
   a. Chart Type - Select **Stacked 3D Column**.
   b. Show Legend - Select **Right**.
   c. Click **Update**.

      Notice the changes to the preview.
   d. Click **Next**.
7. For Query:
   a. Enter the following query:

```
SELECT NULL link,
       sales_month value,
       revenue "Hardware"
   FROM (
SELECT TO_CHAR(o.order_date,'Mon YY') sales_month,
       SUM(oi.quantity * oi.unit_price) revenue,
       TO_DATE(to_char(o.order_date,'Mon YY'),'Mon YY') sales_month_order
  FROM OEHR_PRODUCT_INFORMATION p,
       OEHR_ORDER_ITEMS oi,
       OEHR_ORDERS o,
       OEHR_CATEGORIES_TAB ct
```

```
 WHERE o.order_date <= (trunc(sysdate,'MON')-1)
   AND o.order_date > (trunc(sysdate-365,'MON'))
   AND o.order_id = oi.order_id
   AND oi.product_id = p.product_id
   AND p.category_id = ct.category_id
   AND ct.category_name like '%hardware%'
GROUP BY TO_CHAR(o.order_date,'Mon YY')
ORDER BY sales_month_order
)
```

The value label (in this instance, `Hardware`) is displayed in the legend of stacked charts.

Tip: You can also create a chart query interactively by clicking the **Build Query** button.

b. For When No Data Found Message, enter:

```
No orders found in the past 12 months.
```

c. Click **Next**.

8. Review your selections and click **Finish**.

   The Success page appears.

Adding Additional Series

Now that you have created a page with a region defining the query, you need to add additional series. In the following exercise, you add a series for the categories software and office equipment.

To add additional series:

1. On the Success Page, click **Edit Page**.

   The Page Definition for page 2 appears.

2. Under Regions, click **Flash Chart** next to Revenue by Category.

   The Flash Chart page appears with the Chart Attributes tab selected. Scroll down to Chart Series. Note that only one series appears.

3. To change the name the existing series:

   a. Click the **Edit** icon.

   b. In Series Name, enter `Hardware`.

   c. Click **Apply Changes**.

4. Add a chart series for software:

   a. Scroll down to Chart Series and then click **Add Series**.

   b. For Series Name, enter `Software`.

   c. Scroll down to Series Query.

   d. In SQL, enter:

```
SELECT NULL link,
       sales_month value,
       revenue "Software"
  FROM (
```

```
SELECT TO_CHAR(o.order_date,'Mon YY') sales_month,
       SUM(oi.quantity * oi.unit_price) revenue,
       TO_DATE(to_char(o.order_date,'Mon YY'),'Mon YY') sales_month_order
  FROM OEHR_PRODUCT_INFORMATION p,
       OEHR_ORDER_ITEMS oi,
       OEHR_ORDERS o,
       OEHR_CATEGORIES_TAB ct
 WHERE o.order_date <= (trunc(sysdate,'MON')-1)
   AND o.order_date > (trunc(sysdate-365,'MON'))
   AND o.order_id = oi.order_id
   AND oi.product_id = p.product_id
   AND p.category_id = ct.category_id
   AND ct.category_name like '%software%'
GROUP BY TO_CHAR(o.order_date,'Mon YY')
ORDER BY sales_month_order
)
```

The value label (in this instance, `Software`) is displayed in the legend of stacked charts. Note that this SQL matches the previous series. The only difference is the category in the `WHERE` clause.

e. For When No Data Found Message, enter:

```
No orders found in the past 12 months.
```

f. At the top of the page, click **Apply Changes**.

5. Add a chart series for office equipment:

a. Under Chart Series, click **Add Series**.

b. For Series Name, enter `Office Equipment`.

c. Scroll down to Series Query.

d. In SQL, enter:

```
SELECT NULL link,
       sales_month value,
       revenue "Office Equipment"
   FROM (
SELECT TO_CHAR(o.order_date,'Mon YY') sales_month,
       SUM(oi.quantity * oi.unit_price) revenue,
       TO_DATE(to_char(o.order_date,'Mon YY'),'Mon YY') sales_month_order
  FROM OEHR_PRODUCT_INFORMATION p,
       OEHR_ORDER_ITEMS oi,
       OEHR_ORDERS o,
       OEHR_CATEGORIES_TAB ct
 WHERE o.order_date <= (trunc(sysdate,'MON')-1)
   AND o.order_date > (trunc(sysdate-365,'MON'))
   AND o.order_id = oi.order_id
   AND oi.product_id = p.product_id
   AND p.category_id = ct.category_id
   AND ct.category_name like '%office%'
GROUP BY TO_CHAR(o.order_date,'Mon YY')
ORDER BY sales_month_order
)
```

The value label (in this instance, `Office Equipment`) is displayed in the legend of stacked charts.

e. For When No Data Found Message, enter:

```
No orders found in the past 12 months.
```

f. Scroll up to the top of the page and click **Apply Changes**.

Updating the Sample Data

The sample data that installed with the *OEHR Sample Objects* application is not current. To make the data current, you need to update the dates in the sample data. You will accomplish this by running an update statement in SQL Commands

> **See Also:** "Using SQL Commands" in *Oracle Application Express Application Builder User's Guide*

To update the dates in the seed data:

1. Return to the Workspace home page. Click the **Home** breadcrumb link at the top of the page.
2. On the Workspace home page, click **SQL Workshop** and then **SQL Commands**.

   The SQL Commands page appears.
3. Enter the following in the SQL editor pane:

```
DECLARE
   l_date_offset  number;
BEGIN

FOR c1 IN (SELECT TRUNC(max(order_date)) max_date
             FROM oehr_orders)
LOOP
   l_date_offset := round(sysdate - c1.max_date);
END LOOP;
UPDATE oehr_orders
   set order_date = order_date + l_date_offset;
COMMIT;
END;
/
```

4. Click **Run** (Ctrl+Enter) to execute the command.

Viewing the Chart

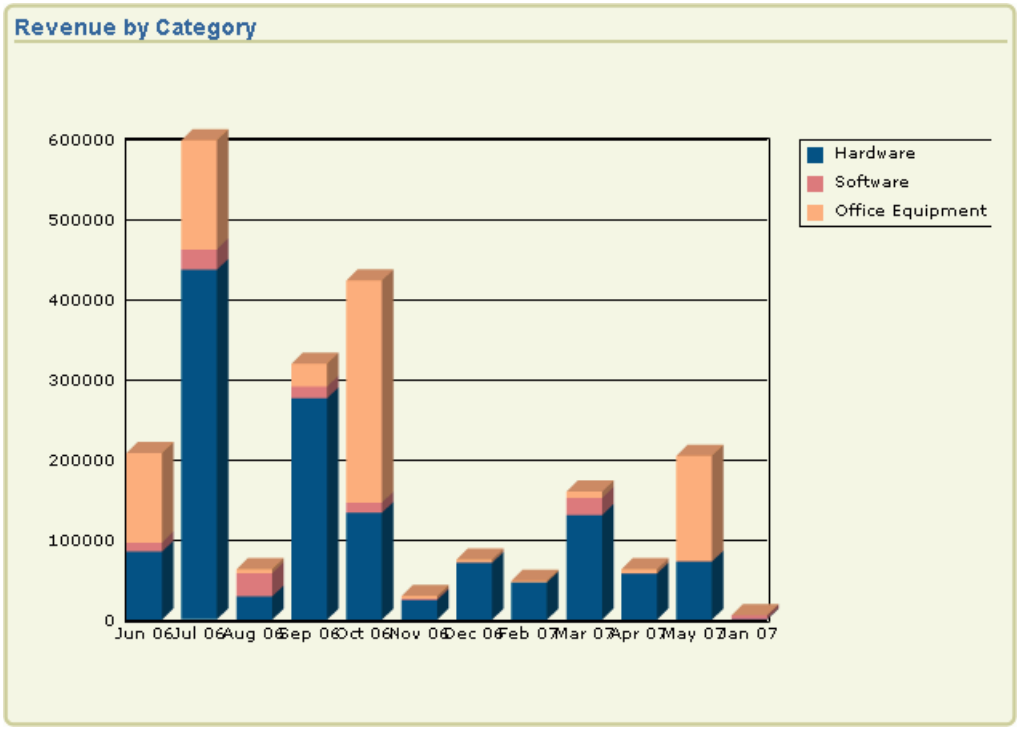
Now that the chart is complete, you can view it.

To run the chart:

1. Return to page 2, Revenue by Category:
   a. Click the **Home** breadcrumb link at the top of the page.
   b. Click **Application Builder** and then click your **Bar Chart** application.
   c. Click **2 - Revenue by Category**.
2. Click the **Run Page** icon in the upper right corner of the page.
3. If prompted for a user name and password, enter your workspace user name and password and click **Login**. See "About Application Authentication" on page 1-6.

   Your bar chart should resemble Figure 8–1.

Figure 8–1 Revenue by Category Stacked Bar Chart



The chart displays the revenue for each product category by month. A legend that defines the color associated with each product appears at the top of the page. Note that the text in X Axis is spaced too closely together. In the next section, you edit the chart attributes to correct this issue.

Editing Chart Attributes

In this exercise, you change the appearance of your chart by editing chart attributes.

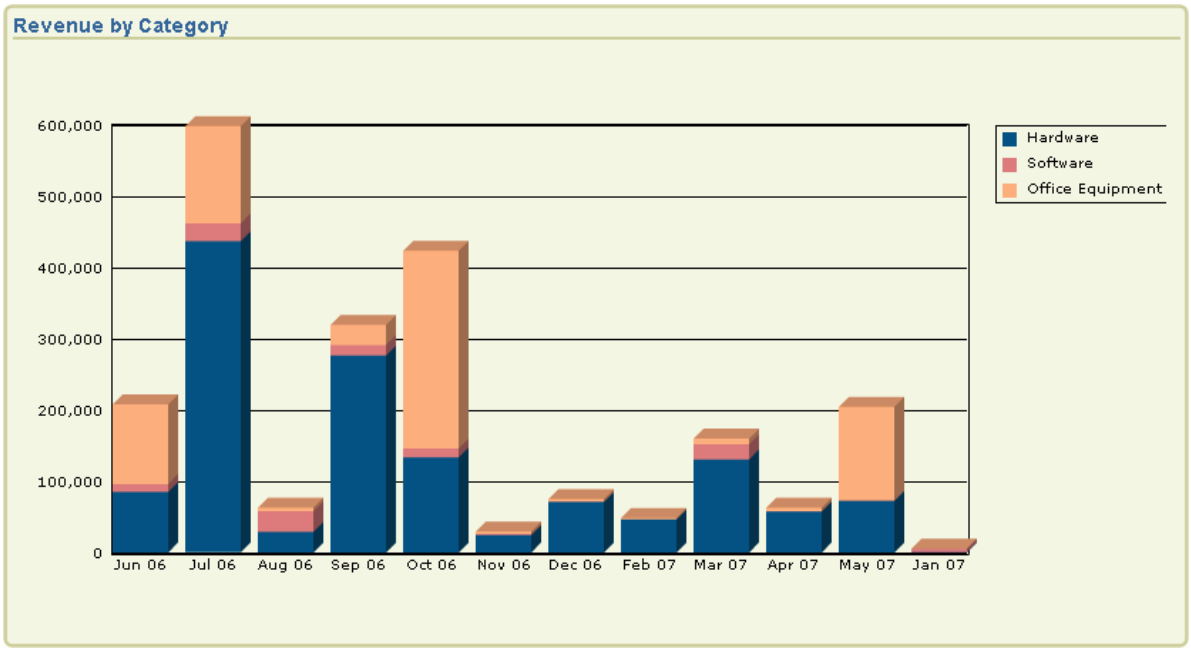
To edit chart attributes:

1. Click **Edit Page 2** on the Developer toolbar.
2. Under Regions, click **Flash Chart**.

   The Chart Attributes page appears.
3. Under Chart Settings, edit the chart width. In Chart Width, enter **800**.
4. Scroll down to Display Settings. From Animation, select **Dissolve**.
5. Scroll down to Axes Settings. From Show Group Separator, select **Yes**.
6. Scroll down to Font Settings. For X Axis Title, select the Font Size **10**.
7. Click **Apply Changes** to a the top of the page.
8. Click the **Run Page** icon in the upper right corner of the page.

   You chart should resemble Figure 8–2.

Figure 8–2 Revised Stacked Bar Chart



Note that the chart displays on-screen gradually using a dissolve and the X Axis displays correctly.

9

How to Upload and Download Files in an Application

Oracle Application Express applications support the ability to upload and download files stored in the database. This tutorial illustrates how to create a form and report with links for file upload and download, how to create and populate a table to store additional attributes about the documents, and finally how to create the mechanism to download the document in your custom table.

This section contains the following topics:

- Creating an Application
- Creating an Upload Form
- Creating a Report with Download Links
- Storing Additional Attributes About the Document
- Storing the Document in a Custom Table
- Downloading Documents from the Custom Table
- Security Issues to Consider

> **Tip:** If you have a table with a column of type `BLOB`, Oracle Application Express release 3.1 includes a built-in mechanism to help the user load the selected file. To learn more, see "About BLOB Support in Forms" in *Oracle Application Express Application Builder User's Guide*.

Creating an Application

First, create a new application using the Create Application Wizard with the assumption you will include an upload form on page 1.

To create an application using the Create Application Wizard:

1. On the Workspace home page, click the **Application Builder** icon.

   The Application Builder home page appears.
2. Click **Create**.
3. Select **Create Application** and then click **Next**.
4. For Name, specify the following:
   a. For Name, enter `Download App`.
   b. Accept the remaining defaults and click **Next**.

5. Add a blank page:
   a. Under Select Page Type, select **Blank** and click Add Page.

      The new page appears in the Create Application list at the top of the page.

   b. Click **Next**.
6. For Tabs, accept the default, **One Level of Tabs,** and click **Next**.
7. For Copy Shared Components from Another Application, accept the default, **No**, and click **Next**.
8. For Attributes, accept the defaults for Authentication Scheme, Language, and User Language Preference Derived From and click **Next**.
9. For User Interface, select **Theme 2** and then click **Next**.
10. Review your selections and click **Create**.

    The Application home page appears.

Creating an Upload Form

Once you create an application, the next step is to create a form to upload documents. In the following exercise, you create a form in an HTML region that contains a file upload item and a button. The button submits the page and returns the user to the same page.

Topics in this section include:

- Create an HTML Region
- Create an Upload Item
- Create a Button

Create an HTML Region

First, you need to create a container to hold the form. In Application Builder, this container is called a region.

To create an HTML region:

1. Click the **Page 1** icon.

   The Page Definition appears.
2. Under Regions, click the **Create** icon as shown in Figure 9–1 on page 9-2.

Figure 9–1 Create Icon

3. For Region:
   a. Identify the type of region to add to this page - Accept the default, **HTML**, and click **Next**.
   b. Select the type of HTML region container you wish to create - Accept the default, **HTML**, and click **Next**.

4. For Display Attributes:
   a. Title - Enter `Submit File`.
   b. Accept the remaining defaults and click **Next**.
5. Accept the remaining defaults and click **Create Region**.

   The Page Definition appears.

Create an Upload Item

Next, you need to create a text field or item. In Application Builder, an item is part of an HTML form. An item can be a text field, text area, password, select list, check box, and so on. In this exercise, you will create a File Browse item. When you create a File Browse item, files you upload are stored in a table named `wwv_flow_file_objects$`.

To create a file upload item:

1. Under Items on the Page Definition for page 1, click the **Create** icon.
2. For Item Type, select **File Browse** and then click **Next**.
3. For Display Position and Name:
   a. Item Name - Enter `P1_FILE_NAME`.
   b. For Sequence, accept the default.
   c. For Region, select **Submit File**.
   d. Click **Next**.
4. Accept the remaining defaults and click **Next**.
5. Click **Create Item**.

   The Page Definition appears.

Create a Button

Next, you need to create a button to submit the file.

To create a button:

1. Under Buttons, click the **Create** icon.
2. For Button Region, select **Submit File (1) 1** and click **Next**.
3. For Button Position, select **Create a button in a region position** and then click **Next**.
4. On Button Attributes:
   a. Button Name - Enter `Submit`.
   b. Accept the remaining defaults.
   c. Click **Next**.
5. For Button Template, accept the default and click **Next**.
6. For Display Properties, accept the defaults and click **Next**.
7. For Branching:
   a. Branch to Page - Select **Page 1**.

This selection causes the page to call itself on submit rather than navigate to another page.

b. Click **Create Button**.

8. Run the page by clicking the **Run Page** icon as shown in Figure 9–2.

Figure 9–2 Run Page Icon

9. If prompted to enter a user name and password, enter your workspace user name and password and click **Login**. See "About Application Authentication" on page 1-6.

When you run the page, it should look similar to Figure 9–3.

Figure 9–3 Submit File Form

Creating a Report with Download Links

Once you create the form to upload documents, the next step is to create a report on the document table that contains links to the uploaded documents. When you use a File Browse item, the files you upload are stored in a table called `wwv_flow_file_objects$`. Every workspace has access to this table through a view called `APEX_APPLICATION_FILES`.

Topics in this section include:

- Create a Report on APEX_APPLICATION_FILES
- Add Link to Download Documents

Create a Report on APEX_APPLICATION_FILES

To create a report on `APEX_APPLICATION_FILES`:

1. Click **Edit Page 1** on the Developer toolbar at the bottom of the page.

   The Page Definition appears.

2. Under Regions, click the **Create** icon.
3. For Region, select **Report** and then click **Next**.
4. For Report Implementation, select **SQL Report** and then click **Next**.
5. For Display Attributes:

   a. Title - Enter `Uploaded Files`.

   b. Accept the remaining defaults and click **Next**.

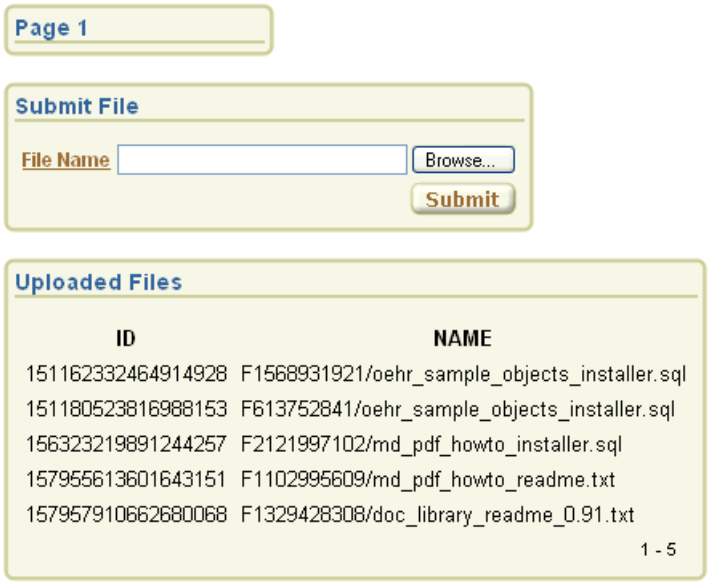
6. For Source, enter the following SQL query:

```
SELECT id,name FROM APEX_APPLICATION_FILES
```

7. Click **Create Region**.
8. Run the page.

Your report should resemble Figure 9–4. Note that your display may differ slightly depending on what files you have uploaded.

Figure 9–4 Uploaded Files Report



Add Link to Download Documents

Next, you need to provide a link to download each document.

To provide a link to download the documents in the report:

1. Click **Edit Page 1** on the Developer toolbar.
2. Under Regions, click **Report** next to Uploaded Files as shown in Figure 9–5.

Figure 9–5 Report Link

The Report Attributes page appears. You can add a link to the ID column by editing Column Attributes.

3. Under Column Attributes, click the **Edit** icon in the ID row.
4. Scroll down to Column Link.
5. Under Column Link:

a. Link Text - Select `#ID#`.

b. Target - Select URL.

c. In the URL field, enter the following:

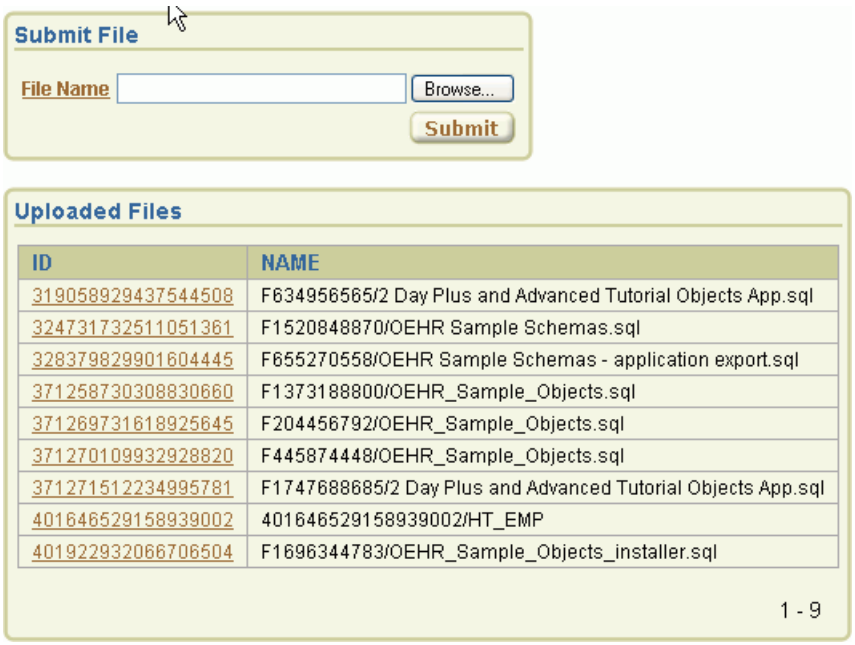
```
p?n=#ID#
```

`#ID#` parses the value contained in the column where `ID` is the column alias.

6. At the top of the page, click **Apply Changes**.
7. Run the page.

   When you run the page, it should look similar to Figure 9–6.

Figure 9–6 Uploaded Files Report with Download Links



8. To test the links, click an ID.

   A File Download dialog box appears.

9. Click **Edit Page 1** on the Developer toolbar to return to the Page Definition.

Storing Additional Attributes About the Document

Next, you create another table to store additional information about the documents that are uploaded. In this exercise, you:

- Add an item to the upload form to capture the information
- Add a process to insert this information along with the name of the file
- Alter the SQL Report of uploaded files to join to the table containing the additional information

Topics in this section include:

- Create a Table to Store Document Attributes
- Create an Item to Capture the Document Subject

- Create a Process to Insert Information
- Show Additional Attributes in the Report Region

Create a Table to Store Document Attributes

First, you create a table in SQL Commands.

> **See Also:** "Using SQL Commands" in *Oracle Application Express Application Builder User's Guide*

To create the table to store additional information about uploaded files:

1. Go to SQL Commands:
   a. Click the **Home** breadcrumb link at the top of the page as shown in Figure 9–7.

Figure 9–7 Breadcrumb Menu

   The Workspace home page appears.
   b. On the Workspace home page, click **SQL Workshop** and then **SQL Commands**.

   The SQL Commands page appears.
2. In the top section, enter:

```
CREATE TABLE oehr_file_subject
   (name     VARCHAR2(4000) primary key,
    subject  VARCHAR2(4000));
```

3. Click **Run**.

   The message `Table created` appears in the Results section.
4. Click the **Home** breadcrumb link.

   The Workspace home page appears.

Create an Item to Capture the Document Subject

To create an item to capture the subject of the document:

1. Go to the Page Definition for page 1:
   a. On the Workspace home page, click the **Application Builder** icon.
   b. On the Application Builder home page, click **Download App**.
   c. On the Application home page, click the **Page 1** icon.

   The Page Definition for Page 1 appears.
2. Under Items, click the **Create** icon.
3. For Item Type, select **Text** and click **Next**.
4. For Text Control Display Type, select **Text Field** and click **Next**.
5. For Display Position and Name:

   a. For Item Name - Enter `P1_SUBJECT`.
   b. Sequence - Accept the default.
   c. Region - Select **Uploaded Files**.
   d. Click **Next**.
6. For Item Attributes:
   a. Label field - Enter `Subject`.
   b. Accept the remaining defaults.
   c. Click **Next**.
7. Click **Create Item**.

Create a Process to Insert Information

Next, you need to create a process to insert the subject information into the new table.

To create a process:

1. Under Page Processing, Processes, click the **Create** icon.
2. For Process Type, select **PL/SQL** and then click **Next**.
3. For Process Attributes:
   a. Name - Enter `Insert file description`.
   b. Sequence- Accept the default.
   c. From Point - Select **On Submit - After Computations and Validations**.
   d. Click **Next**.
4. In Enter PL/SQL Page Process, enter the following:

```
INSERT INTO oehr_file_subject(name, subject) VALUES(:P1_FILE_NAME,:P1_SUBJECT);
```

5. Click **Next**.
6. For Messages:
   a. Success Message - Enter:

```
Subject inserted
```

   b. Failure Message - Enter:

```
Error inserting subject
```

   c. Click **Next**.
7. For Process Conditions:
   a. When Button Pressed - Select **SUBMIT**.
   b. Accept the remaining defaults and click **Create Process**.

Show Additional Attributes in the Report Region

Finally, you need to alter the SQL Report region to join it to the additional attributes table. To accomplish this, you edit the Region Source attribute on the Region Definition page.

To edit the Region Source:

1. Under Regions, click **Uploaded Files**.

   The Region Definition appears.

2. Scroll down to Source.

3. Replace the Region Source with the following:

```
SELECT w.id,w.name,s.subject
FROM APEX_APPLICATION_FILES w,oehr_file_subject s
WHERE w.name = s.name
```

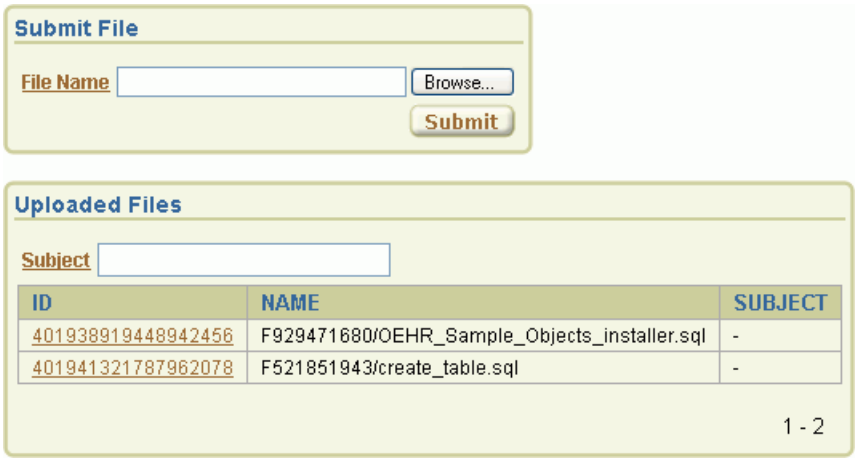
4. Click **Apply Changes**.

5. Run the page.

6. Click **Browse**, locate a file to upload, and click **Submit**.

   As shown in Figure 9–8, the Uploaded Files report now contains a Subject column.

Figure 9–8 Uploaded Files Report with Subject Column



7. Click **Edit Page 1** on the Developer toolbar to return to the Page Definition.

Storing the Document in a Custom Table

In certain cases, you may want to store uploaded documents in a table owned by your schema. For example, if you want to create an Oracle Text index on uploaded documents, you need to store the documents in a custom table.

To store documents in your custom table:

- Add a column of type `BLOB` to hold the document
- Alter the process to insert documents into the custom table

To add a BLOB column to the `oehr_file_subject` table:

1. Go to SQL Commands:

   a. Click the **Home** breadcrumb link at the top of the page.

      The Workspace home page appears.

   b. On the Workspace home page, click **SQL Workshop** and then **SQL Commands**.

      The SQL Commands page appears.

2. In the top section, enter the following SQL statement:

```
ALTER TABLE oehr_file_subject ADD(id number,blob_content BLOB,mime_type
varchar2(4000) );
```

3. Click **Run**.

   The message `Table Altered` appears.

4. Click the **Home** breadcrumb link at the top of the page.

To alter the process to insert documents into the `oehr_file_subject` table:

1. On the Workspace home page, click **Application Builder**.
2. Click **Download App**.
3. Click **Page 1**.
4. Under Processes, click the **Insert file description** link.
5. Scroll down to Source.
6. Under Source, replace the process with the following:

```
IF ( :P1_FILE_NAME is not null ) THEN
   INSERT INTO oehr_file_subject(id,NAME, SUBJECT, BLOB_CONTENT, MIME_TYPE)
    SELECT ID,:P1_FILE_NAME,:P1_SUBJECT,blob_content,mime_type
          FROM APEX_APPLICATION_FILES
          WHERE name = :P1_FILE_NAME;
 DELETE from APEX_APPLICATION_FILES WHERE name = :P1_FILE_NAME;
END IF;
```

7. Click **Apply Changes**.

Downloading Documents from the Custom Table

Now that documents are being stored in a custom table, you need to provide a way to download them. You do this by creating a procedure and granting execute on that procedure to the pseudo user `APEX_PUBLIC_USER`.

To accomplish this you need to change:

- The SQL report region to no longer join to the `APEX_APPLICATION_FILES` view
- The URL supplied for the ID column in the SQL report to execute the new procedure instead of executing the previous procedure

Topics in this section include:

- Create a Procedure to Download Documents
- Edit the Uploaded Files Region
- Change the Download Link to Use the New Procedure

Create a Procedure to Download Documents

To create a procedure to download documents from the `oehr_file_subject` table and grant execute to public:

1. Go to SQL Commands:

   a. Click the **Home** breadcrumb link at the top of the page.

      The Workspace home page appears.

b. On the Workspace home page, click **SQL Workshop** and then **SQL Commands**.

   The SQL Commands page appears.

2. Enter the following SQL statement:

```
CREATE OR REPLACE PROCEDURE download_my_file(p_file in number) AS
        v_mime  VARCHAR2(48);
        v_length  NUMBER;
        v_file_name VARCHAR2(2000);
        Lob_loc  BLOB;
BEGIN
        SELECT MIME_TYPE, BLOB_CONTENT, name,DBMS_LOB.GETLENGTH(blob_content)
                INTO v_mime,lob_loc,v_file_name,v_length
                FROM oehr_file_subject
                WHERE id = p_file;
               --
               -- set up HTTP header
               --
                     -- use an NVL around the mime type and
                     -- if it is a null set it to application/octect
                     -- application/octect may launch a download window from
windows
                     owa_util.mime_header( nvl(v_mime,'application/octet'),
FALSE );

                  -- set the size so the browser knows how much to download
                  htp.p('Content-length: ' || v_length);
                 -- the filename will be used by the browser if the users does a
save as
                  htp.p('Content-Disposition:  attachment;
filename="'||replace(replace(substr(v_file_name,instr(v_file_
name,'/')+1),chr(10),null),chr(13),null)|| '"');
                  -- close the headers
                  owa_util.http_header_close;
                  -- download the BLOB
                  wpg_docload.download_file( Lob_loc );
end download_my_file;
/
```

3. Click **Run**.

   The message `Procedure Created` appears.

   Next, you want to run another SQL statement.

4. Click the **SQL Workshop** breadcrumb link and then click **SQL Commands**.

   The SQL Commands page appears.

5. In the top section, replace the existing SQL statement with the following:

```
GRANT EXECUTE ON download_my_file TO PUBLIC
/
```

6. Click **Run**.

   The message `Statement processed` appears.

7. Click the **Home** breadcrumb link at the top of the page to return to the Workspace home page.

Edit the Uploaded Files Region

To change the SQL report region to no longer join with the `APEX_APPLICATION_FILES` view:

1. Go to the Page Definition of page 1:
   a. On the Workspace home page, click **Application Builder**.
   b. On the Application Builder home page, click **Download App**.
   c. On the Application home page, click **Page 1**.
2. Under Regions, click **Uploaded Files**.
3. Scroll down to Source.
4. Replace the Region Source with the following:

```
SELECT s.id,s.name,s.subject FROM oehr_file_subject s
```

5. Click **Apply Changes**.

   The Page Definition appears.

Change the Download Link to Use the New Procedure

Next you need to change the download link to call the PL/SQL download_my_file procedure.

> **Note:** If you are using Application Express in Database 11g, instead of calling the PL/SQL procedure as described in this section, please follow the steps outlined in the following section Create Download Page for Embedded PL/SQL Gateway.

To change the download link to use the new download procedure:

1. Under Regions, click **Report** next to Uploaded Files.
2. In the ID row, click the **Edit** icon.
3. Scroll down to the Column Link section.
4. In the URL field, replace the existing URL with the following:

```
#OWNER#.download_my_file?p_file=#ID#
```

   In this URL:

   – `#OWNER#` is the parsing schema of the current application.
   – `download_my_file` is the new procedure you just created.
   – You are passing in the value of the column ID to the parameter `p_file`.

5. Click **Apply Changes**.

   The Page Definition appears.

Create Download Page for Embedded PL/SQL Gateway

The Oracle XML DB HTTP Server with the embedded PL/SQL Gateway is typically used for Application Express in Oracle Database 11g. Calling a PL/SQL procedure

directly from a URL that is not known in a list of allowed procedures, as shown in Change the Download Link to Use the New Procedure, results in an error message.

To avoid this situation, there are a couple of available options. The first option is to modify the PL/SQL function `WWV_FLOW_EPG_INCLUDE_MOD_LOCAL` to include the PL/SQL `download_my_file` procedure and then recompile. The second, described below, is to create a page in the application that has a before header branch to the PL/SQL `download_my_file` procedure. You then create a hidden item on that page for the document ID of the document to be downloaded.

To accomplish the second option you need to:

- Create a page with a before header branch to the PL/SQL procedure download_my_file.
- Change the download link to use the new page to display the file.

To create a page with a before header branch to the PL/SQL procedure download_my_file:

1. On the Application Home page, click **Create Page**.
2. Select **Blank Page** and click **Next**.
3. For Page Number, enter 2 and click **Next**.
4. For Name, enter Download File and click **Next**.
5. For Tabs, select **No** and click **Next**.
6. Click **Finish**.

   The Success page appears.
7. Click **Edit Page** icon.

   The Page Definition for page 2 appears.
8. Under Regions, click the **Create** icon.
9. For Region:
   a. Identify the type of region to add to this page - Accept the default, **HTML**, and click **Next**.
   b. Select the type of HTML region container you wish to create - Accept the default, **HTML**, and click **Next**.
10. For Display Attributes, specify the following:
    a. For Title - Enter `Display Document`.
    b. Accept the remaining default values.
    c. Click **Next**.
11. Click **Create Region**.

    The Page Definition for page 2 appears. A confirmation message displays at the top of the page: Region created.
12. Under Items on the Page Definition for page 2, click **Create** icon.
13. For Item Type, select **Hidden** and then click **Next**.
14. For Hidden Item Type, select **Hidden** and then click **Next**.
15. For Item Name, enter **P2_DOC_ID** and then click **Next**.

16. For Source, accept all defaults and then click **Create Item**.

    The Page Definition for Page 2 appears.

17. Under Branches, click the **Create** icon.
18. For the Branch Point list, select **On Load: Before Header**.
19. For the Branch Type, select **Branch to PL/SQL Procedure** and then click **Next**.
20. For the Identify PL/SQL procedure to call text box, enter `download_my_file(:P2_DOC_ID)` and click **Next**.
21. For Branch Conditions, accept defaults and click **Create Branch**.

    The Page Definition for page 2 appears.

Change the download link to display to the Download Display page:

1. Click the Page 1 icon.

   The Page Definition for page 1 appears.

2. Under Regions, click **Report** next to Uploaded Files.

   The Report Attributes page appears.

3. Under Column Attributes, click the **Edit** icon in the ID row.
4. Scroll down to Column Link
5. Under Column Link:
   a. Target - select **Page in this Application**
   b. Page - enter **2**
   c. Item 1 Name - `P2_DOC_ID`
   d. Item 1 Value - `#ID#`
6. Click **Apply Changes**.
7. Run the application.

Security Issues to Consider

The application you built in this tutorial provides download links that invoke the procedure `download_my_file`. Note that this approach has security implications that you need to be aware of.

To invoke your procedure, a user can click the links you provide, or a user can enter similar URLs in the Web browser's Address (or Location) field. Be aware that a curious or malicious user could experiment with your `download_my_file` procedure, passing in any file ID as the `p_file` argument. A hacker could determine what file IDs exist in your table by legitimate or illicit means. Worse yet, in a mechanized attack, a hacker could submit successive IDs until an ID matches a file in your table at which time your procedure would download the file to the hacker.

The measures you take to protect your data from unauthorized access depend upon:

- Your assessment of the degree of harm that would result if a hacker were able to download a file.
- The likelihood of such an attack balanced against the cost and difficulty of providing controls.

One technique you can use to protect an application is to call one of the Oracle Application Express security APIs from within the procedure in order to ensure that the user has already been authenticated. For example, you could include a block of code into the procedure so that it runs first. Consider the following example:

```
-- Assuming your application's numeric ID is 100, set g_flow_id to
--     that value, otherwise change the value as required.
--
APEX_APPLICATION.G_FLOW_ID := 100;

IF NOT wwv_flow_custom_auth_std.is_session_valid then
    --
    --
    -- display this message or a custom message.
    --
htp.p('Unauthorized access - file will not be retrieved.');
    --
    -- You can do whatever else you need to here to log the
    --     unauthorized access attempt, get the requestor's
    --     IP address, send email, etc.
    --
    RETURN;
END IF;
```

10

How to Incorporate JavaScript into an Application

Adding JavaScript to a Web application is a great way to add features that mimic those found in client/server applications without sacrificing all of the benefits of Web deployment. Oracle Application Express includes multiple built-in interfaces especially designed for adding JavaScript.

Remember that JavaScript is not appropriate for data intensive validations. For example, to verify that a name is contained within a large database table, you would need to pull down every record to the client, creating a huge HTML document. In general, complex operations are much better suited for server-side Oracle Application Express validations instead of JavaScript.

This tutorial describes some usage scenarios for JavaScript and includes details about how to implement them in your application.

This section contains the following topics:

Understanding How to Incorporate JavaScript Functions

There are two primary places to include JavaScript functions:

- In the HTML Header attribute of a page
- In a `.js` file referenced in the page template

Topics in this section include:

Incorporating JavaScript in the HTML Header Attribute

One way to include JavaScript into your application is to add it to the HTML Header attribute of the page. This is a good approach for functions that are specific to a page as well as a convenient way to test a function before you include it in a `.js` file.

You can add JavaScript functions to a page by entering the code in the HTML Header attribute on the Page Attributes page.

To add JavaScript code in the HTML Header attribute:

1. On the Workspace home page, click the **Application Builder** icon.
2. Select an application.

   The Application home page appears, displaying its set of pages.
3. Click a page.

   The Page Definition for that page appears.
4. In the Page section, click the **Edit page attributes** icon.

   The Edit Page appears.
5. Scroll down to the HTML Header section.
6. Enter code into HTML Header and then click **Apply Changes**.

For example, adding the following code would test a function accessible from anywhere on the current page.

```
<script type="text/javascript">
  function test(){
    window.alert('This is a test.');
  }
</script>
```

Including JavaScript in a .js File Referenced by the Page Template

In Oracle Application Express you can reference a `.js` file in the page template. This approach makes all the JavaScript in that file accessible to the application. This is the most efficient approach because a `.js` file loads on the first page view of your application, and is then cached by the browser.

The following code demonstrates how to include a .js file in the header section of a page template. Note the line `script src=` that appears in bold.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
    <title>#TITLE#</title>
    #HEAD#
    <script src="http://myserver.myport/my_images/custom.js"
type="text/javascript"></script>
</head>
<body #ONLOAD#>#FORM_OPEN#
```

> **See Also:** "Customizing Templates" in *Oracle Application Express Application Builder User's Guide*

About Referencing Items Using JavaScript

When you reference an item, the best approach is to reference the item name as it is defined within the page. Note that item name is different than the item label. The item name displays on the Page Definition and the label displays on a running page. For example, if you create an item with the name `P1_FIRST_NAME` and a label of `First Name`, you would reference the item using `P1_FIRST_NAME`.

Referencing an item by the item name enables you to use the JavaScript method `getElementById()` to get and set item attributes and values. The following example demonstrates how to reference an item by ID and display its value in an alert box.

```
<script type="text/javascript">
  function firstName(){
    window.alert('First Name is ' + document.getElementById('P1_FIRST_NAME').value
);
  }
 // or a more generic version would be
  function displayValue(id){
    alert('The Value is ' + document.getElementById(id).value );
  }
</script>

  // Then add the following to the "Form Element Attributes" Attribute of the
item:
  onchange="displayValue('P1_FIRST_NAME');"
```

Calling JavaScript from a Button

Calling a JavaScript from a button is a great way to confirm a request. Oracle Application Express uses this technique for the delete operation of most objects. For example, when you delete a button, a JavaScript message appears asking you to confirm your request. Consider the following example:

```
<script type="text/javascript">
  function deleteConfirm(msg)
  {
var confDel = msg;
if(confDel ==null)
  confDel= confirm("Would you like to perform this delete action?");
else
  confDel= confirm(msg);

if (confDel== true)
  doSubmit('Delete');
  }
</script>
```

This example creates a function to confirm a delete action and then calls that function from a button. Note that the function optionally submits the page and sets the value of the internal variable `:REQUEST` to `Delete`, thus performing the delete using a process that conditionally executes based on the value of request.

Note that when you create the button, you need to select **Action Redirect to URL without submitting page**. Then, you specify a URL target, such as the following:

```
confirmDelete('Would you like to perform this delete action?');
```

Changing the Value of Form Elements

In the following example, there are four text boxes in a region. The fourth text box contains the sum of the other three. To calculate this sum, you add a JavaScript function to the HTML Header attribute and then call that function from the first three items.

To add a function to the HTML Header attribute:

1. Go to the appropriate Page Definition.
2. In the Page section, click the **Edit page attributes** icon.

   The Edit Page appears.
3. In the HTML Header section, enter the following:

```
<script type="text/javascript">
  function sumItems(){
    function getVal(item){
   if(document.getElementById(item).value != "")
     return parseFloat(document.getElementById(item).value);
   else
     return 0;
    }
    document.getElementById('P1_TOTAL').value =
  getVal('P1_ONE') + getVal('P1_TWO') + getVal('P1_THREE');
  }
</script>
```

4. Click **Apply Changes**.

To call the function from all three items:

1. Go to the appropriate Page Definition.
2. For each item:
   a. Select the item name by clicking it.
   b. Scroll down to Element.
   c. In HTML Form Element Attributes, enter:

```
onchange="sumItems();"
```

   d. Click **Apply Changes**.

Creating a Client Side JavaScript Validation

Client side validations give immediate feedback to users using a form. One very common JavaScript validation is field not null. This type of validation works well in the page header rather than in a `.js` because it is so specific to a page.

Before you begin, you need to import and install the *OEHR Sample Objects* application in order to access the necessary sample database objects. See "About Loading Sample Objects" on page 1-2.

Topics in this section include:

- Create an Application on the OEHR_EMPLOYEES Table
- Add a Function to the HTML Header Attribute
- Edit an Item to Call the Function

- Test the Validation by Running the Page

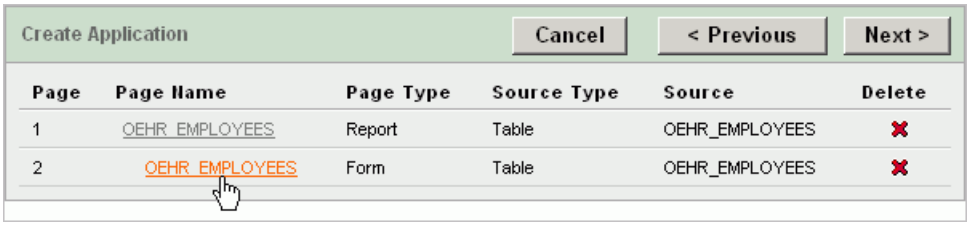
Create an Application on the OEHR_EMPLOYEES Table

To create a new application on the `OEHR_EMPLOYEES` table:

1. On the Workspace home page, click **Application Builder**.
2. Click **Create**.
3. For Method, select **Create Application** and then click **Next**.
4. For Name, specify the following:
   a. Name - Enter `JavaScript Example`.
   b. Application - Accept the default.
   c. Create Application - Select **From scratch**.
   d. Schema - Select the schema where you installed the OEHR sample objects.
   e. Click **Next**.
5. Add a page containing a report by specifying the following in the Add Page section:
   a. Select Page Type - Select **Report and Form**.
   b. Table Name - Select **OEHR_EMPLOYEES**.
   c. Click **Add Page**.

   The new pages appear in the list at the top of the page. Next, change the name of page 2 to **Update Form**.
6. To change the name of page 2:
   a. Under Create Application at the top of the page, click the page name **OEHR_EMPLOYEES** for page 2 as shown in Figure 10–1 on page 10-5.

Figure 10–1 Newly Created Pages



   The Page Definition appears.
   b. In Page Name, enter `Update Form`.
   c. Click **Apply Changes**
   d. Click **Next**.
7. For Tabs, accept the default, **One Level of Tabs** and then click **Next**.
8. For Shared Components, accept the default, **No**, and click **Next**.
9. For Attributes, accept the defaults for Authentication Scheme, Language, and User Language Preference Derived From and click **Next**.
10. For User Interface, select **Theme 2** and click **Next**.

11. Click **Create**.

    The Application home page appears. Note the new application contains three pages:

    - 1 - OEHR_EMPLOYEES
    - 2 - Update Form
    - 101 - Login

To view the application:

1. Click the **Run Application** icon as shown in Figure 10–2.

Figure 10–2 Run Application Icon

2. When prompted for a user name and password, enter your workspace user name and password and click **Login**. See "About Application Authentication" on page 1-6.

   A standard report appears. To view the update form, click either the **Create** button or **Edit** icon.

3. Click **Application** on the Developer toolbar to return to the Application home page.

Add a Function to the HTML Header Attribute

Next, you need to add a function to the HTML Header attribute on page 2 that displays a message when the Last Name field does not contain a value.

To add a function to the HTML Header attribute on page 2:

1. On the Application home page, click **2 - Update Form**.

   The Page Definition for page 2 appears.

2. Under Page, click the **Edit page attributes** icon as shown in Figure 10–3.

Figure 10–3 Edit Page Attributes Icon

   The Edit Page appears.

3. Scroll down to HTML Header.

   Note that HTML Header already contains a script. When the user clicks the Delete button, this script displays the following message.

```
Would you like to perform this delete action?
```

4. In HTML Header, scroll down and place your cursor after the last `</script>` tag.
5. After the last `</script>` tag, enter the following script:

```
<script type="text/javascript">
  function notNull(object){
    if(object.value=="")
   alert('This field must contain a value.');
  }
</script>
```

6. At the top of the page, click **Apply Changes**.

   The Page Definition for page 2 - Update Form appears.

Edit an Item to Call the Function

Next, you need to edit the `P2_LAST_NAME` item to call the function.

To edit the `P2_LAST_NAME` item to call the function:

1. Under Items, click **P2_LAST_NAME**.
2. Scroll down to Element.
3. In HTML Form Element Attributes, enter the following:

```
onblur="notNull(this);"
```

4. At the top of the page, click **Apply Changes**.

   The Page Definition appears. Next, run the page.

Test the Validation by Running the Page

Next, navigate to page 1 and run the page.

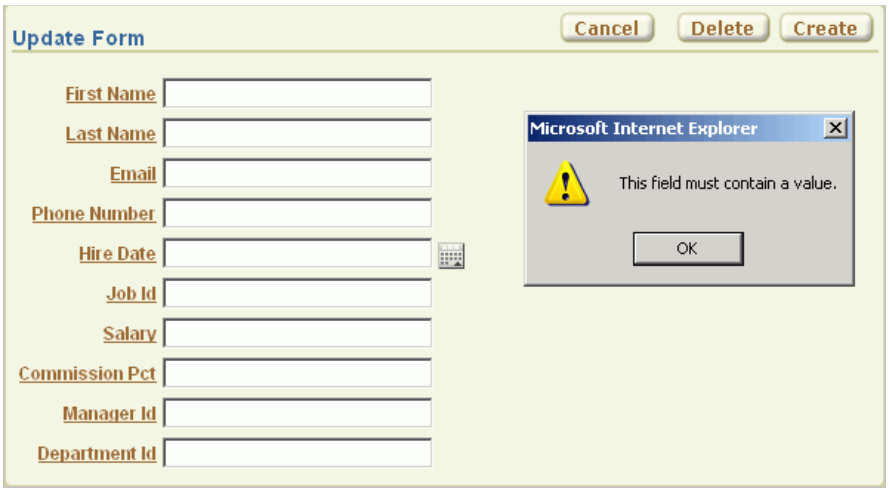
To test the validation:

1. Enter **1** in the Page field on the Page Definition and click **Go**.

   The Page Definition for page 1 appears.
2. Click the **Run Page** icon in the upper right corner.
3. When the application appears, click **Create**.

   The Update Form appears.
4. Position your cursor in the Last Name field and then click **Create**. The message `This field must contain a value` appears as shown in Figure 10–4.

Figure 10–4 Update Form



Enabling and Disabling Form Elements

While Oracle Application Express enables you to conditionally display a page item, it is important to note that a page must be submitted for any changes on the page to be evaluated. The following example demonstrates how to use JavaScript to disable a form element based on the value of another form element.

First, you write a function and place it in the HTML Header attribute of the page containing your update form. Second, you call the function from an item on the page. The following example demonstrates how to add a JavaScript function to prevent users from adding commissions to employees who are not in the Sales Department (P2_DEPARTMENT_ID = 80).

Topics in this section include:

- Add a Function to the HTML Header Attribute
- Edit an Item to Call the Function
- Change the Item to a Select List
- Create a Call to the disFormItems Function

Add a Function to the HTML Header Attribute

To add a function to the HTML Header attribute on page 2:

1. Go to the Page Definition for page 2.
2. Under Page, click the **Edit page attributes** icon.

   The Edit Page appears.
3. Scroll down to HTML Header.
4. In HTML Header, scroll down and place your cursor after the last `</script>` tag.
5. After the last `</script>` tag, enter the following script:

```
<script language="JavaScript1.1" type="text/javascript">
function html_disableItem(nd,a){
     var lEl = document.getElementById(nd);
     if (lEl && lEl != false){
       if(a){
```

```
        lEl.disabled = false;
         lEl.style.background = '#ffffff';
       }else{
       lEl.disabled = true;
         lEl.style.background = '#cccccc';
        }}
    return true;}

function disFormItems(){
 var lOptions = document.getElementById('P2_DEPARTMENT_ID').options
 var lReturn;
 for(var i=0;i<lOptions.length;i++){
    if(lOptions[i].selected==true){lReturn = lOptions[i].value;}
 }
 var lTest = lReturn == '80';
 html_disableItem('P2_COMMISSION_PCT',lTest); }

</script>
```

6. Click **Apply Changes**.

Edit an Item to Call the Function

The next step is to edit the `P2_DEPARTMENT_ID` item and add code to the HTML Form Element Attributes attribute to call the function.

To edit the `P2_DEPARTMENT_ID` item to call the function:

1. Under Items, select **P2_DEPARTMENT_ID**.
2. Scroll down to Element.
3. In HTML Form Element Attributes, enter the following:

   ```
   onchange="disFormItems()"
   ```

4. Click **Apply Changes**.

Change the Item to a Select List

To change the `P2_DEPARTMENT_ID` to display as a select list:

1. Under Items, select **P2_DEPARTMENT_ID**.
2. From the Display As list in the Name section, select **Select List**.
3. Scroll down to List of Values.
4. Under List of Values, specify the following:
   a. From Display Null, select **No**.
   b. In List of Values definition, enter:

      ```
      SELECT department_name, department_id FROM oehr_departments
      ```

5. Click **Apply Changes**.

   Tip: For simplicity, this tutorial has you create an item-level list of values. As a best practice, however, consider creating a named LOV and referencing it.

See Also: "Creating Lists of Values" in *Oracle Application Express Application Builder User's Guide*

Create a Call to the disFormItems Function

Finally, you need to create a call to the `disFormItems` function after the page is rendered to disable `P2_COMMISSION_PCT` if the selected employee is not a Sales representative. A good place to make this call would be from the Page HTML Body Attribute.

To create a call to the `disFormItems` function:

1. Go to the Page Definition for page 2.
2. Under Page, click the **Edit page attributes** icon.

   The Edit Page appears.
3. Locate the Display Attributes section.
4. From Cursor Focus, select **Do not focus cursor**.

   Selecting **Do not focus cursor** prevents conflicts between generated JavaScript and custom JavaScript.
5. Scroll down to the HTML Body Attribute section.
6. In the Page HTML Body Attribute, enter the following:

   ```
   onload="disFormItems(); first_field();"
   ```
7. Click **Apply Changes**.
8. Run the page.

Figure 10–5 on page 10-10 demonstrates the completed form. Note that Department ID displays as a select list. Also notice that the Commission Pct field is unavailable since the Department ID is Administration.

Figure 10–5 Revised Update Form

11

How to Build an Access Control Page

You can control access to an application, individual pages, or page components by creating an Access Control Administration page. The page contains a list of application modes and an Access Control List.

This tutorial explains how to build an Access Control Administration page and then restrict access to an application so that only privileged users can perform specific functions.

This section contains the following topics:

How Access Control Administration Works

You create an access control list by running the Access Control Wizard to create an Access Control Administration page. This page contains a list of application modes and an Access Control List. Once you create the Access Control Administration page, you:

1. Run the Access Control Administration page.
2. Select an application mode:
   - Full access to all, access control list not used.
   - Restricted access. Only users defined in the access control list are allowed.
   - Public read only. Edit and administrative privileges controlled by access control list.
   - Administrative access only.
3. Add users to the Access Control List.

In addition to creating the Access Control Administration page, the Access Control Wizard also creates:

- two tables within the application's default schema to manage access control

- the authorization schemes that correspond to the application mode list options
- the privileges available in the Access Control List

You can control access to a specific page or page component by selecting one of these authorization schemes on the page or component attributes pages. Once you create an Access Control, you can customize the page, tables and values to suit the specific needs of your application.

Creating an Application

First, you need to create an application based on employee data in a spreadsheet.

Topics in this section include:

- Download Spreadsheet Data
- Create an Application Based on Spreadsheet Data
- Run the Application

Download Spreadsheet Data

Download the following *.csv file to your local system:

1. In your Web browser go to:

   `http://www.oracle.com/technology/products/database/application_express/packaged_apps/acl_employees.zip`

2. Download the `acl_employees.zip` file to your computer.
3. Unzip and extract the `acl_employees.csv` file:
   - Microsoft Windows - Double-click the `acl_employees.zip` file
   - UNIX or Linux - Enter the following command:

     ```
     $ unzip acl_employees.zip
     ```

Create an Application Based on Spreadsheet Data

To create a new application based on spreadsheet data:

1. On the Workspace home page, click the **Application Builder** icon.

   The Application Builder home page appears.
2. Click **Create**.
3. Select **Create from Spreadsheet** and click **Next**.
4. Select **Upload file, comma separated (*.csv) or tab delimited** and click **Next**.
5. For Load Method:
   a. Select **Upload file, comma separated (*.csv) or tab delimited**.
   b. Click **Next**.
6. For Data:
   a. Text File - Click **Browse** and navigate to the `acl_employees.csv` file.
   b. Accept the remaining defaults and click **Next**.

7. For Table Properties:
   a. Schema - Select the appropriate schema.
   b. Table Name - Enter `ACL_EMPLOYEES`.
   c. Accept the remaining defaults and click **Next**.
8. For User Interface Defaults:
   a. Singular Name - Enter `Employee`.
   b. Plural Names - Enter `Employees`.
   c. Click **Next**.
9. For Summary Page:
   a. Summary by Column - Select `DEPARTMENT_ID` and click **Next**.
   b. Aggregate by Column - Do not make a selection and click **Next**.
10. For Application Options, accept the defaults and click **Next**.
11. For User Interface, select **Theme 2** and click **Next**.

    A theme is collection of templates that define the layout and style of an application. You can change a theme at any time.
12. Click **Create**.

    The Application home page appears.

Run the Application

To run the application:

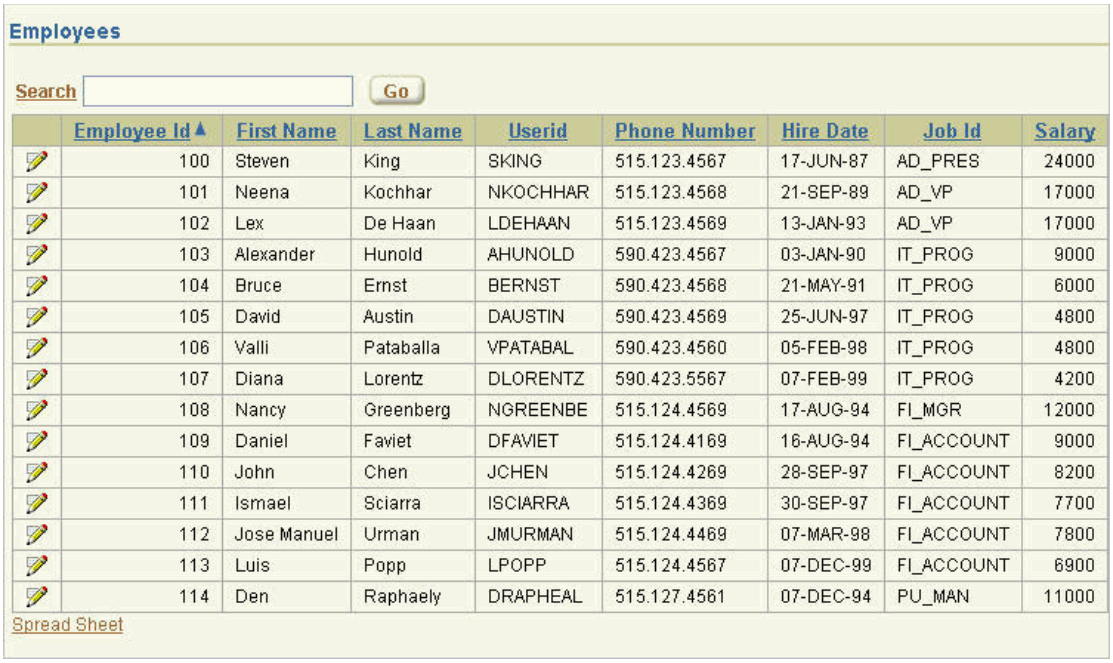
1. Click the **Run Application** icon as shown in Figure 11–1.

Figure 11–1 Run Application Icon

2. If prompted to enter a user name and password, enter your workspace user name and password and click **Login**. See "About Application Authentication" on page 1-6.

   The report appears as shown in Figure 11–2 on page 11-4.

Figure 11–2 ACL_EMPLOYEES Application



The `ACL_EMPLOYEES` application enables you to view and update employee data. To update a specific record, click the **Edit** icon in the far left column. Clicking the **Analyze** tab provides you with access to both a visual and tabular breakdown of the number of employees in each department.

Creating an Access Control Administration Page

Next, you need to secure your application so that only privileged users can perform certain operations. When you implement access control on an Oracle Application Express application, the best approach is to use an authorization scheme defined at the application level. The first step is to create an access control page by running the Access Control Page Wizard.

Topics in this section include:

- Create an Access Control Page
- View the Page
- Add Users to the Access Control List

Create an Access Control Page

To create an access control page:

1. Click **Create** on the Developer toolbar.
2. Select **New page** and click **Next**.
3. For Page, select **Access Control** and click **Next**.

   The Access Control Wizard appears.
4. In Administration Page Number, enter `8` and click **Next**.
5. For Tabs:

a. Tab Options - Select **Use an existing tab set and create a new tab within the existing tab set**.

b. Tab Set - Select **TS1 (Employees, Analyze)**.

c. Tab Set Label - Enter `Administration`.

d. Click **Next**.

6. Review the confirmation page and click **Finish**.

   A Success page appears.

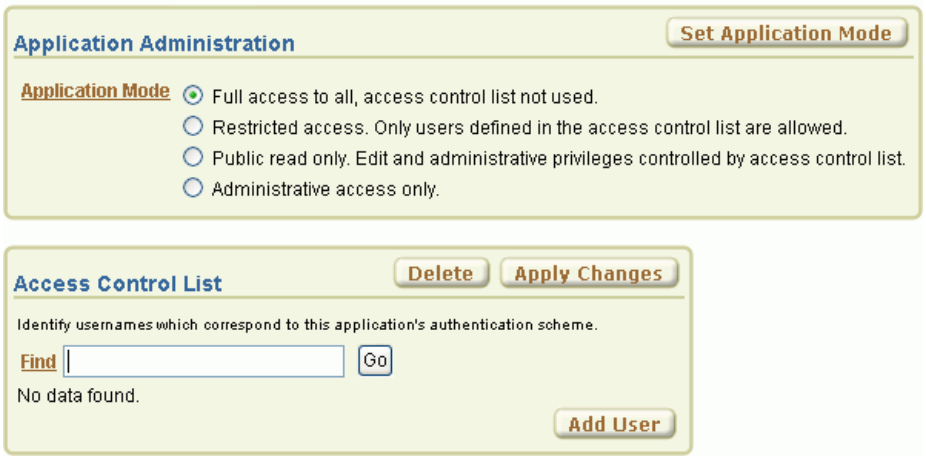
View the Page

To run the page:

1. Click **Run Page**.

   A new page appears as shown in Figure 11–3.

Figure 11–3 Access Control Administration Page



   Notice the page is divided into two regions: Application Administration and Access Control List. Also note that the default Application Mode is Full Access.

2. Under Application Mode, select **Restricted access. Only users defined in the access control list are allowed**.

3. Click **Set Application Mode**.

Add Users to the Access Control List

Next, add three users to the Access Control List:

- Luis Popp (`LPOPP`) will have View privileges.
- Adam Fripp (`AFRIPP`) will have Edit privileges.
- John Chen (`JCHEN`) will have Administrator privileges.

To add users to the Access Control List:

1. Under Access Control List, click **Add User**.

   A new row appears.

2. Enter the first user:

a. Username - Enter `LPOPP`.

b. Privilege - Select **View**.

c. Click **Apply Changes**.

d. Click **Add User** to add a blank row where you can enter the first user.

3. Enter the next user:

a. Username - Enter `AFRIPP`.

b. Privilege - Select **Edit**.

c. Click **Apply Changes**.

d. Click **Add User** to add a blank row where you can enter the next user.

4. Enter the next user:

a. Username - Enter `JCHEN`.

b. Privilege - Select **Administrator**.

c. Click **Apply Changes**.

5. Click **Application** on the Developer toolbar.

   The Application home page appears.

Creating an Authentication Function

Next, you need to make employees in the ACL_EMPLOYEES table the users of the application. To accomplish this, you create a simple authentication function in the current authentication scheme. Note that the function checks for the userid and its associated last name as a password.

To create the authentication function:

1. On the Application Builder home page, click the **Home** breadcrumb link.

   The the Workspace home page appears.

2. Click **SQL Workshop** and then **SQL Commands**.

3. In the SQL editor pane:

a. Enter the following code:

```
CREATE OR REPLACE FUNCTION acl_custom_auth (
    p_username IN VARCHAR2,
    p_password IN VARCHAR2)
RETURN BOOLEAN IS
BEGIN
  FOR c1 IN (SELECT 1
               FROM acl_employees
              WHERE upper(userid) = upper(p_username)
                AND upper(last_name) = upper(p_password))
  LOOP
    RETURN TRUE;
  END LOOP;
  RETURN FALSE;
END;
/
```

b. Click **Run**.

4. Click the **Home** breadcrumb link.

   The Workspace home page appears.

Updating the Current Authentication Scheme

Next, you need to update the current authentication scheme to use the new function.

To update the current authentication scheme.

1. Click **Application Builder** and then click **ACL Employees**.

   The Application home page appears.
2. Click **Shared Components**.
3. Under Security, click **Authentication Schemes**.

   The Authentication Schemes page appears.
4. Click the **Application Express - Current** icon.
5. Scroll down to Login Processing.
6. In Authentication Function, replace `-BUILTIN-` with the following:

   ```
   return acl_custom_auth
   ```

7. Scroll back to the top of the page and click **Apply Changes**.

Applying Authorization Schemes to Components

Next you need to associate the authorization scheme with the appropriate application components. As you may recall, you previously added three users to the Access Control List:

- `LPOPP` had `View` privileges.
- `AFRIPP` had `Edit` privileges
- `JCHEN` had `Administrator` privileges

In this exercise, you associate the View, Edit, and Administrator privileges with specific application components to control which users are allowed to perform what actions.

Topics in this section include:

Associate an Authorization Scheme with the Application

First, you need to specify that users will only be able to access the application if they have View privileges. To accomplish this, you associate the **access control - view** authorization scheme with the application.

To associate an authorization scheme with your application:

1. Click the **Application ID** breadcrumb link.

   The Application home page appears.
2. Click **Shared Components**.
3. Under Application, click **Definition**.
4. Click the **Security** tab.
5. Scroll down to Authorization.
6. From Authorization Scheme, select **access control - view**.
7. Click **Apply Changes** at the top of the page.

Associate Edit Privileges with the ID Column

For this exercise, only users with at least Edit privileges should be able to edit or delete data. To accomplish this, you associate the **access control - edit** authorization scheme with the ID column. This hides the Edit icon on page 1 for users with View privileges, but displays it for users with Edit or Administrator privileges.

To associate edit privileges with the ID column:

1. Click the **Application ID** breadcrumb link.

   The Application home page appears.
2. Click **1 - Report Page**.

   The Page Definition for page 1 appears.
3. Under Regions, click the **Report** link.

   The Report Attributes page appears.
4. Click the **Edit** icon for ID. The Edit icon resembles a small page with a pencil on top of it.

   The Column Attributes page appears.
5. Scroll down to Authorization.
6. From Authorization Scheme, select **access control - edit**.
7. Click **Apply Changes** at the top of the page.

Associate Edit Privileges with the Create Button

Next, associate the **access control - edit** authorization scheme to the Create button. This will hide the Edit icon for unprivileged users.

To associate edit privileges with the Create button:

1. Go to the Page Definition for Page 1. Click the **Page 1** breadcrumb link.

   The Page Definition for page 1 appears.
2. Under Buttons, click the **Create** link (not the icon).
3. Scroll down to Authorization.
4. From Authorization Scheme, select **access control - edit**.
5. Click **Apply Changes** at the top of the page.

   The Page Definition for Page 1 appears.

Associate Edit Privileges with Page 2

Next, associate the **access control - edit** authorization scheme with page 2.

To specify an authorization scheme for page 2:

1. Go to page 2. In the Page field enter 2 and click **Go**.

   The Page Definition for page 2 appears.
2. Under Page, click the **Edit page attributes** icon.
3. Scroll down to Security.
4. From Authorization Scheme, select **access control - edit**.
5. Click **Apply Changes** at the top of the page.

Restrict Access to Page 8

Lastly, you need to restrict access to page 8, Access Control Administration. To accomplish this, you specify the **access control - administrator** authorization scheme with all of page 8 and with the Administration tab.

Specify an Authorization Scheme for Page 8

To specify an authorization scheme for page 8:

1. Go to page 8. In the Page field, enter 8 and click **Go**.

   The Page Definition for page 8 appears.
2. Under Page, click the **Edit page attributes** icon.
3. Scroll down to Security.
4. From Authorization Scheme, select **access control - administrator**.
5. Click **Apply Changes** at the top of the page.

   The Page Definition for page 8 appears.

Specify an Authorization Scheme for the Administration Tab

To specify an authorization scheme for page 8:

1. Under Tabs, click the **Administration** link.
2. Scroll down to Authorization.
3. From Authorization Scheme, select **access control - administrator**.
4. Click **Apply Changes** at the top of the page.

   The Page Definition for page 8 appears.

Testing the Application

At the beginning of this tutorial, you added three users to the Access Control List:

- Luis Popp (LPOPP) has View privileges
- Adam Fripp (AFRIPP) has Edit privileges
- John Chen (JCHEN) has Administrator privileges

Next, test your application by logging in as each of these users.

Topics in this section include:

- Log In with View Privileges
- Log In with Edit Privileges
- Log In with Administrator Privileges

Log In with View Privileges

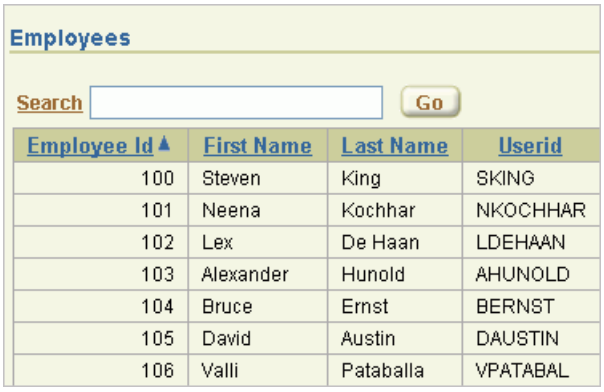
Luis Popp (LPOPP) has View privileges.

To log in as Luis Popp:

1. Click the **Run Page** icon in the upper right corner.
2. When prompted, specify the following:
   a. Username - LPOPP.
   b. Password - Popp.
   c. Click **Login**.

   The Employees Report page appears as shown in Figure 11–4.

Figure 11–4 Employees Report with View Privileges



Note that the Edit icon and the Administration tab no longer appear.

3. Click **Logout** in the upper right corner.

Log In with Edit Privileges

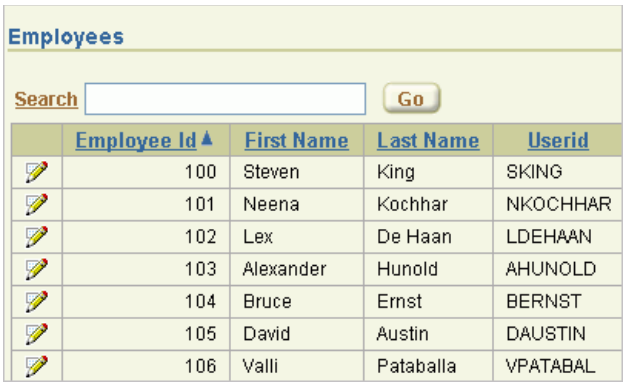
Adam Fripp (AFRIPP) has Edit privileges.

To log in as Adam Fripp:

1. When prompted, specify the following:
   a. Username - AFRIPP.
   b. Password - Fripp.
   c. Click **Login**.

   The Employees Report page appears as shown in Figure 11–5.

Figure 11–5 Employees Report with Edit Privileges



Note that the Edit icon now appears to the left of the Employee Id column, but the Administration tab still does not appear.

2. Click **Logout** in the upper right corner.

Log In with Administrator Privileges

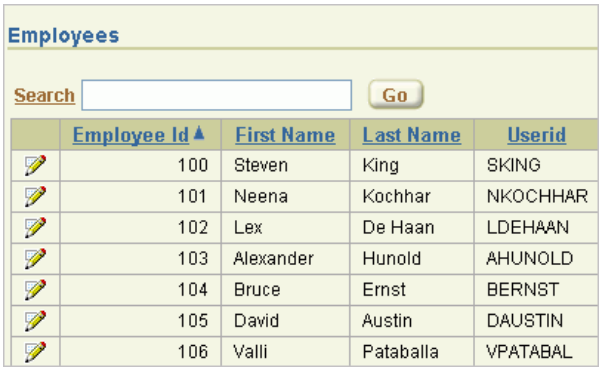
John Chen (JCHEN) has Administrator privileges.

To log in as John Chen:

1. When prompted, specify the following:
   a. Username - JCHEN
   b. Password - Chen
   c. Click **Login**.

   The Employees Report page appears as shown in Figure 11–6.

Figure 11–6 Employees Report with Administrator Privileges



2. Click **Administrator** tab.

12

How to Review a Packaged Application

A packaged application is a fully functional application that you can view, use, and customize. Each packaged application includes installation scripts that define the application's supporting objects (including database objects, images, and seed data) as well as any preinstallation validations. Packaged applications can also include a de-installation script which can be used to remove all the application's supporting objects.

This tutorial walks you through the *OEHR Sample Objects* packaged application. By reviewing the supporting objects behind this application, you can learn how to define them in your own applications.

Before you begin, you need to import and install the *OEHR Sample Objects* application. See "About Loading Sample Objects" on page 1-2.

This section contains the following topics:

What Is a Packaged Application?

Importing and installing an application is a complicated process. First, you create the target database objects and seed data. Second, you import and install the application definition and all related files, including images, themes, and any other required static files.

Creating a packaged application simplifies this process. Instead of performing numerous steps to create the database objects and then import and install the application and all supporting files, you can define the supporting objects so that the application and supporting files can be installed using one simple wizard.

After users import and install the application definition, a wizard guides them through a few simple configuration steps. Then, the wizard asks whether or not to install the supporting application objects. Users have the option of installing the supporting application objects then or doing it later.

See Also: "How to Create a Packaged Application" in *Oracle Application Express Application Builder User's Guide*.

About the Supporting Object Utility

In "About Loading Sample Objects", you imported and installed the *OEHR Sample Objects* application. Installing this application creates the database objects and loads the sample data needed to complete the tutorials in this guide.

You create a packaged application like *OEHR Sample Objects* using the Supporting Object Utility. By creating a packaged application, users can install and deinstall your application as well as the underlying database objects, files, and other supporting objects. Supporting objects consists of everything that your application needs to work properly, including the application definition, images, themes, and any other required static files.

Creating a packaged application is also an effective way to move an application to another Oracle Application Express instance or to share an application with others. The Supporting Object Utility provides the easiest way to ensure that all objects your application depends on are migrated in an efficient manner.

Accessing the Supporting Objects Page

You access the Supporting Object Utility on the Supporting Objects page.

To go to the Supporting Objects page:

1. On the Workspace home page, click **Application Builder**.
2. Click **OEHR Sample Objects**.
3. On the Application home page, click **Supporting Objects**.

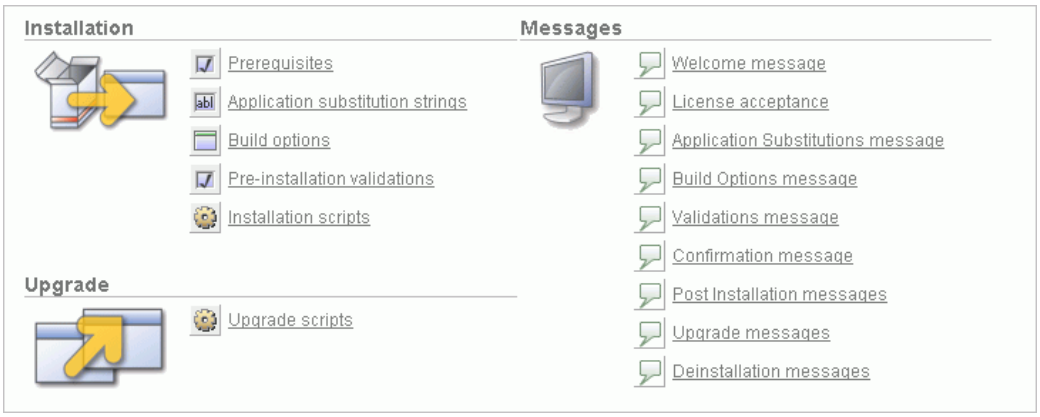
   The Supporting Objects page appears.

The top of the Supporting Objects page displays a detailed report about the currently selected application, including the number of scripts and the amount of required free space. After that, the page is divided into these sections:

- Installation
- Upgrade
- Deinstallation
- Messages

About Supporting Objects Installation

During the installation of supporting objects, a sequence of scripts and validations are run. If a validation fails, the installation process stops. At certain points in the process, messages also display for users to review.



This section describes the Installation and Messages sections of the Supporting Objects page. Depending upon the needs of your application, you may use all or only some of the functionality.

In order to give context, this section describes the supporting objects in the sequence that they would appear to a user who is performing the installation.

Topics in this section include:

- Welcome Message
- Prerequisites
- License Acceptance
- Application Substitution Strings
- Build Options
- Pre-Installation Validations
- Pre-Installation Confirmation Message
- Installation Scripts
- Success Message

Welcome Message

The welcome message contains text that displays to users immediately after they indicate that they want to install the supporting objects. Use this message to introduce the application to users and to explain any information that they may need during the rest of the installation process.

To review the welcome message text:

1. Go to Supporting Objects page as described in "Accessing the Supporting Objects Page" on page 12-2.
2. In the Messages section, click **Welcome message**.

   The Supporting Object Messages page appears. The Welcome message appears in the first section.

Prerequisites

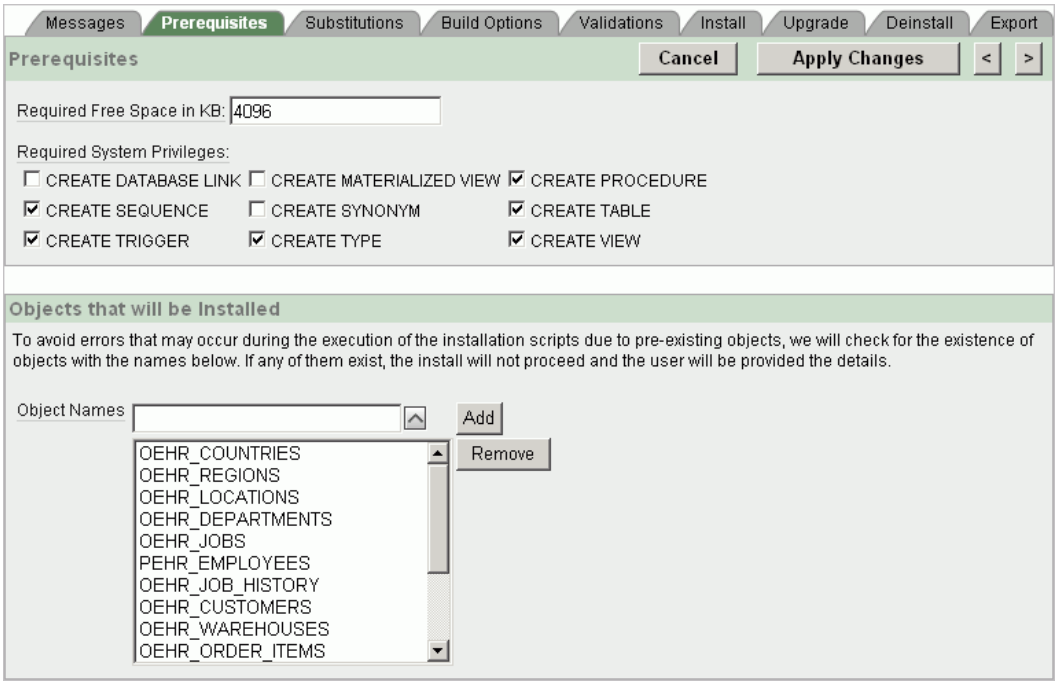
Prerequisites include anything that must be valid for the installation process to continue.

To review the prerequisites:

1. On the Supporting Object Messages page, click the **Prerequisites** tab.

   The Prerequisites page displays the following sections:

   - Required Free Space in KB
   - Required System Privileges
   - Objects that will be Installed



About Required Free Space

If you are creating tables and loading data during the installation, you should specify the amount of required free space. If there is not enough free space available, specifying this prerequisite prevents an error during the installation of supporting objects. For the *OEHR Sample Objects* application, the required free space is 4800 kilobytes (KB).

To determine the space that your objects require:

1. Remove your supporting objects by either manually deleting them, or running your deinstallation script.

   a. Go to the Supporting Objects page as described in "Accessing the Supporting Objects Page" on page 12-2.

   b. From the Tasks list on the right side of the page, click **Deinstall Supporting Objects**.

   c. On the Deinstall page, select **Deinstall Supporting Objects** and click **Deinstall**.

2. View that amount of used space:

   a. Go to the Workspace home page.

   b. From the Administration list on the right, click **Manage Services**.

   c. Under Workspace section, click **Workspace Overview**.

d. Click the **Detailed Tablespace Utilization Report (may take several seconds)** link.

   The Detailed Tablespace Utilization Report appears.

e. Write down the number that appears in the Amount Used column.

   This shows the amount of used space in the tablespace where your schema is located.

3. Install your supporting objects by either manually creating them again, or running your installation scripts.

   a. Go to the Supporting Objects page as described in "Accessing the Supporting Objects Page" on page 12-2.

   b. From the Tasks list on the right side of the page, click **Install Supporting Objects**.

   c. For Install Supporting Objects, select **Yes** and click **Next**.

   d. Click **Install**.

4. View that amount of used space again:

   a. Go to the Workspace home page.

   b. From the Administration list on the right, click **Manage Services**.

   c. Under Workspace section, click **Workspace Overview**.

   d. Click the **Detailed Tablespace Utilization Report (may take several seconds)** link.

      The Detailed Tablespace Utilization Report appears.

   e. Write down the number that appears in the Amount Used column.

5. Subtract the initial Amount of Used number from the new number to determine your required free space.

   Tip: The value in the Amount Used column is in megabytes (MB). To calculate the amount in kilobytes (KB), multiply the final figure by 1,024 (1 MB = 1,024 KB).

About Required System Privileges

The Required System Privileges section contains a list of system privileges. If you are creating objects through installation scripts, you need to select the appropriate system privileges. Selecting these privileges runs a pre-installation check to determine if the user has the appropriate privileges. This prevents the user from creating some objects but not others due to a lack of privileges.

For the *OEHR Sample Objects* application, required system privileges include: `CREATE SEQUENCE`, `CREATE TRIGGER`, `CREATE TYPE`, `CREATE PROCEDURE`, `CREATE TABLE`, and `CREATE VIEW`.

About Objects Installed

The Objects that will be Installed section lists all objects that your installation script creates. If the target schema contains an object with the same name, an error displays to the user. This prerequisite prevents the user from getting errors during installation that result in some objects being created while others are not created because an object of the same name exists.

This prerequisite is important because of the deinstallation implications. If a user installs and already has an object with the same name, the deinstallation script might drop the previously existing object.

For example, if a PROJECTS table already exists and the installation script creates a table with the same name, you would want to alert users before they start the installation process. This warning provides them with the opportunity to rename their table or decide to not install supporting objects.

License Acceptance

If your application requires that users accept a specific license, specify the terms in this area. For a required license, users are prompted to accept the terms. If they do not, the installation does not proceed.

To review the license area:

1. On the Supporting Object Messages page, click the **Messages** tab.
2. Locate the section, License.

Since the *OEHR Sample Objects* application does not require a license, no text appears in the License section.

Application Substitution Strings

Each application can include substitution strings defined within the Application Definition. You can use substitution strings to include phrases or labels occurring in many places within an application that might need to be changed for specific installations.

When you define substitution strings for your application, they display within the Supporting Object Utility. You can then decide which substitution strings you want to display during the installation process.

For substitution strings that display during the installation process, you can provide a custom prompt to explain what each string is used for to assist the user in determining the proper value. You can also define a custom header message if you are including substitution strings.

Since the *OEHR Sample Objects* application does not contain any substitution strings, in the next section you add two substitution strings and then view them the Supporting Object Utility.

Add Substitution Stings

To add substitution strings to the *OEHR Sample Objects* application:

1. Go to the Shared Components page:
   a. Click the **Application** breadcrumb.
   b. Click **Shared Components**.
2. Under Application, click **Definition**.
3. Scroll down to the Substitutions region.
4. Enter the following Substitution Strings and Substitution Values:

Substitution String	Substitution Value
APP_DATE_FORMAT	DD-MON-YYYY

Substitution String	Substitution Value
APP_DATETIME_FORMAT	DD-MON-YYYY HH24:MI

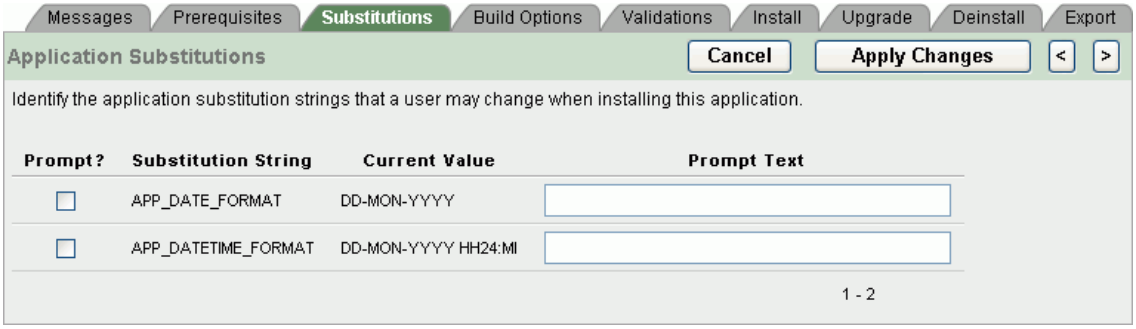
5. Scroll back to the top and click **Apply Changes**.

Review Substitution Strings

To view the substitution strings in Supporting Object Utility:

1. Go to the Supporting Objects page:
   a. Click the **Application** breadcrumb.
   b. Click **Supporting Objects**.
2. In the Installation section, click **Application Substitution strings**.

   The Edit Substitutions page appears, showing the two substitution strings that you defined. If have a prompt appear during installation, select it and enter text in the Prompt Text field.



Build Options

Build options are shared components that enable you to conditionally display objects. Each build option has a status set to Include or Exclude. If an object is associated with a disabled build option, the object does not appear to the user.

Use build options to include functionality in an application that may not be ready for use, or should not be accessible to all installations. The code is included in the application but not exposed to end users. Later, you can enable a build option so that the feature becomes accessible. For each build option, you can define a custom header message to display to the user.

Build options display in the same way as the substitution strings. You specify which build options you want to appear to users. The users can then select them and determine their status.

The OEHR Sample Objects application does not contain any build options.

> **See Also:** "Using Build Options to Control Configuration" in *Oracle Application Express Application Builder User's Guide*

Review the Build Options Page

To review the Build Options page:

1. On the Edit Substitutions page, click the **Build Options** tab.
2. To view the Build Options message:

a. Click the **Messages** tab.

b. Scroll down to the Build Options section.

Pre-Installation Validations

Validations ensure that the target database and target schema are capable of running the installation scripts. Built-in validation types include Current Language, Exists, and so on. You can also use any SQL or PL/SQL expressions. Validations can also be conditionally executed.

Use these validations to check for anything that is not built-in under Prerequisites. For example, you can check for a minimum database version or for the installation of Oracle Text. Just as with other supporting objects, there is a Validations Message.

The *OEHR Sample Objects* application does not contain any validations.

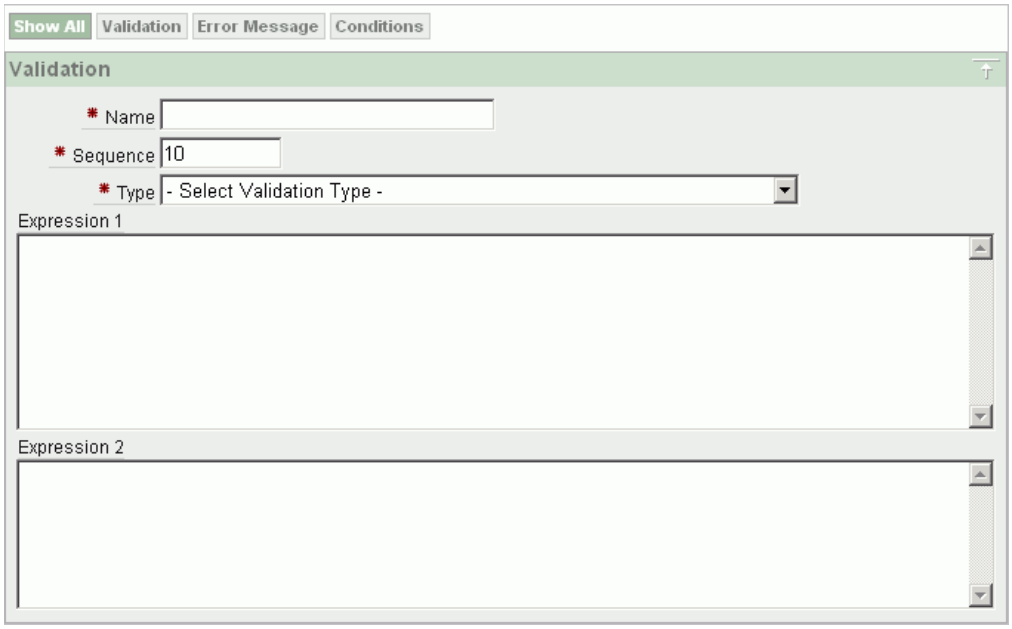
> **See Also:** "Understanding Validations" in *Oracle Application Express Application Builder User's Guide*

Review the Validation Page

To review the Validations page:

1. Click the **Validations** tab.
2. Click **Create**.

   The Validation page appears.



3. For this exercise, click **Cancel**.
4. To view the Validations Message:
   1. Click the **Messages** tab.
   2. Scroll down to the Validations section.

Pre-Installation Confirmation Message

If the installation process is not terminated early, one last confirmation message appears before the installation scripts display to the user.

To review the confirmation message:

1. Click the **Messages** tab.
2. On the Supporting Object Messages page, scroll down to the Confirmation section.

Installation Scripts

Installation scripts are the core of a supporting object installation. Each application can have several installation scripts.

The *OEHR Sample Objects* application contains nine installation scripts. These scripts create objects and load data.

> **Tip:** You can also create a custom installation script that loads files.
> See "About Creating Scripts to Install Files" on page 12-13.

Topics in this section include:

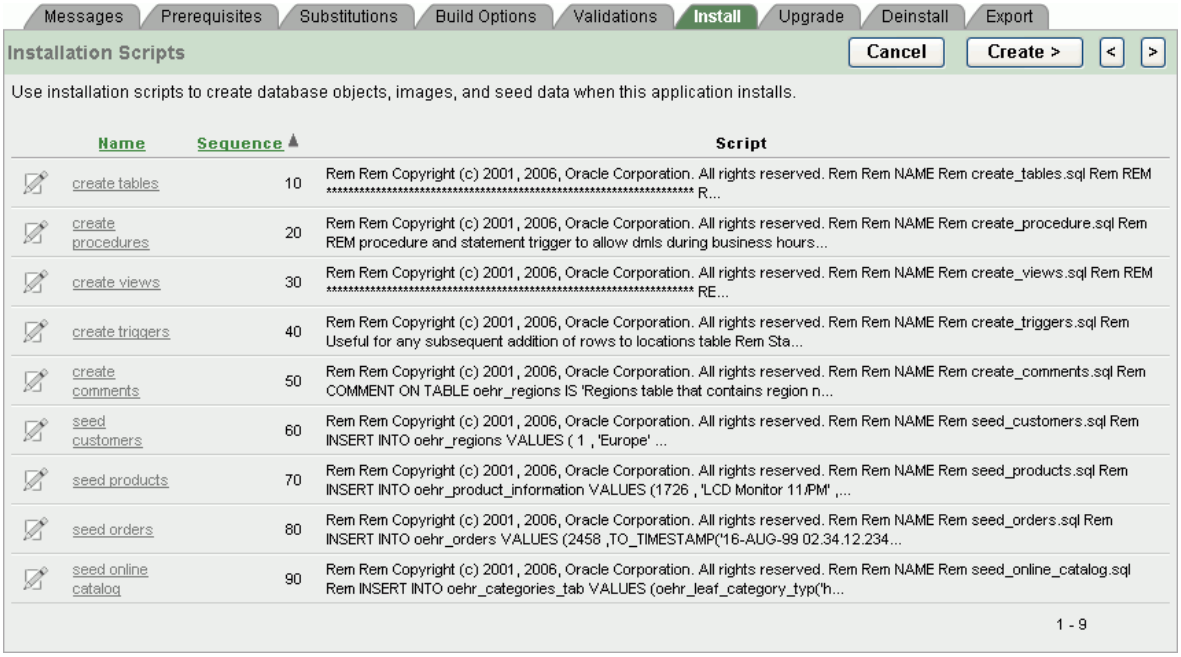
- View Installation Scripts
- Review an Existing Installation Script
- About Creating Installation Scripts from Files
- About Creating Scripts to Install Files

View Installation Scripts

To review the installation scripts in the *OEHR Sample Objects* application:

- On the Supporting Object Messages page, click the **Install** tab.

  The Installation Scripts page shows the list of scripts.



Notice that each script has a name and a sequence. It is very important to order your installation scripts so that dependent objects are created or compiled correctly.

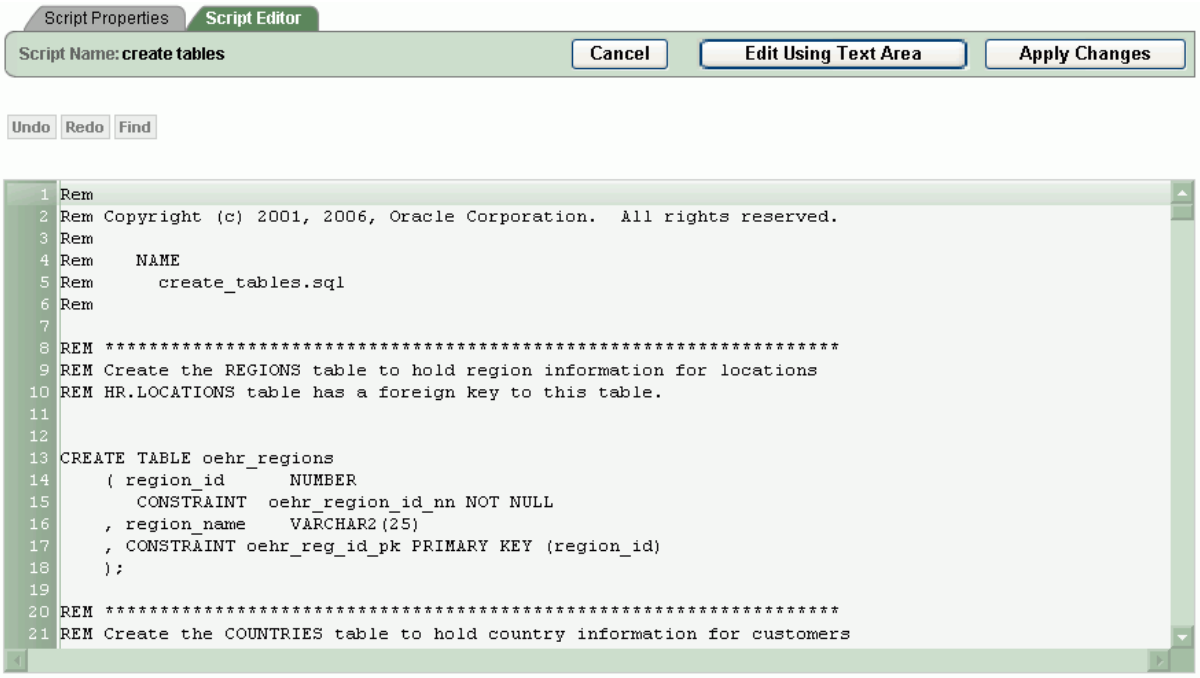
Review an Existing Installation Script

To review or update an existing script:

1. Click the **Edit** icon adjacent to the script name.

   The Script Editor page displays the script for you to edit or review.

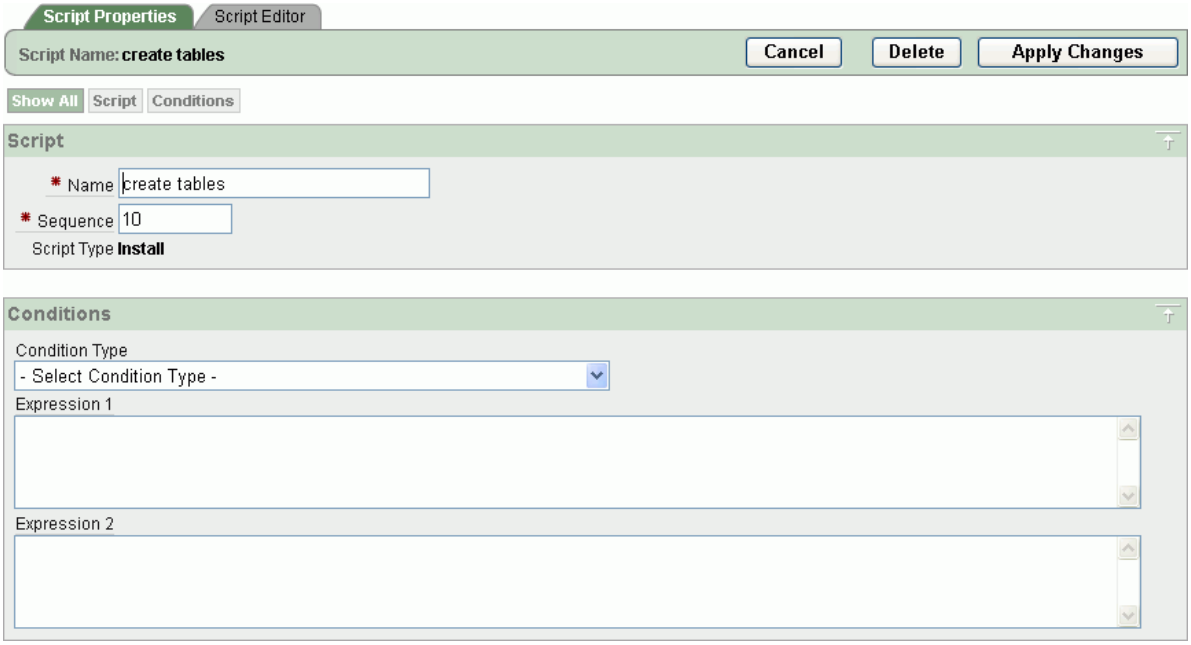
Figure 12–1 Supporting Object Script Editor



> **Tip:** You can toggle between this page and a text page, where you can also edit the script, by clicking the **Edit Using Text Area** button.

2. To view and edit the properties of the script, click the **Script Properties** tab.

   Use the Script Properties page to change the script name or sequence. You can also specify if it should be conditionally executed.



3. For this exercise, click **Cancel**.

About Creating Installation Scripts from Files

You can create objects and install seed date through files that you create from scratch or upload.

Topics in this section include:

- Creating a New Script from Scratch
- Uploading a File
- Creating Scripts for Access Control Tables

Creating a New Script from Scratch To create a new script from scratch:

1. Go to the Installation Scripts page:
   a. Click the **Supporting Objects** breadcrumb.
   b. Under Installation, click **Installation scripts**
2. On the Installation Scripts page, click **Create**.
3. Accept the default, **Create from Scratch**, and click **Next**.

   The Create Script wizard appears.
4. For Script Attributes:
   a. Name - Enter `Test Script`.
   b. Sequence - Accept the default, **100**.
   c. Use Editor - Select this option.

      Selecting **Use Editor** enables you to use the built-in editor to define your script. If you do not select this, a standard text area appears instead.
   d. Click **Next**.

   The Script Editor opens, where you can type in or paste your script contents.

5. For this exercise, click **Cancel**.

Uploading a File Often, you have files that you want to use as the basis of your installation scripts.

To upload a file:

1. On the Installation Scripts page, click **Create**.
2. Select **Create from file** and click **Next**.
3. For Script Attributes:
   a. Name - Enter `Test Script`.
   b. Sequence - Accept the default, **100**.
   c. Click **Next**.
4. For Define Script, select the file to upload.
5. For this exercise, click **Cancel**.

Creating Scripts for Access Control Tables If your application includes an access control list, you can create scripts to create the underlying access control tables.

> **See Also:** "Controlling Access to Applications, Pages, and Page Components" in *Oracle Application Express Application Builder User's Guide*

To create scripts for access control tables:

1. On the Installation Scripts page, click **Create**.
2. Click the **Create Scripts for Access Control Tables** link.
3. If you had wanted to create installation scripts for access control tables, you would click Create Script.
4. For this exercise, click **Cancel**.

About Creating Scripts to Install Files

To include other types of objects, such as cascading style sheets, images, and static files that your application references, you need to create scripts that install the files as well as bundle the files as supporting objects.

Before you begin, make sure the files you want to include have been added as Shared Components.

> **See Also:** "Using Custom Cascading Style Sheets," "Managing Images," and "Managing Static Files" in *Oracle Application Express Application Builder User's Guide*

To create a script that installs files:

1. On the Installation Scripts page, click **Create**.
2. Click the **Create Scripts to Install Files** link.

   A list of file types appear. Each object is created by its own script, but you can select to create more than one at once.

   The *OEHR Sample Objects* application does not include any files to install.

If you had wanted to include a script you would select the appropriate scripts and click **Create Scripts**. The name of the script defaults to the name of the object being created, and the sequence is defaulted as well. You can alter these attributes after creation.

For each file that you install, the appropriate deinstallation statement is also written to the deinstallation script.

3. For this exercise, click **Cancel**.

Success Message

Upon completion of the supporting objects installation, either a success or failure message appears. Use these messages to give the user information they may need to continue.

For example, for a successful installation, you may want to provide instructions on running the application, such as built-in usernames and passwords. For a failed installation, you might want to tell users whom to contact or instruct them to deinstall the application.

To review the success or fail messages:

1. On the Installation Scripts page, click the **Messages** tab.
2. Scroll down to the **Post Installation** section to review the messages.

About Supporting Objects Deinstallation

The final part of Supporting Object Utility is the deinstallation process. Deinstalling enables a user who has installed your application to cleanly remove all objects created during the installation process.

On the Supporting Objects page, the Deinstallation section includes a Deinstallation script link.

Topics in this section include:

- View the Deinstallation Confirmation Message
- Review the Deinstallation Script
- View the Deinstallation Success Message

View the Deinstallation Confirmation Message

When a user initiates the deinstallation process, a deinstallation message appears.

To review the deinstallation confirmation message section:

1. On the Supporting Object Messages page, scroll down to **Deinstallation**.
2. Scroll down to the Deinstallation section.

   For the *OEHR Sample Objects* application, no deinstallation message is defined.

Review the Deinstallation Script

When a user selects to deinstall supporting objects, the deinstallation script runs. Unlike installation, only one script exists for deinstallation. This script should remove each object created during installation, including database objects, as well as any loaded files, cascading style sheets, or images.

To review the deinstallation script for *OEHR Sample Objects*:

1. On the Supporting Object Messages page, click the **Deinstall** tab.

   The Deinstall Script page appears.

2. Click the **Edit** icon to the left of the script.

   Notice that the drop table statements contain `CASCADE CONSTRAINTS` clauses. Using these clauses is the easiest way to avoid contention between foreign keys when dropping tables. If you create installation scripts for any files, the code to drop those is included in the deinstallation script for you.

3. For this exercise, click **Cancel**.

View the Deinstallation Success Message

The final part to the deinstallation process is the deinstallation success message. This is the message displayed after all the deinstallation scripts have been executed.

To review the deinstallation success message:

1. On the Deinstall Script page, click the **Messages** tab.
2. Scroll down to the Deinstallation section.

   For *OEHR Sample Objects*, note the post-deinstall message.

About Supporting Objects Export

After defining your supporting objects, you can decide whether or not to include them when you export your application. You set a default value for this export setting on the Export tab.

Topics in this section include:

- Review the Supporting Objects Export Setting
- Where the Export Default Appears When Exporting

Review the Supporting Objects Export Setting

To review the export setting:

1. On the Supporting Object Messages page, click the **Export** tab.

   Use the Export Status page to specify whether to include supporting objects in the export. For *OEHR Sample Objects*, note that Include Supporting Object Definitions in Export is set to **Yes**.

2. For this exercise, click **Cancel**.

Where the Export Default Appears When Exporting

Now that you know where to set the export default, next you learn where this setting appears in the export steps.

To see where the export setting default appears in the export steps:

1. Go to the Supporting Objects page by clicking the **Supporting Objects** breadcrumb.
2. From the Tasks list on the right, click **Export Application**.

   In the Export Application section, notice that Export Supporting Object Definitions is set to **Yes**. If this option is set to No, your export only included your application definition.
3. To return to the Supporting Object Installation page, click **Manage Supporting Objects** on the Tasks list on the right.

About Creating Upgrade Scripts

Creating upgrade scripts enables you to modify a distributed application, but enables users to retain their existing objects. You can use upgrade scripts to add database objects, alter existing database objects (for example, add columns or change column definitions), or create new files.

If you plan to include upgrade scripts, remember to also modify your installation scripts and deinstallation scripts. Then, when new users install your application, they will have the full set of supporting objects and if the upgraded application is removed, all the objects will also be removed.

The wizard that installs the supporting objects needs to be able to determine whether to run the installation scripts or the upgrade scripts. This decision point is determined by the Detect Existing Supporting Objects query. If the query returns at least one row then the upgrade scripts are run. If the query returns no rows, the installation scripts are run. A common approach is to have the query check the Oracle data dictionary for the existence of the tables that the application depends on as shown in the following example:

```
SELECT 1
  FROM user_tables
 WHERE table_name in ('OEHR_ORDERS','OEHR_ORDER_ITEMS')
```

If your upgrade consists only of application-level changes (that is, there are no supporting objects changes) you do not need to create upgrade scripts. You can simply instruct users to install the new application, but not install supporting objects.

Utilizing Upgrade Messages

When creating an upgrade, there are four messages you can use: Welcome Message, Confirmation Message, Success Message, and Failure Message. Use these messages to inform the user that they are upgrading their supporting objects instead of just installing them.

If you distribute an application that is used as an upgrade, you should recommend that users install the application using a new application ID. This approach retains the initial application and its associated supporting object definitions. Once the user runs and reviews the upgraded application, they can then delete the prior version. Remember to also make sure users understand that they should not remove supporting objects.

About Refining Your Installation Scripts

Once you create your supporting objects, you should test them. As a best practice, be sure to:

1. Perform the test in another workspace, or in the same workspace using a different schema.
2. Export your application and include the supporting object definitions.
3. Then, import the application into another workspace and experience the installation process.

   This will enable you to edit the messages to fit your application.
4. If the supporting objects are created successfully, you should run and test the application.
5. If the applications displays and performs as expected, delete the application and then deinstall your supporting objects.
6. Finally, use Object Browser to verify that all your supporting objects have been removed.

   Since installation scripts are rarely correct the first time, this is typically an iterative process.

Downloading Public Packaged Applications and Sample Code

If you are using Oracle Application Express 2.2 or higher, you can download packaged applications and sample code from the Oracle Application Express Web site. Packaged applications are fully functional applications that you can view, use, and customize. Sample code is provided as packaged applications that contain a snippet of code to explain a solution.

To download public packaged applications and sample code:

1. In a Web browser, go to the following Web site:

   `http://www.oracle.com/technology/products/database/application_express/packaged_apps/packaged_apps.html`
2. Scroll down to review the available applications or code.
3. Click the link for the zip file you want to download, and save the file to your computer.
4. Unzip the file.
5. Review the Readme file.

   Each application includes a Readme file that contains details about the application functionality, installation procedures, how to remove sample data (if appropriate), and default user names and passwords (if applicable).
6. Log in to the workspace in Oracle Application Express where you want to use the packaged application or code.
7. Import and install the application.

   Follow the same steps you performed when installing the sample objects for this guide. See "About Loading Sample Objects" on page 1-2.

13

How to Create a Master Detail PDF Report

Oracle Application Express supports the ability to print a report by exporting a report region to PDF. Defined declaratively, report printing enables users to view and print reports that include page headings and that properly conform to specified page sizes. When users print a report, the report data is transformed to a PDF format using an externally defined report server.

In addition to enabling printing for report regions, you can also define output using report queries and report layouts that are linked to an application.

This tutorial explains how to build a master detail form, define a report query and RTF template, and then create a button to expose the new report.

This section contains the following topics:

> **See Also:** *PDF Printing in Application Express 3.0* and "Configuring Printing for Reports" in *Oracle Application Express Application Builder User's Guide*

About Oracle BI Publisher Licensing Requirements

Advanced PDF Printing requires Oracle Application Express release 3.0 or higher and a valid license of Oracle BI Publisher. If your Oracle Application Express instance is not currently configured to use BI Publisher, you can learn more about installing and configuring PDF Printing in *PDF Printing in Application Express 3.0*.

Viewing a Hosted Version of the Application

If you do not have a valid Oracle BI Publisher license, or have not yet installed or configured Oracle BI Publisher, you can follow the exercises in this tutorial and view a completed version of the application on `apex.oracle.com`.

To view hosted version of the application:

1. In your Web browser go to:

   `http://apex.oracle.com/pls/otn/f?p=15610`

   The hosted application appears.
2. Click the **Edit** icon adjacent to a customer.
3. On the Order Details page, click **Print PDF of Order**.
4. Save the PDF to your hard drive so you can view it.

Loading the Required Objects

In order to complete this tutorial, you must download and install the packaged application *How To Create a Master-Detail PDF Report* from Oracle Technology Network (OTN). Importing and installing this packaged application creates the required objects needed to complete this tutorial. Additionally, it also contains the final version the application you will build. That way, you can refer to it as a code reference.

Topics in this section include:

- Download the Packaged Application
- Import and Install the Packaged Application

Download the Packaged Application

To download the packaged application from OTN:

1. Download the `md_pdf_howto.zip` file to your computer. In your Web browser go to:

   `http://www.oracle.com/technology/products/database/application_express/packaged_apps/md_pdf_howto.zip`

2. Unzip and extract the `md_pdf_howto.zip` file:
   - Microsoft Windows - Double-click the `md_pdf_howto.zip` file
   - UNIX or Linux - Enter the following command:

     ```
     $ unzip md_pdf_howto.zip
     ```

3. Review the `md_pdf_howto_readme.txt` file.

Import and Install the Packaged Application

To import and install the application, *How To Create a Master-Detail PDF Report*:

1. Log in to Oracle Application Express. See "Logging In To Oracle Application Express" in *Oracle Application Express Application Builder User's Guide*.
2. On the Workspace home page, click **Application Builder**.

   The Application Builder home page appears.
3. Click the **Import** button.
4. For Specify File, specify the following:
   a. Import file - Click **Browse** and go to the `md_pdf_howto_installer.sql` file.
   b. File Type - Select **Application, Page, or Component Export**.

c. Verify that File Character Set is correct.

d. Click **Next**.

5. For File Import Confirmation, click **Next** to install the imported file.

   The Install Application Wizard appears.

6. In the Install Application Wizard, specify the following:

   a. Parsing Schema - Select the appropriate schema.

   b. Build Status - Select **Run and Build Application**.

   c. Install As Application - Select **Auto Assign New Application ID**.

   d. Click **Install**.

7. For Supporting Objects, select **Yes** and click **Next**.

8. Confirm your selections by clicking **Install**.

9. Click the **Home** breadcrumb link at the top of the page.

   The Application Builder home page appears.

About Checking the Available Space in Your Workspace

If you experience problems installing the *How To Create a Master-Detail PDF Report* application, verify the available space in your workspace. You many need to request additional storage space.

If you are a workspace administrator, you can:

1. Determine if you need additional storage space. See "Viewing the Workspace Overview Report" in *Oracle Application Express Application Builder User's Guide*.
2. Request additional storage space. See "Requesting Additional Storage" in *Oracle Application Express Application Builder User's Guide*.

About Deleting the Packaged Application

Deleting the *How To Create a Master-Detail PDF Report* application and selecting to deinstall the supporting objects completely removes all associated objects and sample data.

To delete the *How To Create a Master-Detail PDF* Report application:

1. Log in to Oracle Application Express.
2. On the Workspace home page, click **Application Builder**.

   The Application Builder home page appears.

3. Select the *How To Create a Master-Detail PDF Report* application.

   The Application home page appears.

4. On the Tasks list, click **Delete this Application**.

   The Deinstall page appears.

5. To remove all associated objects and sample data, select **Remove Application Definition** and **Deinstall Supporting Objects**.
6. Click **Deinstall**.

Creating a Master Detail Form

To create a master detail form:

1. On the Workspace home page, click the **Application Builder** icon.
2. Select the *How To Create a Master-Detail PDF Report* application.
3. Click **Create Page**.
4. Select **Form** and click **Next**.
5. Select **Master Detail Form** and click **Next**.

   The Master Detail Wizard appears.
6. On Master Table, select the following:
   a. Table/View Owner - Select the appropriate schema.
   b. Table/View Name - Select `MD_PDF_ORDERS`.

      The columns in that object appear under Available Columns.
   c. Press the **Shift** key to select all the columns and then click the **Add** button to move them to Displayed Columns.
   d. Click **Next**.
7. On Detail Table, specify the following:
   a. Show Only Related Tables - Accept the default, **Yes**.
   b. Table/View Owner - Select the appropriate schema.
   c. Table/View Name - Select `MD_PDF_ORDER_ITEMS`.

      The columns in that object appear under Available Columns.
   d. Press the **Shift** key to select all the columns and then click the **Add** button to move them to Displayed Columns.
   e. Click **Next**.
8. On Primary Key Source:
   a. For the Primary Key Column `ORDER_ID`, accept the default, **Existing Trigger** and click **Next**.
   b. For the Primary Key Column `ORDER_ITEM_ID`, accept the default, **Existing Trigger** and click **Next**.
9. To accept the remaining defaults, click **Finish**.
10. Click **Create**.

    A Success page appears.

View the Report

To view the new pages:

1. Click the **Run Page** icon.

   The PDF Orders report appears as shown in Figure 13–1 on page 13-5.

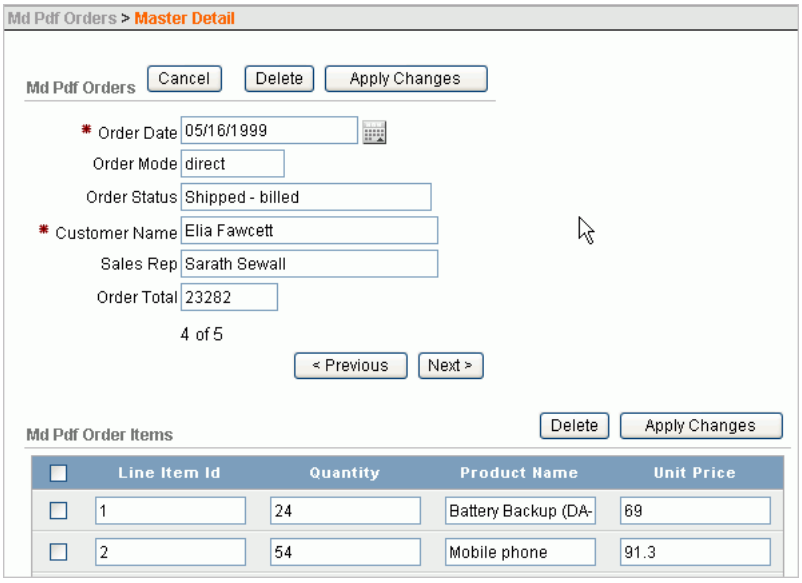
Figure 13–1 Md Pdf Orders Report

Note that the wizard added two pages to the application, a report on the master table and a master detail form that references both tables.

2. To edit an order and view the master detail form, click the **Edit** icon.

   The Master Details page appears as shown in Figure 13–2.

Figure 13–2 Master Details Page



Creating the Report Query

The form you created displays order items for five orders. Using standard PDF printing, you could enable region printing on the tabular form. This would result in an output that would display just the order items records and only for the selected master. To create a report that incorporates all the order information, along with the Order Items, you will create a report query. The order information will be retrieved from the session state for each order item.

To create a report query:

1. Go to the Report Queries page:
   a. To return to the Application home page, click the **Application** link on the Developer toolbar.
   b. Click on the **Shared Components** icon.
   c. Under Reports, click **Report Queries**.

      The Report Queries page appears.
2. Click **Create**.
3. For Report Query Definition, enter the following:
   a. Report Query Name - Enter `order_details`.
   b. Session State - Select **Include application and session information**.

      The Select Items fields appear.
   c. Select Items - Select **P3_CUSTOMER_NAME**, and click **Add**.

      Repeat this procedure for:

      P3_ORDER_DATE

      P3_ORDER_ID

      P3_ORDER_MODE

      P3_ORDER_STATUS

      P3_ORDER_TOTAL

      P3_SALES_REP
   d. Accept remaining defaults and click **Next**.
4. For Source Queries, enter the following:
   a. SQL Query - Enter the following code:

```
SELECT LINE_ITEM_ID,
       QUANTITY,
       ORDER_ITEM_ID,
       PRODUCT_NAME,
       to_char(UNIT_PRICE,'$9,999.99') unit_price
  FROM MD_PDF_ORDER_ITEMS
 WHERE ORDER_ID = :P3_ORDER_ID
```

   b. Click **Next**.
5. For Data Source for Report Layout, select **XML Data** and the click **Download** button.

   This step downloads an XML file with the name of the report query, `order_details.xml`.
6. Save the resulting file to your hard drive.

   You will create an RTF template using this XML file and then upload it. First, however, you must complete the process of creating the report query using a generic report layout. You will then edit the report query to reference the new report layout once it is created and uploaded.
7. Click **Next**.

8. For Report Layout File, browse to and select the `order_details.xml` file you saved.
9. Click **Next**.

   Notice the URL that displays. This is the URL that will be used to call this report from within your application.
10. Click **Finish**.

    Next, you need to create an RTF template using this XML file.

Creating the RTF Template

To edit the XML produced from your report query, you need the Oracle BI Publisher Desktop. When properly loaded, Oracle BI Publisher Desktop adds a new menu option to Microsoft Word.

Topics in this section include:

- Download Oracle BI Publisher Desktop
- Load the XML Details
- Insert the Fields
- Include Line Items

See Also: *Oracle BI Publisher Desktop White Papers and Demonstrations*

Download Oracle BI Publisher Desktop

If you do not have the Desktop loaded, you can download it here:

`http://www.oracle.com/technology/software/htdocs/devlic.html?url=/technology/software/products/ias/htdocs/101320bi.html`

See Also: "About Oracle BI Publisher Licensing Requirements" on page 13-1

Load the XML Details

To load the XML generated by your report query:

1. Open Microsoft Word.
2. From the Oracle BI Publisher menu, select **Data** and then **Load Sample XML Data**.
3. Select the file you generated when creating report query, `order_details.xml`.

   The following message appears:

   ```
   Data loaded successfully
   ```
4. Click **Ok**.

   Note that nothing appears on the page.

Insert the Fields

To insert the order columns:

1. From the Oracle BI Publisher menu, select **Insert** and then **Field**.

2. Select `P3 Customer Name` and click **Insert**.
3. Continue to insert each of the order columns.
4. Exit the Field dialog box, click **Close**.

   Although each of the order items displays on the page, they are in one long string.
5. Edit the page so that each item displays on a separate line and is prefaced by a descriptive label. Consider the following example:

```
Customer: P3_CUSTOMER_NAME
Order Date: P3_ORDER_DATE
Order ID: P3_ORDER_ID
Order Mode: P3_ORDER_MODE
Order Status: P3_ORDER_STATUS
Order Total: P3_ORDER_TOTAL
Sales Rep: P3_SALES_REP
```

Include Line Items

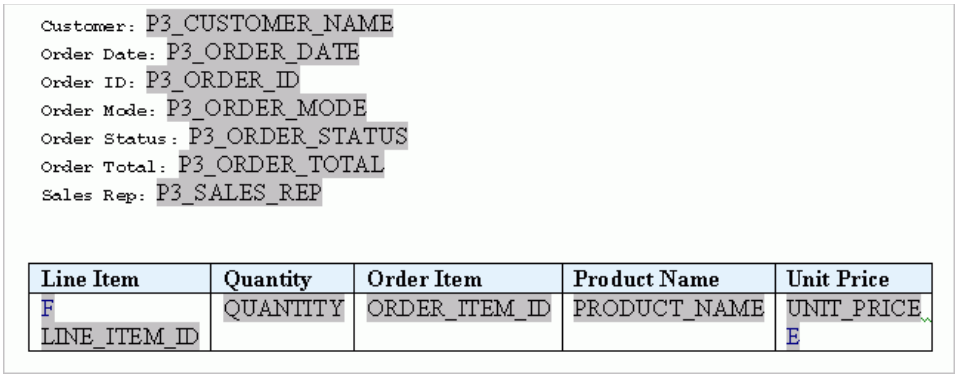
To include line items:

1. Insert a few blank lines below your order details.
2. From the Oracle BI Publisher menu, select **Insert** and then **Table Wizard**.
3. Accept the default, **Table**, and click **Next**.
4. For Grouping Field, select **DOCUMENT/REGION/ROWSET/ROW** and click **Next**.
5. Select each field and move it to the right column. Click **Next**.

   You do not need to make any grouping selections since your report will select just one order.
6. Click **Next**.
7. For Sort By, select **Line Item Id** and click **Next**.
8. For Labels, edit the following:
   a. Line Item Id - Change to `Line Item`.
   b. Order Item Id - Change to `Order Item`.
9. Click **Finish**.

   Your basic framework is created and should resemble Figure 13–3 on page 13-9.

Figure 13–3 Report Layout Template



10. To define a header and footer, from the View menu, select **Header and Footer**.

    For example, the header could contain the name of the report (for example, `My Order Report`) and the footer could contain page numbers or the date the report was executed. You can also customize the font family or font size.

11. Save your changes:

    a. From the File menu, select **Save As**.

    b. For Save as type, select **Rich Text Format (*.rtf)**.

    c. For File Name, enter `order_details.rtf`.

    d. Exit Microsoft Word.

About Including Variables in Your RTF

You can include variables in the header and footer of your RTF. For example, the RTF file used in the sample pages included in the *How To Create a Master-Detail PDF Report* application include the date the report was executed and the user that executed the report.

To download and view this RTF:

1. Go to the Report Layouts page:

    a. Click on the **Shared Components** icon.

    b. Under Reports, click **Report Layouts**.

    The Report Queries page appears.

2. Click **order_details2**.

3. Click **Download** and save `order_details2.rtf` to your hard drive.

4. Open `order_details2.rtf` in Microsoft Word and note the variables included in the footer.

Creating the Report Layout

Once you complete your report layout template, you need to upload it to Application Builder and associate it with your report query.

Topics in this section include:

- Create a Report Layout
- Associate the Report Layout and Report Query

Create a Report Layout

To create a report layout:

1. Go to the Shared Components page:
   a. Click the **Shared Components** breadcrumb.
   b. Under Reports, click **Report Layouts**.

      The Report Layouts page appears.
2. Click **Create**.
3. For Layout Type, select **Named Columns** and click **Next**.
4. For Layout Type, specify the following:
   a. For Layout Name, enter `order details`.
   b. For Report Layout File, click **Browse** and select **order_details.rtf**.
5. Click **Create Layout**.

Associate the Report Layout and Report Query

To associate the report layout and report query:

1. Go to the Shared Components page. Click the **Shared Components** breadcrumb.
2. Under Reports, click **Report Queries**.
3. Edit the query by clicking **order_details**.
4. Under Report Query Attributes, for Report Layout, select **order details**.
5. Click **Apply Changes**.

Linking the PDF Report to the Application

In this exercise, you link the PDF report to the application by creating a button.

Topics in this section include:

- Create a Button
- Run the Page

Create a Button

To create a button:

1. Go to the Page Definition for the Master Detail page:
   a. Click the Application breadcrumb link.
   b. Select **Master Detail**.
2. Under Buttons, click the **Create** icon.

Figure 13–4 Create Icon

3. For Button Region, select **Md Pdf Orders** and click **Next**.

   This selection positions the button in the top (or master) region.

4. For Button Position, select accept the default, **Create a button in a region position**, and click **Next**.

5. For Button Attributes:

   a. Button Name - Enter `PRINT`.

   b. Label - Enter `Print PDF of Order`.

   c. For Button Type, accept the default, **Template Driven**.

   d. For Action, select **Download printable report query**.

   e. Click **Next**.

6. For Button Template, accept the default and click **Next**.

7. For Display Properties:

   a. Position - Select **Region Template Position #CREATE#**.

      This positions the button to the right of the Apply Changes button.

   b. Accept the remaining defaults and click **Next**.

8. For Branching:

   a. Report Query - Select `order_details`.

   b. Click **Next**.

9. For Conditional Display:

   a. Condition Type - Select **Value of Item in Express 1 Is NOT NULL**.

   b. Expression 1 - Enter `P3_ORDER_ID`.

      By creating this condition, the print button will not display when you are creating a new order.

10. Click **Create Button**.

Run the Page

To run the page:

1. Click the **Run Page** icon in the upper right corner as shown in Figure 13–5.

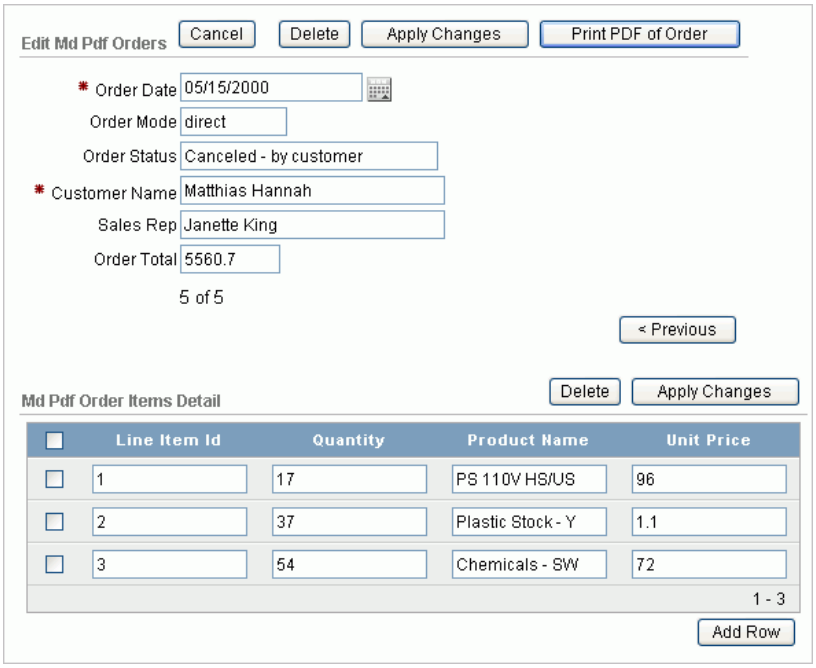
Figure 13–5 Run Page Icon

2. When the report appears, click **Cancel** to return to the Md Pdf Order page.

3. Click the **Edit** icon to select another order.

   Your Master Detail page should resemble Figure 13–6. Note the new Print PDF or Order button.

Figure 13–6 Revised Master Detail Page



4. If you do not have an Oracle BI Publisher license, or have not yet installed or configured Oracle BI Publisher, you will not be able to save the report as a PDF.

   To view a fully functional version of this application:

   - View the hosted version on `apex.oracle.com`. See "Viewing a Hosted Version of the Application" on page 13-1.
   - View the example pages included with the demonstration application. Return to the Application home page, run the Orders page, click **Edit** to edit an order, and click **Print PDF Order**.

About Creating One PDF Containing All Orders

This tutorial created a report that was specific to one order. You could also create the same report without passing in the Order ID. This would create one PDF that would contain all orders. To accomplish this, you would:

1. Create one report query that returns all the required columns. There would be no need to include session state variables.
2. Load the resulting XML into Microsoft Word just as you did in this tutorial and use the Table Wizard.
3. When prompted to specify Groupings, select the columns that you want to use as your master.

   The resulting display will include the grouped columns above a table that contains your detail records. You can then add labels and customize the display. When the report is run, each master value will display on a new page.

14

How to Build and Deploy an Issue Tracking Application

This tutorial describes how to create and deploy an application that tracks the assignment, status, and progress of issues related to a project. This tutorial walks you through all the steps necessary to create a robust issue tracking application, including planning the project, creating the underlying database objects, loading demonstration data, and building a rich user interface.

> **Note:** This tutorial takes approximately four to five hours to complete. It is recommended that you read through the entire document first to become familiar with the material before you attempt specific exercises.

Topics in this section include:

Planning and Project Analysis

Effective project management is the key to completing any project on time and within budget. Within every project there are always multiple issues that need to be tracked, prioritized, and managed.

In this business scenario, MRVL Company has several projects that must be completed on time for the company to be profitable. Any missed project deadlines will result in lost revenue. The company's project leads use various methods to track issues, including manually recording statuses in notebooks, organizing issues in text documents, and categorizing issues by using spreadsheets.

By creating a hosted application in Oracle Application Express, project leads can easily record and track issues in one central location. This approach offers each project lead access to just the data they need and makes it easier for management to determine if critical issues are being addressed.

Planning and Project Analysis

Before beginning development on an Oracle Application Express application, you first need to define application requirements. Then, you use the defined requirements to design a database and an outline that describes how the user interface accepts and presents data.

For this business scenario, the project leads establish requirements that define the information that must be tracked, security requirements, data management functions, and how to present data to users.

Topics in this section include:

Gather the Necessary Data

Currently, each project lead tracks information slightly differently. Together, everyone agrees that the application should include the following information:

- Summary of the issue
- Detailed description of the issue
- Who identified the issue
- The date on which the issue was identified
- Which project the issue is related to
- Who the issue is assigned to
- A current status of the issue
- Priority of the issue
- Target resolution date
- Actual resolution date
- Progress report
- Resolution summary

Define Security Requirements

Because the project leads are concerned about everyone having access to all the information, they agree upon the following access rules:

- Each team member and project lead is only assigned to one project at a time
- Each team member and project lead must be assigned to a project
- Managers are never assigned to a specific project
- Only managers can define and maintain projects and people
- Everyone can enter new issues
- Once assigned, only the person assigned or a project lead can change data about the issue

- Management needs views that summarize the data without access to specific issue details

Select Data Management Functions

Next, the project leads determine how information will be entered into the system. For this project, users must be able to:

- Create issues
- Assign issues
- Edit issues
- Create projects
- Maintain projects
- Create people
- Maintain people information
- Maintain project assignments

Select Data Presentation Functions

Once the data is entered into the application, users need to view the data. The team decides that users must be able to view the following:

- All issues by project
- Open issues by project
- Overdue issues, by project and for all
- Recently opened issues
- Unassigned issues
- Summary of issues by project, for managers
- Resolved issues by month identified
- Issue resolution dates displayed on a calendar
- Days to Resolve Issues by person

Define Special Function Requirements

Finally, the project leads determine that the application must support the following special functions:

- Notify people when an issue is assigned to them
- Notify the project lead when any issue becomes overdue

Designing the Database Objects

Once you have defined the database requirements, the next step is to turn these requirements into a database design and an outline that describes how the user interface accepts and presents data. In this step you need to think about how information should be organized in the tables in the underlying database. Given the requirements described "Planning and Project Analysis" on page 14-2, for this project you need to create three tables:

- `Projects` tracks all current projects

- `People` contains information about who can be assigned to handle issues
- `Issues` tracks all information about an issue, including the project to which it is related and the person assigned to the issue

In addition to the tables, you also need to create additional database objects, such as sequences and triggers, to support the tables. System generated primary keys will be used for all tables so that all the data can be edited without executing a cascade update.

Topics in this section include:

- About the Projects Table
- About the People Table
- About the Issues Table

About the Projects Table

Each project must include project name, project start date, target date, and actual end date columns. These date columns help determine if any outstanding issues are jeopardizing the project end date. Table 14–1 describes the columns to be included in the Projects table.

Table 14–1 Project Table Details

Column Name	Type	Size	Not Null?	Constraints	Description
project_id	integer	n/a	Yes	Primary key	A unique numeric identification for each project. Populated by a sequence using a trigger.
project_name	varchar2	100	Yes	Unique key	A unique alphanumeric name for the project.
start_date	date	n/a	Yes	None	The project start date.
target_end_date	date	n/a	Yes	None	The targeted project end date.
actual_end_date	date	n/a	No	None	The actual end date.

About the People Table

Each person will have a defined name and role. Project leads and team members will also have an assigned project. To tie the current user to their role within the organization, email addresses will be used for user names.

Table 14–2 on page 14-4 describes the columns that will be included in the People table.

Table 14–2 People Table Details

Column Name	Type	Size	Not Null?	Constraints	Description
person_id	integer	n/a	Yes	Primary key	A numeric ID that identifies each user. Populated by a sequence using a trigger.

Table 14–2 (Cont.) People Table Details

Column Name	Type	Size	Not Null?	Constraints	Description
person_name	varchar2	100	Yes	Unique key	A unique name that identifies each user.
person_email	varchar2	100	Yes	None	User email address.
person_role	varchar2	7	Yes	Check constraint	The role assigned to each user.

Note: For the purposes of this exercise, this application has been simplified. User data is usually much more elaborate and is often pulled from a corporate Human Resource system. Also, users typically work on more than one project at a time. If the roles that are assigned to a user need to be dynamic, you would implement roles as a separate table with a foreign key that relates to the people table.

About the Issues Table

When the project leads defined their application requirements, they decided to track separate issues assigned to each person. Issues will be included in columns along with additional columns to provide an audit trail. The audit trail will track who created the issue, when it was created, as well as who modified the issue last and on what date that modification was made.

Table 14–3 describes the columns to be included in the Issues table.

Table 14–3 Issue Table Details

Column Name	Type	Size	Not Null?	Constraints	Description
issue_id	integer	n/a	Yes	primary key	A unique numeric ID that identifies an issue. Populated by a sequence using a trigger.
issue_summary	varchar2	200	Yes	None	A brief summary of the issue.
issue_description	varchar2	2000	No	None	A detailed description of the issue.
identified_by	integer	n/a	Yes	foreign key to People	The user who identifies the issue.
identified_date	date	n/a	Yes	None	The date the issue was identified
related_project	integer	n/a	Yes	foreign key to Projects	Projects related to the issue.
assigned_to	integer	n/a	No	foreign key to People	The person who owns this issue.
status	varchar2	8	Yes	check constraint	The issue status. Automatically set to `Open` when new and set to `Closed` when actual resolution date entered.
priority	varchar2	6	No	check constraint	The priority of the issue.

Table 14–3 (Cont.) Issue Table Details

Column Name	Type	Size	Not Null?	Constraints	Description
target_resolution_date	date	n/a	No	None	The target resolution date.
progress	varchar2	2000	No	None	The progress of the issue.
actual_resolution_date	date	n/a	No	None	Actual resolution date of the issue.
resolution_summary	varchar2	2000	No	None	Resolution summary.
created_date	date	n/a	Yes	None	Populated by a trigger.
created_by	varchar2	60	Yes	None	User who created this issue.
last_modified_date	date	n/a	No	None	Populated by a trigger.

Note: A real-world application might need more extensive auditing. For example, you might need to track each change to the data rather than just the last change. Tracking each change to the data would require an additional table, linked to the issues table. If the valid priorities assigned to issues need to be dynamic, you would be required to add a separate table with a foreign key that relates to the issues table.

Implementing Database Objects

This first step in building an application is to create the database objects.

Topics in this section include:

- About Building Database Objects
- About Building Database Objects
- View the Created Database Objects

About Building Database Objects

There are several ways to create objects in Oracle Application Express. You can:

- **Create an Object in Object Browser**. Use Object Browser to create tables, views, indexes, sequences, types, packages, procedures, functions, triggers database links, materialized views, and synonyms. A wizard walks you through the choices necessary to create the selected database object. To create an object in Object Browser, navigate to SQL Workshop, then Object Browser, and click **Create**. See "Managing Database Objects with Object Browser" in *Oracle Application Express Application Builder User's Guide*.
- **Execute SQL Commands.** Run SQL Commands by typing or pasting them into the SQL Commands. To access SQL Commands, click the **SQL Workshop** icon on Workspace home page and then click **SQL Commands**. See "Using SQL Commands" in *Oracle Application Express Application Builder User's Guide*.
- **Upload a script.** Upload a script to the SQL Script Repository that contains all the necessary create object statements. To upload a script, click **SQL Workshop** on the Workspace home page, click **SQL Scripts** and then click **Upload**. See "Uploading a SQL Script" in *Oracle Application Express Application Builder User's Guide*.

- **Create script online.** Create a script online in the Script Repository. You will use this method to create database objects for this exercise. To create a script online, click the **SQL Workshop** icon on the Workspace home page, select **SQL Scripts** and then click **Create**. See "Creating a SQL Script in the Script Editor" in *Oracle Application Express Application Builder User's Guide*.

For this exercise, you create and run a script.

Create and Run a Script

To build database objects by creating a script:

1. Log in to Oracle Application Express.
2. On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.
3. Click **Create**.
4. In the Script Editor:
   a. For Script Name, enter `DDL for Issue Management Application`.
   b. Copy the data definition language (DDL) in "Creating Application Database Objects DDL" on page A-1 and paste it into the script.
   c. Click **Save**.
5. On the SQL Scripts page, click the **DDL for Issue Management Application** icon.

   The Script Editor appears.
6. Click **Run**.

   A summary page appears.
7. Click **Run** again.

   The Manage Script Results page displays a message that the script has been submitted for execution.

View the Created Database Objects

You can view database objects using Object Browser.

To view database objects in Object Browser:

1. Return to the Workspace home page. Click the **Home** breadcrumb link.
2. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.
3. From the Object list on the left side of the page, select **Tables**.
4. To view the details of a specific object, select one of the following tables:
   - `HT_ISSUES`
   - `HT_PEOPLE`
   - `HT_PROJECTS`

   See Also: "Managing Database Objects with Object Browser" in *Oracle Application Express Application Builder User's Guide*.

Loading Demonstration Data

Once you have created all the necessary database objects, the next step is to load data into the tables. You can manually load data using the import functionality available in SQL Scripts. In the following exercise, however, you use SQL Scripts to load demonstration data.

Look at the DDL you copied from "Creating Application Database Objects DDL" on page A-1. Notice that the sequences used for the primary keys start at 40 in order to leave room for the demonstration data. The `BEFORE INSERT` triggers are coded so that the sequence is only accessed if a primary key value is not provided: they will not need to be disabled in order for you to load data.

Topics in this section include:

- Load Projects Data
- Load People Data
- Load Issues Data

Load Projects Data

To import data into the Projects table:

1. Click the **SQL Workshop** breadcrumb link.
2. Click **SQL Scripts**.
3. Click **Create**.
4. In the Script Editor, specify the following:
   a. Script Name - Enter `Load Project Data`.
   b. Script - Copy and paste the following:

```
INSERT INTO ht_projects
      (project_id, project_name, start_date, target_end_date)
   VALUES
      (1, 'Internal Infrastructure', sysdate-150, sysdate-30)
/
INSERT INTO ht_projects
      (project_id, project_name, start_date, target_end_date)
   VALUES
      (2, 'New Payroll Rollout', sysdate-150, sysdate+15)
/
INSERT INTO ht_projects
      (project_id, project_name, start_date, target_end_date)
   VALUES
      (3, 'Email Integration', sysdate-120, sysdate-60)
/
INSERT INTO ht_projects
      (project_id, project_name, start_date, target_end_date)
   VALUES
      (4, 'Public Website Operational', sysdate-60, sysdate+30)
/
INSERT INTO ht_projects
      (project_id, project_name, start_date, target_end_date)
   VALUES
      (5, 'Employee Satisfaction Survey', sysdate-30, sysdate+60)
/
```

   c. Click **Save**.

5. On the SQL Scripts page, click the **Load Project Data** icon.

   The Script Editor appears.

6. Click **Run**.

   A summary page appears.

7. Click **Run** again.

   The Manage Script Results page displays a message that the script has been submitted for execution.

Update Dates to Make the Projects Current

Although you have created the projects, the dates need to be updated to make the projects current. To accomplish this, you run another script.

To update the project dates and make the projects current:

1. Click the **SQL Scripts** breadcrumb link.
2. Click **Create**.
3. In the Script Editor:
   a. Script Name - Enter `Update Project Dates`.
   b. In Script, copy and paste the following:

```
UPDATE ht_projects
   SET start_date = sysdate-150,
       target_end_date = sysdate-30
 WHERE project_id = 1
/

UPDATE ht_projects
   SET start_date = sysdate-150,
       target_end_date = sysdate+15
 WHERE project_id = 2
/
UPDATE ht_projects
   SET start_date = sysdate-120,
       target_end_date = sysdate-60
 WHERE project_id = 3
/
UPDATE ht_projects
   SET start_date = sysdate-60,
       target_end_date = sysdate+30
 WHERE project_id = 4
/
UPDATE ht_projects
   SET start_date = sysdate-30,
       target_end_date = sysdate+60
 WHERE project_id = 5
/
```

   c. Click **Save**.

4. On the SQL Scripts page, click the **Update Project Dates** icon.

   The Script Editor appears.

5. Click **Run**.
6. Click **Run** again.

The Manage Script Results page displays a message that the script has been submitted for execution.

Load People Data

After you have loaded data into the Project table, you can load People data. Because of foreign keys in the Projects table, People data must be loaded after Project data. You load data into the People table by creating and running a script in SQL Workshop.

To load data into the People table:

1. Click the **SQL Scripts** breadcrumb link.
2. On the SQL Scripts page, click **Create**.
3. In the Script Editor:
   a. Script Name - Enter `Load People Data`.
   b. Script - Copy and paste the following:

```
INSERT INTO ht_people
     (person_id, person_name, person_email, person_role, assigned_project)
  VALUES
     (1, 'Joe Cerno', 'joe.cerno@mrvl-bademail.com', 'CEO', null)
/
INSERT INTO ht_people
     (person_id, person_name, person_email, person_role, assigned_project)
  VALUES
     (2, 'Kim Roberts', 'kim.roberts@mrvl-bademail.com', 'Manager', null)
/
INSERT INTO ht_people
     (person_id, person_name, person_email, person_role, assigned_project)
  VALUES
     (3, 'Tom Suess', 'tom.suess@mrvl-bademail.com', 'Manager', null)
/
INSERT INTO ht_people
     (person_id, person_name, person_email, person_role, assigned_project)
  VALUES
     (4, 'Al Bines', 'al.bines@mrvl-bademail.com', 'Lead', 1)
/
INSERT INTO ht_people
     (person_id, person_name, person_email, person_role, assigned_project)
  VALUES
     (5, 'Carla Downing', 'carla.downing@mrvl-bademail.com', 'Lead', 2)
/
INSERT INTO ht_people
     (person_id, person_name, person_email, person_role, assigned_project)
  VALUES
     (6, 'Evan Fanner', 'evan.fanner@mrvl-bademail.com', 'Lead', 3)
/
INSERT INTO ht_people
     (person_id, person_name, person_email, person_role, assigned_project)
  values
     (7, 'George Hurst', 'george.hurst@mrvl-bademail.com', 'Lead', 4)
/
INSERT INTO ht_people
     (person_id, person_name, person_email, person_role, assigned_project)
VALUES
     (8, 'Irene Jones', 'irene.jones@mrvl-bademail.com', 'Lead', 5)
/
INSERT INTO ht_people
```

```
     (person_id, person_name, person_email, person_role, assigned_project)
  VALUES
     (9, 'Karen London', 'karen.london@mrvl-bademail.com', 'Member', 1)
/
INSERT INTO ht_people
     (person_id, person_name, person_email, person_role, assigned_project)
   values
     (10, 'Mark Nile', 'mark.nile@mrvl-bademail.com', 'Member', 1)
/
INSERT INTO ht_people
     (person_id, person_name, person_email, person_role, assigned_project)
  VALUES
     (11, 'Jane Kerry', 'jane.kerry@mrvl-bademail.com', 'Member', 5)
/
INSERT INTO ht_people
     (person_id, person_name, person_email, person_role, assigned_project)
  VALUES
     (12, 'Olive Pope', 'olive.pope@mrvl-bademail.com', 'Member', 2)
/
INSERT INTO ht_people
     (person_id, person_name, person_email, person_role, assigned_project)
  VALUES
     (13, 'Russ Sanders', 'russ.sanders@mrvl-bademail.com', 'Member', 3)
/
INSERT INTO ht_people
     (person_id, person_name, person_email, person_role, assigned_project)
  VALUES
     (14, 'Tucker Uberton', 'tucker.uberton@mrvl-bademail.com', 'Member',
3)
/
INSERT INTO ht_people
      (person_id, person_name, person_email, person_role, assigned_project)
  VALUES
      (15, 'Vicky Williams', 'vicky.willaims@mrvl-bademail.com', 'Member',
4)
/
INSERT INTO ht_people
      (person_id, person_name, person_email, person_role, assigned_project)
  VALUES
      (16, 'Scott Tiger', 'scott.tiger@mrvl-bademail.com', 'Member', 4)
/
INSERT INTO ht_people
      (person_id, person_name, person_email, person_role, assigned_project)
  VALUES
      (17, 'Yvonne Zeiring', 'yvonee.zeiring@mrvl-bademail.com', 'Member',
4)
/
```

c. Click **Save**.

4. On the SQL Scripts page, click the **Load People Data** icon.

   The Script Editor appears.

5. Click **Run**.

6. Click **Run** again.

   The Manage Script Results page displays a message that the script has been submitted for execution.

Load Issues Data

The last data you need to load is the Issues data. As with People data, you create and run a script to populate the `Issues` table.

To load data into the `Issues` table:

1. Click the **SQL Scripts** breadcrumb link.
2. Click **Create**.
3. In the Script Editor:
   a. Script Name - Enter `Load Issue Data`.
   b. Script - Copy and paste the script in "Creating Issues Script" on page A-6.
   c. Click **Save**.
4. On the SQL Scripts page, click the **Load Issue Data** icon.

   The Script Editor appears.
5. Click **Run**.
6. Click **Run** again.

   The Manage Script Results page displays a message that the script has been submitted for execution.

Building a Basic User Interface

After you create the objects that support your application and load the demonstration data, the next step is to create a user interface. In this exercise, you use the Create Application Wizard in Application Builder to create an application and then the pages that support the data management and data presentation functions described in "Planning and Project Analysis" on page 14-2.

Topics in this section include:

- Create the Application
- Add Pages to Maintain Projects
- Add Pages to Track People
- Add Pages to Track Issues
- Create Summary Reports
- Add Content to the Home Page
- Add a Breadcrumb Menu

Create the Application

You use the Create Application Wizard to create an application containing pages that enable users to view reports on and create data for the selected tables within a schema. Alternatively, you can create an application first and then add pages to it. As the application requirements include customized overview pages, for this exercise you will use the latter approach.

To create the application:

1. Click the **Home** breadcrumb link.
2. On the Workspace home page, click **Application Builder**.

3. Click **Create**.
4. For Method, select **Create Application** and then click **Next**.
5. For Name:
   a. Name - Enter `Issue Tracker`.
   b. Create Application - Select **From scratch**.
   c. Click **Next**.
6. Next, add a blank page. Under Add Page:
   a. Under Select Page Type, select **Blank**.
   b. Click **Add Page**.
   c. Click **Next**.
7. For Tabs, select **No Tabs** and then click **Next**.
8. For Shared Components, accept the default, **No**, and click **Next**.
9. For Attributes, accept the defaults for Authentication Scheme, Language, and User Language Preference Derived From and click **Next**.
10. For User Interface, select **Theme 10** and click **Next**.
11. Click **Create**.

To view the application:

1. Click the **Run Application** icon on the Applications home page.
2. When prompted, enter your workspace user name and password and then click **Login**. See "About Application Authentication" on page 1-6.

   This authentication is part of the default security of any newly created application. As shown in Figure 14–1 on page 14-13, the home page appears.

Figure 14–1 Issue Tracking Application Home Page

Although the page has no content, notice that the Create Application Wizard has created the following items:

- **Navigation Links** - A navigation bar entry displays in the upper right of the page. Logout enables the user to log out of the application.

- **Developer Links** - The Developer toolbar appears on the page. These links only display if you are logged in as a developer. Users who only have access to run the application cannot see these links.

3. Click **Application** on the Developer toolbar to return to the Application home page.

   Notice that the Create Application Wizard also created a Login page.

Once you have created the basic application structure, the next step is to create individual pages.

Add Pages to Maintain Projects

First, you need to create pages that enable users to view and add data to tables. To accomplish this, you use the Form on a Table with Report Wizard. This wizard creates a report page and maintenance page for each table.

Topics in this section include:

- Create Pages for Maintaining Projects
- Refine the Appearance of the Projects Report Page
- Refine the Create/Edit Project Page

Create Pages for Maintaining Projects

To create pages for maintaining the `HT_PROJECTS` table:

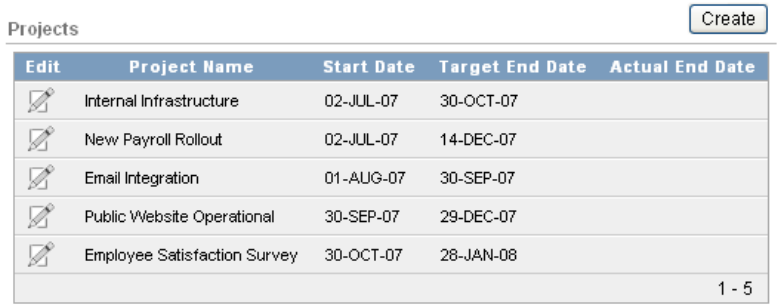
1. On the Application home page, click **Create Page**.
2. Select **Form** and then click **Next**.
3. Select **Form on a Table with Report** and click **Next**.
4. For Table/View Owner, select the appropriate schema and then click **Next**.
5. For Table/View Name, select `HT_PROJECTS` and click **Next**.
6. For Define Report Page:
   a. Page Number: Enter `2`.
   b. Page Name and Region Title - Enter `Projects`.
   c. Accept the remaining defaults and click **Next**.
7. For Tab Options, accept the default, **Do not use tabs**, and then click **Next**.
8. For Select Column(s), select all columns except `PROJECT_ID` and then click **Next**.

   Note that Project Name is unique and identifies the project. The ID was added to simplify the foreign key and enable cascading updates.
9. For Edit Link Image, select the fourth option (the word Edit) and then click **Next**.
10. For Define Form Page:
    a. Page - Enter `3`.
    b. Page Name and Region Title - Enter `Create/Edit Project`.
    c. Click **Next**.
11. For Tab Options, accept the default, **Do not use tabs**, and click **Next**.
12. For Primary Key, accept the default, **PROJECT_ID** and click **Next**.

13. For Source Type, accept the default, **Existing Trigger**, and click **Next**.
14. For Select Column(s), select all columns and click **Next**.
15. Under Identify Process Options, accept the defaults for **Insert**, **Update** and **Delete**, and click **Next**.
16. Review your selections and click **Finish**.
17. Click the **Run Page** icon.

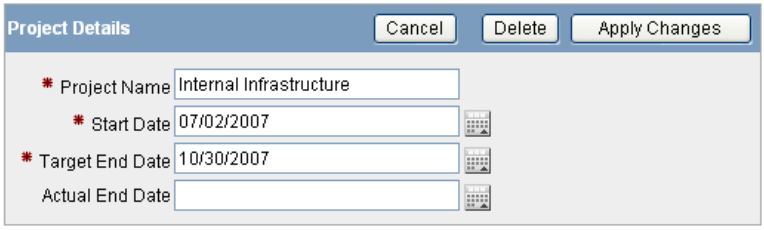
As shown in Figure 14–2, the newly created report displays the demo data.

Figure 14–2 Projects Page



Click the **Edit** link to view an existing row or click the **Create** button to create a new record. If you click **Edit** to the left of Employee Satisfaction Survey, a form resembling Figure 14–3 appears.

Figure 14–3 Create/Edit Project Form



Refine the Appearance of the Projects Report Page

You can change the appearance of the Projects report page by adding a format mask to the dates.

To add a format mask to the dates on the Create/Edit Project page:

1. Go to the Page Definition for page 2, Projects:
   a. Click **Application** on the Developer toolbar.
   b. On the Application home page, click **2 - Projects**.
2. Under Regions, click **Report** next to Projects.
3. Edit the format for `START_DATE`:
   a. Click the **Edit** icon to the left of `START_DATE`.

      The Column Attributes page appears.

b. Under Column Formatting, for Number/Date Format, enter `DD-MON-YYYY`.

4. Edit the format for the `TARGET_END_DATE`:
   a. Click the Next button (>) at the top of the page to go to the next Report Item.

      The Column Attributes page appears.
   b. Under Column Formatting, for Number/Date Format, enter `DD-MON-YYYY`.
5. Edit the format for the `ACTUAL_END_DATE`:
   a. Click the Next button (>) at the top of the page to go to the next Report Item.

      The Column Attributes page appears.
   b. Under Column Formatting, for Number/Date Format, enter `DD-MON-YYYY`.
6. Click **Apply Changes**.

   The Report Attributes page appears.
7. For `PROJECT_ID`, delete the Heading **Edit**.
8. For the `START_DATE`, `TARGET_END_DATE` and `ACTUAL_END_DATE` columns, select **center** for Column Alignment and Heading Alignment.

   Next, enable users to sort on column headings.
9. To enable column heading sorting, check **Sort** for all columns except `PROJECT_ID`.
10. For `PROJECT_NAME`, select **1** for Sort Sequence.

    This selection specifies `PROJECT_NAME` as the default column to sort on. Note this functionality can be overridden by any user selections.
11. Scroll down to Sorting. For Ascending and Descending Image, select the light gray arrow.
12. Scroll down to Messages. In When No Data Found Message, enter the following:

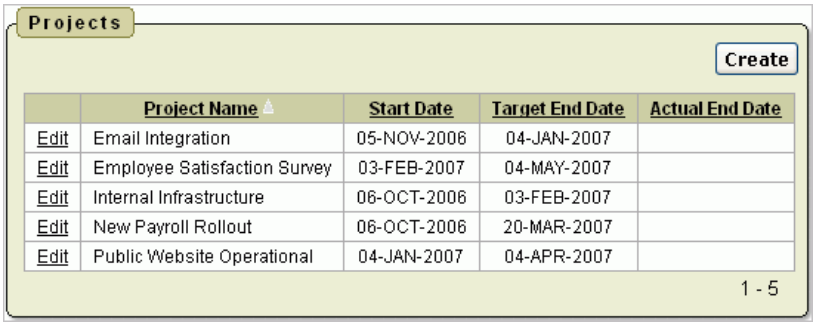
    ```
    No Projects found.
    ```

13. At the top of the page, click **Apply Changes**.

To view your changes, click the **Run Page** icon in the upper right of the page.

As shown in Figure 14–4, note the addition of a sort control on the Project Name column and the format of the dates in the Start Date and Target End Date columns.

Figure 14–4 Projects Page with Sort Control



Refine the Create/Edit Project Page

Next, you need to customize the Create/Edit Project page to make the Project Name field larger and the date fields smaller. You also need to change the date picker type, add a format mask for dates, and add validations that check if the target and actual end dates are after the start date.

Edit Fields To make the Project Name field larger and the date fields smaller:

1. Go to the Page Definition for Page 3, Create/Edit Project:
   a. From the Developer toolbar, click **Application**.
   b. Click **3 - Create/Edit Project**.
2. Under Items, click the **Edit All** icon.

   The Edit All icon resembles a small grid with a pencil on top of it.
3. Scroll to the right and locate the **Width** column:
   a. For Project Name, enter `60`.
   b. For Start Date, enter `12`.
   c. For Target End Date, enter `12`.
   d. For Actual End Date, enter `12`.
   e. Click **Apply Changes**.
4. Return to the Page Definition. Click the **Edit Page** icon in the upper right corner. The Edit Page icon resembles a small green piece of paper and pencil.

Change the Date Picker Type To change the date picker type and add a format mask for dates:

1. Edit the item `P3_START_DATE`.
   a. Under Items, click **P3_START_DATE**.
   b. From the Display As list in the Name section, select **Date Picker (DD-MON-YYYY)**.
   c. Click **Apply Changes**.
2. Edit the item `P3_TARGET_END_DATE`.
   a. Under Items, select **P3_TARGET_END_DATE**.
   b. From the Display As list in the Name section, select **Date Picker (DD-MON-YYYY)**.
   c. Click **Apply Changes**.
3. Edit the item `P3_ACTUAL_END_DATE`.
   a. Under Items, select **P3_ACTUAL_END_DATE**.
   b. From the Display As list in the Name section, select **Date Picker (DD-MON-YYYY)**.
   c. Click **Apply Changes**.

Add a Validation Next, add validations to check if the target and actual end dates are after the start date.

To add validations:

1. Under Page Processing, Validations, click the **Create** icon.
2. For Level, accept the default **Item level validation** and click **Next**.
3. For Item, select **Create/Edit Project: 40. P3_TARGET_END_DATE (Target End Date)** and click **Next**.
4. For Validation Method, select **PL/SQL** and click **Next**.
5. Specify the type of validation you want to create. Accept the default **PL/SQL Expression** and click **Next**.
6. For Validation Name, enter `TARGET_AFTER_START` and click **Next**.
7. For Validation:
   a. Validation - Enter:

      ```
      to_date(:P3_ACTUAL_END_DATE,'DD-MON-YYYY') >= to_date(:P3_START_
      DATE,'DD-MON-YYYY')
      ```

   b. Error Message - Enter:

      ```
      Actual End Date must be same or after Start Date.
      ```

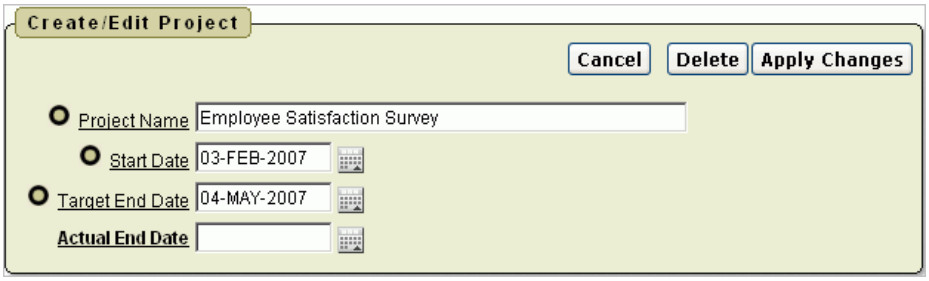
   c. Click **Next**.
8. For Conditions:
   a. **Condition Type** - Select **Value of Item in Expression 1 Is NOT NULL**, or click the shortcut link **[item not null]**.
   b. Expression 1 - Enter:

      ```
      P3_ACTUAL_END_DATE.
      ```

      This selection ensures that this validation executes only if the user enters an Actual End Date.
   c. Click **Create.**

To view your changes, click the **Run Page** icon in the upper right of the page. (See Figure 14–5.)

Figure 14–5 Modified Create/Edit Project



Add Pages to Track People

Once the initial Projects pages are complete, you create pages for maintaining people.

Topics in this section include:

- Create Pages for Maintaining People
- Modify the People Report Page

- Refine the Create/Edit People Page

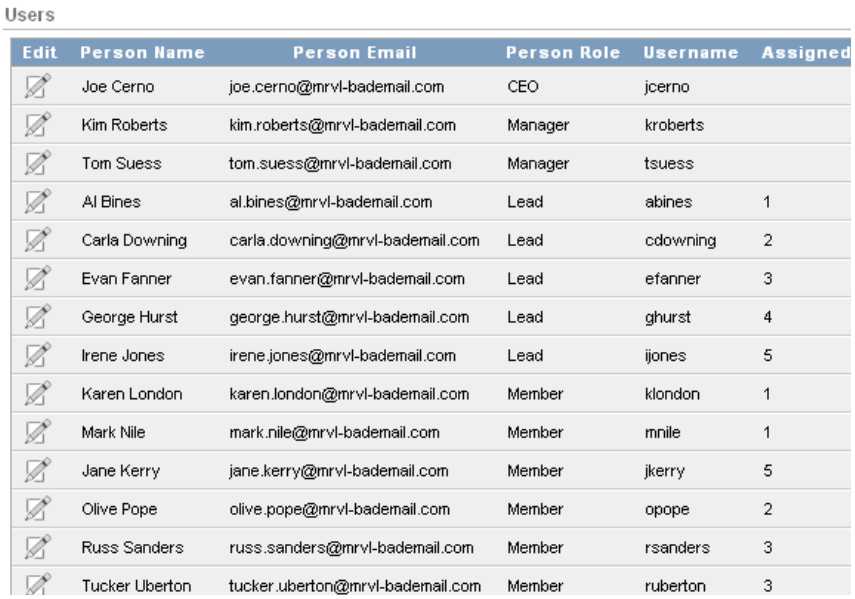
Create Pages for Maintaining People

To create pages for maintaining the `HT_PEOPLE` table:

1. Click **Application** on the Developer toolbar.
2. Click **Create Page**.
3. Select **Form** and click **Next**.
4. Select **Form on a Table with Report** and click **Next**.
5. For Table/View Owner, select the appropriate schema and click **Next**.
6. For Table/View Name, select **HT_PEOPLE** and click **Next**.
7. For Define Report Page:
   a. Page - Enter `4`.
   b. Page Name and Region Title - Enter `People`.
   c. Click **Next.**
8. For **Tab Options**, accept the default, **Do not use tabs**, and click **Next**.
9. For **Select Column(s)**, select all columns except `PERSON_ID` and click **Next**.
10. For **Edit Link Image**, select the fourth option (the word Edit) and click **Next**.
11. For Define Form Page:
    a. Page Number - Enter `5`.
    b. Page Name and Region Title - Enter `Create/Edit Person Information`.
    c. Click **Next.**
12. For Tab Options, accept the default, **Do not use tabs**, and click **Next**.
13. For Primary Key, accept the default, **PERSON_ID**, and click **Next**.
14. Specify the source for the primary key columns. Accept the default, **Existing Trigger**, and click **Next**.
15. For Select Column(s), select all the columns and click **Next**.
16. For Insert, Update and Delete, accept the defaults and click **Next**.
17. Review your selections and then click **Finish**.

To preview your page, click **Run Page**. As shown in Figure 14–6, notice the newly created report displays the demo data.

Figure 14–6 People Page



To preview the page for adding or editing people, click the **Edit** button in the far left column.

Modify the People Report Page

Next, you alter the People Report by changing the query to include a join to the Projects table and modify the headings.

Change the Query to Include a Join To change the query to include a join to the Projects table:

1. Go to the Page Definition for page 4 - People:
   a. If you are viewing a running form, click **Application** on the Developer toolbar.
   b. On the Application home page, click **4 - People.**
2. Under Regions, click **People**.
3. Scroll down to Source.
4. In Region Source, replace the existing query with the following:

```
SELECT a."PERSON_ID",
       a."PERSON_NAME",
       a."PERSON_EMAIL",
       a."PERSON_ROLE",
       b."PROJECT_NAME"
  FROM "#OWNER#"."HT_PEOPLE" a,
       "#OWNER#"."HT_PROJECTS" b
 WHERE a.assigned_project = b.project_id (+)
```

   Note that the outer join is necessary because the project assignment is optional.

Modify the Headings To modify the headings:

1. Click the Report Attributes tab at the top of the page.

2. Under Column Attributes, edit the following column headings:
   a. PERSON_ID - Remove the Heading **Edit**.
   b. PERSON_NAME - Change Heading to `Name`.
   c. PERSON_EMAIL - Change Heading to `Email`.
   d. PERSON_ROLE - Change Heading to `Role`.
   e. PROJECT_NAME - Change Heading to `Assigned Project` and then select **left** for Heading Alignment.

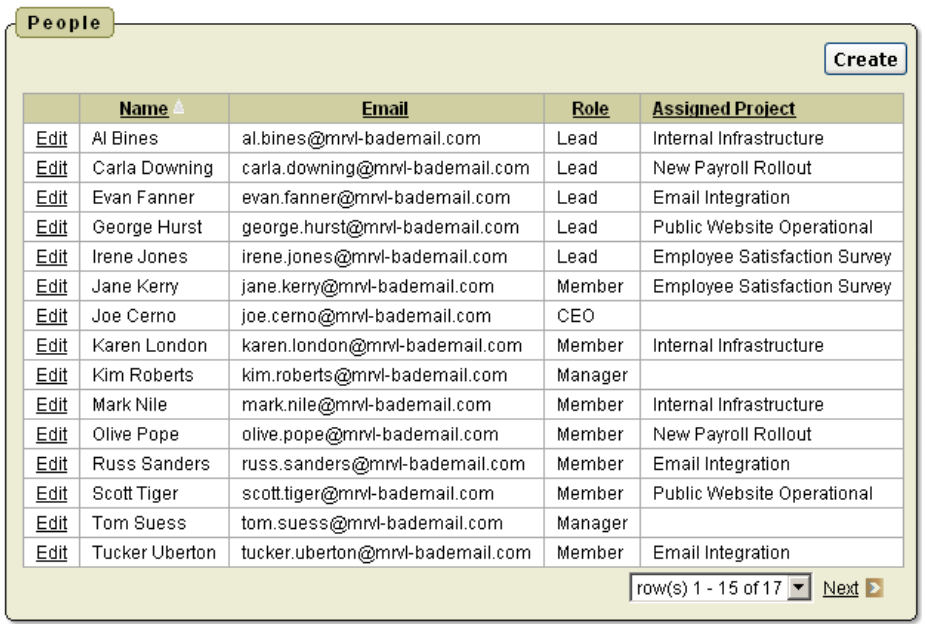
      Next, enable column heading sorting.
3. To enable column heading sorting:
   a. Select **Sort** for all columns except PERSON_ID.
   b. For `PERSON_NAME`, select **1** for Sort Sequence.

      This selection specifies `PERSON_NAME` as the default column to sort on. Note this functionality can be overridden by user selections.
   c. Scroll down to Sorting. For Ascending and Descending Image, select the light gray arrow.
4. Under Messages, enter the following in When No Data Found Message:

   ```
   No people found.
   ```
5. Click **Apply Changes** at the top of the page.

Run the Page To view your changes, click the **Run Page** icon in the upper right of the page. As shown in Figure 14–7, note the addition of a sort control on the Name column.

Figure 14–7 Revised People Page



Refine the Create/Edit People Page

Next, you customize the Create/Edit People page by adding lists of values to make it easier for users to select a Role or Assigned Project.

Add a List of Values for Projects To add a list of values for Projects:

1. Go to the Page Definition for page 5, Create/Edit Person:
   a. If you are viewing a form, click **Application** on the Developer toolbar.
   b. On the Application home page, click **5 - Create/Edit Person Information**.
2. Under Shared Components, locate the Lists of Values section and then click the **Create** icon.
3. For Source, accept the default, **From Scratch**, and then click **Next**.
4. For Name and Type:
   a. Name - Enter `PROJECTS`.
   b. Type - Select **Dynamic**.
   c. Click **Next**.
5. In Query, replace the existing statements with the following:

```
SELECT project_name d, project_id v
  FROM ht_projects
 ORDER BY d
```

6. Click **Create List of Values**.

Add a List of Values for Roles To add a list of values for Roles:

1. Under Shared Components, locate the Lists of Values section and then click the **Create** icon.
2. For Source, accept the default, **From Scratch**, and then click **Next**.
3. For Name and Type:
   a. Name - Enter `ROLES`.
   b. Type - Select **Static**
   c. Click **Next**.
4. Enter the display value and return value pairs shown in Table 14–4:

Table 14–4 Display Value and Return Value pairs

Display Value	Return Value
CEO	CEO
Manager	Manager
Lead	Lead
Member	Member

5. Click **Create List of Values**.
6. On the Lists of Values page, click the **Edit Page** icon in the upper right corner.

Edit Display Attributes To edit display attributes for P5_PERSON_ROLE:

1. Under Items, click **P5_PERSON_ROLE**.
2. From the Display As list in the Name section, select **Radiogroup**.
3. Scroll down to Label.
4. Change Label to `Role`.
5. Under Element, enter the following in Form Element Option Attributes:

   ```
   class="instructiontext"
   ```

   This specifies that the text associated with each radio group option is the same size as other items on the page.
6. Scroll down to List of Values.
7. From the Named LOV list, select **ROLES**.
8. Click **Apply Changes**.

To edit display attributes for `P5_ASSIGNED_PROJECT`:

1. Under Items, click **P5_ASSIGNED_PROJECT**.
2. From the Display As list in the Name section, select **Select List**.
3. Scroll down to List of Values.
4. Under List of Values:
   a. From the Named LOV list, select **PROJECTS**.

      Next, specify that the underlying column is not mandatory.

   b. For Null display value, enter:

      ```
      - None -
      ```
5. Click **Apply Changes**.

To alter the display of fields and field labels:

1. Under Items, click the **Edit All** icon.
2. For `P5_PERSON_NAME`:
   a. Prompt - Enter `Name`.
   b. Width - Enter `60`.
3. For `P5_PERSON_EMAIL`:
   a. Prompt - Enter `Email Address`.
   b. For Width, enter `60`.
4. Click **Apply Changes**.
5. Click the **Edit Page** icon in the upper right corner to return to the Page Definition for Page 5.

Create a Validation The Form on a Table with Report Wizard created not null validations for Name, Email, and Role. You must manually create another validation to ensure that Leads and Members have an assigned project while the CEO and Managers do not. As a best practice, it is generally best to use built-in validation types because they are faster. However, for this compound type of validation, you will write a PL/SQL validation.

To add validations to ensure the correct people are assigned projects:

1. Under Page Processing, Validations, click the **Create** icon.
2. For Level, accept the default, **Item level validation**, and click **Next**.
3. For Item, select **Create/Edit Person Information: 50. P5_ASSIGNED_PROJECT (Assigned Project)** and click **Next**.
4. For Validation Method:
   a. Select **PL/SQL** and click **Next**.
   b. Accept the default, **PL/SQL Expression** and click **Next**.
5. For Sequence and Name:
   a. Validation Name - Enter `PROJECT_MAND_FOR_LEADER_AND_MEMBER`.
   b. Accept the remaining defaults and click **Next**.
6. For Validation and Error Message:
   a. Validation - Enter:

```
(:P5_PERSON_ROLE IN ('CEO','Manager') AND
:P5_ASSIGNED_PROJECT = '%'||'null%') OR
(:P5_PERSON_ROLE IN ('Lead','Member') AND
:P5_ASSIGNED_PROJECT != '%'||'null%')
```

   Oracle Application Express passes nulls as `%null%`. It also replaces `%null%` with a null when it processes data. Therefore, to keep it in the validation, you need to break the string apart so that it is not recognized and replaced.

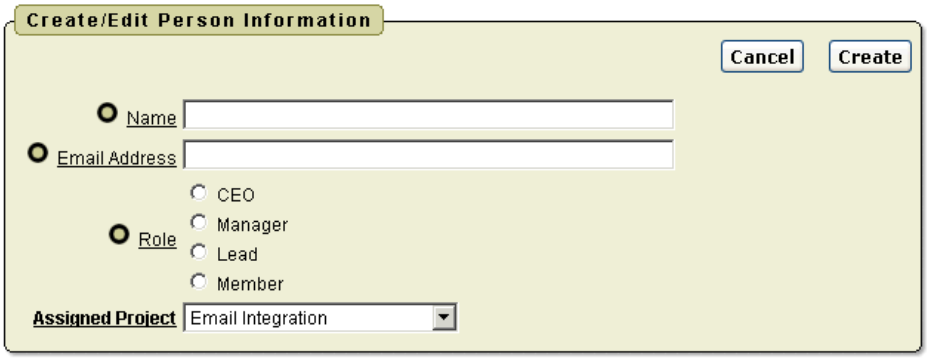
   b. Error Message - Enter:

```
Leads and Members must have an Assigned Project. CEO and Managers cannot
have an Assigned Project.
```

   c. Click **Next**.
7. Click **Create**.

Run the Page To view your changes, click the **Run Page** icon in the upper right of the page. The revised form appears as shown in Figure 14–8.

Figure 14–8 Revised Create/Edit Person Information Form



Try entering some records to test the validation. Enter a CEO with a project and then enter a Lead without a project. Both cases should fail and display the error message you defined.

Add Pages to Track Issues

Lastly, you need to create pages for HT_ISSUES. This application needs multiple views on Issues. You can create these views as single reports or as separate reports. For this exercise, you create a complex report that includes an Issues maintenance form. You then link this maintenance form in multiple places. Ultimately, the Issues report will display Issues by the person who identified the issue, project, assigned person, status, or priority.

Topics in this section include:

- Create a Report for HT_ISSUES
- Refine the Create/Edit Issues Page
- Refine the Issues Report
- Add a Page to Support Assigning Multiple Issues Simultaneously

Create a Report for HT_ISSUES

To create a report for maintaining HT_ISSUES:

1. Click **Application** on the Developer toolbar.
2. Click **Create Page**.
3. Select **Form** and click **Next**.
4. Select **Form on a Table with Report** and click **Next**.
5. For Table/View Owner, select the appropriate schema and click **Next**.
6. For Table/View Name, select **HT_ISSUES** and click **Next**.
7. On Define Report Page:
   a. Page Number - Enter 6.
   b. Page Name and Region Title - Enter Issues.
   c. Click **Next**.
8. For Tab Options, accept the default, **Do not use tabs**, and click **Next**.
9. For Select Column(s):
   a. Press **CTRL** to select the following columns:
      - ISSUE_SUMMARY
      - IDENTIFIED_BY
      - RELATED_PROJECT
      - ASSIGNED_TO
      - STATUS
      - PRIORITY
      - TARGET_RESOLUTION_DATE
      - ACTUAL_RESOLUTION_DATE
   b. Click **Next**.
10. For **Edit Link Image**, select the fourth option (the word Edit) and click **Next**.
11. On Define Form Page:

a. Page Number - Enter `7`.

b. For Page Name and Region Title, enter `Create/Edit Issues`.

c. Click **Next.**

12. For Tab Options, accept the default, **Do not use tabs**, and click **Next**.
13. For Primary Key, accept the default, **ISSUE_ID,** and click **Next**.
14. For Define the source for the primary key columns, accept the default, **Existing Trigger**, and click **Next**.
15. For Select Column(s), select all the columns and click **Next**.
16. For Insert, Update and Delete, accept the default value, **Yes**, and click **Next**.
17. Review your selections and click **Finish**.
18. Click **Edit Page**.

Refine the Create/Edit Issues Page

You refine the Create/Edit Page for the following reasons:

- Add lists of values to make it easier for users to select foreign key columns
- Organize and clean up items
- Change the display of audit columns
- Add a button to make data entry faster

Add Lists of Values Next, you need to add lists of values for Status, Priorities, and People.

To add a list of values for Status:

1. Go to the Page Definition for page 7, Create/Edit Issues. On the Page Definition, enter **7** in the Page field and click **Go**.
2. Under Shared Components, Lists of Values, click the **Create** icon.
3. For Create List of Values, accept the default, **From Scratch**, and click **Next**.
4. On Create List of Values:

   a. Name - Enter `STATUS`.

   b. For Type, select **Static**.

   c. Click **Next**.
5. Enter the Display Value and Return Value pairs shown in Table 14–5:

Table 14–5 Display Value and Return Value Pairs

Display Value	Return Value
Open	Open
On-Hold	On-Hold
Closed	Closed

6. Click **Create List of Values**.

To add a list of values for Priorities:

1. On the Lists of Values page, click **Create**.

2. For Create List of Values, accept the default, **From Scratch,** and click **Next**.
3. On Create List of Values:
   a. Name - Enter `PRIORITIES`.
   b. For Type, select **Static**.
   c. Click **Next**.
4. Enter the Display Value and Return Value pairs shown in Table 14–6.

Table 14–6 Display Value and Return Value Pairs

Display Value	Return Value
High	High
Medium	Medium
Low	Low

5. Click **Create List of Values**.

To add a list of values for People:

1. On the Lists of Values page, click **Create**.
2. For Create List of Values, accept the default, **From Scratch,** and click **Next**.
3. On Create List of Values:
   a. For Name, enter `PEOPLE`.
   b. For Type, select **Dynamic**.
   c. Click **Next**.
4. In Query, replace the existing statements with the following:

```
SELECT person_name d, person_id v
   FROM ht_people
 ORDER BY 1
```

5. Click **Create List of Values**.
6. Go to the Page Definition for page 7.

Edit Specific Items Next, you edit individual items.

To edit `P7_IDENTIFIED_BY`:

1. Under Items on the Page Definition for Page 7, click **P7_IDENTIFIED_BY**.
2. From the Display As list in the Name section, select **Select List**.
3. Under List of Values:
   a. Named LOV - Select **PEOPLE**.
   b. Display Null - Select **Yes**. The base column is mandatory, but you do not want the first name in the list becoming the default value.
   c. Null display value - Enter:

```
- Select Person -
```

4. Click the Next button (>) at the top of the page to go to the next item, `P7_IDENTIFIED_DATE`.

The Edit Page Item page appears.

To edit P7_IDENTIFIED_DATE:

1. From the Display As list in the Name section, select **Date Picker (DD-MON-YYYY)**.
2. Scroll down to Default:
   a. Default value - Enter:
   ```
   to_char(sysdate,'DD-MON-YYYY')
   ```
   b. Default Value Type - Select **PL/SQL Expression**.
3. Click the Next button (>) at the top of the page to go to the next item, P7_RELATED_PROJECT.

   The Edit Page Item page appears.

To edit P7_RELATED_PROJECT:

1. From the Display As list in the Name section, select **Select List**.
2. Scroll down to List of Values. For List of Values:
   a. Named LOV - Select **PROJECTS**.
   b. Display Null- Select **Yes**.
   c. Null display value - Enter:
   ```
    - Select Project -
   ```
3. Click the Next button (>) at the top of the page until you go to P7_STATUS.

To edit P7_STATUS:

1. From the Display As list in the Name section, select **Radiogroup**.
2. Under Label, enter the following in the Label field:
   ```
   Status:
   ```
3. Under Element, enter the following in the Form Element Option Attributes field:
   ```
   class="instructiontext"
   ```
4. Under Default, enter Open in the Default Value field.
5. Under List of Values:
   a. Named LOV - Select **STATUS**.
   b. Number of Columns - Enter 3.

      This selection enables the three valid values to display side by side.
6. Click the Next button (>) at the top of the page to go to P7_PRIORITY.

To edit P7_PRIORITY:

1. From the Display As list in the Name section, select **Radiogroup**.
2. Under Label, enter the following in the Label field:
   ```
   Priority:
   ```
3. Under Element, enter the following in the Form Element Option Attributes field:
   ```
   class="instructiontext"
   ```

4. Under Default, enter `Open` in Default value.
5. Under List of Values:
   a. Named LOV - Select **PRIORITIES**.
   b. Display Null - Select **Yes**.
   c. Number of Columns - Enter 4.

      This selection reflects the fact there are three valid values plus the null value.
   d. Null display value - Enter the following.

      ```
      None-
      ```
6. Click the Next button (>) at the top of the page to go to `P7_TARGET_RESOLUTION_DATE`.

To edit `P7_TARGET_RESOLUTION_DATE`:

1. From the Display As list in the Name section, select **Date Picker (DD-MON-YYYY)**.
2. Click the Next button (>) at the top of the page until you go to `P7_ACTUAL_RESOLUTION_DATE`.

To edit `P7_ACTUAL_RESOLUTION_DATE`:

1. From the Display As list in the Name section, select **Date Picker (DD-MON-YYYY)**.
2. Click **Apply Changes**.

Create Regions to Group Items Currently all items are grouped into one large region. Displaying items in logical groups makes data entry easier for users. Therefore, you next create four new regions named Buttons, Progress, Resolution, and Audit Information. You also rename an existing region.

To create new regions to group items:

1. Under Regions, click the **Create** icon.
2. Select **Multiple HTML** and then click **Next**.
3. For the first row:
   - For Sequence, enter `5`.
   - For Title, enter `Buttons`.
   - For Template, select **Button Region without Title**.
4. For the second row, in Title enter `Progress`.
5. For the third row, in Title enter `Resolution`.
6. For the fourth row, in Title enter `Audit Information`.
7. Click **Create Region(s)**.

Now that the new regions exist, rename the first region, Create/Edit Issues:

1. Under Regions, click **Create/Edit Issues**.
2. In **Title**, enter:.

   ```
   Issue Identification
   ```

3. Click **Apply Changes**.

Move Items to the Appropriate Regions Next, move each item to the appropriate region. Note that you also need to modify some item widths.

To move items to the appropriate regions:

1. Under Items, click the **Edit All** icon.

   The Page Items summary page appears.
2. Under Region, select **Progress** for the following items:
   - P7_ASSIGNED_TO
   - P7_STATUS
   - P7_PRIORITY
   - P7_TARGET_RESOLUTION_DATE
   - P7_PROGRESS
3. Under Region, select **Resolution** for the following items:
   - P7_ACTUAL_RESOLUTION_DATE
   - P7_RESOLUTION_SUMMARY
4. Under Region, select **Audit Information** for the following items:
   - P7_CREATED_DATE
   - P7_CREATED_BY
   - P7_LAST_MODIFIED_DATE
   - P7_LAST_MODIFIED_BY
5. Under Width, make the following edits:
   a. For P7_ISSUE_SUMMARY, enter `60`.
   b. For P7_IDENTIFIED_DATE, enter `12`.
   c. For P7_TARGET_RESOLUTION_DATE, enter `12`.
   d. For P7_ACTUAL_RESOLUTION_DATE, enter `12`.
6. Click **Apply Changes**.
7. Click the **Edit Page** icon in the upper right to return to the Page Definition of Page 7.

To move buttons to the Button region:

1. Under Buttons, click the **Edit All** icon.
2. Under Region, select **Buttons** for all buttons.
3. Click **Apply Changes**.
4. Click the **Edit Page** icon in the upper right to return the Page Definition of Page 7.

Change the Display of Audit Columns Because the Audit columns should be viewable but not editable, you need to make them display only. In the following exercise, you create a condition for the Audit Information region. As a result, the Audit Information region displays when a user edits an existing issue, but does not appear when a user creates a new issue.

To create a condition for the Audit Information region.

1. Under Regions, click **Audit Information**.
2. Scroll down to Conditional Display.
3. From Condition Type, select **Value of Item in Expression 1 is NOT NULL**.
4. In Expression 1, enter the following:

   ```
   P7_ISSUE_ID
   ```

5. Click **Apply Changes**.

Next, change the audit columns to display only.

To edit `P7_CREATED_DATE`:

1. Under Items, click **P7_CREATED_DATE**.
2. From the Display As list in the Name section, select **Display as Text (saves state)**.
3. Under Label:
   a. For Label, enter:

      ```
      Created Date:
      ```

   b. For Template, select **Optional Label with Help**.
   c. For HTML Table Cell Attributes, enter:

      ```
      class="instructiontext"
      ```

4. Under Source, enter the following in Format Mask:

   ```
   DD-MON-YYYY
   ```

5. Click the Next button (>) at the top of the page to go to the next item, `P7_CREATED_BY`.

To edit `P7_CREATED_BY`:

1. From the Display As list in the Name section, select **Display as Text (saves state)**.
2. Under Label:
   a. For Label, enter:

      ```
      Created By:
      ```

   b. For Template, select **Optional Label with Help**.
   c. For HTML Table Cell Attributes, enter:

      ```
      class="instructiontext"
      ```

3. Click the Next button (>) at the top of the page to go to the next item, `P7_LAST_MODIFIED_DATE`.

To edit `P7_LAST_MODIFIED_DATE:`

1. From the Display As list in the Name section, select **Display as Text (saves state)**.
2. Under Label:
   a. For Label, enter:

      ```
      Last Modified Date:
      ```

b. For Template, select **Optional Label with Help**.

c. For HTML Table Cell Attributes, enter:

```
class="instructiontext"
```

3. Under Source, enter the following in Format Mask:

```
DD-MON-YYYY
```

4. Click the Next button (>) at the top of the page to go to the next item, `P7_LAST_MODIFIED_BY`.

To edit `P7_LAST_MODIFIED_BY`:

1. From the Display As list in the Name section, select **Display as Text (saves state)**.
2. Under Label:
   a. For Label, enter:

```
Last Modified By:
```

   b. For Template, select **Optional Label with Help**.
   c. For HTML Table Cell Attributes, enter:

```
class="instructiontext"
```

3. Click **Apply Changes**.

Remove Unnecessary Validations The Form on a Table with Report Wizard created not null validations for Issue Summary, Identified By, Related Project, Status, Created Date, and Created By. Since the Audit columns are set by a trigger, you need to remove these validations.

To remove not null validations:

1. Delete P7_CREATED_DATE not null:
   a. Under Page Processing, Validations, click **P7_CREATED_DATE not null**.
   b. Click **Delete**.
   c. Click **OK** to confirm your selection.
2. Delete P7_CREATED_BY not null:
   a. Under Page Processing, Validations, click **P7_CREATED_DATE not null**.
   b. Click **Delete**.
   c. Click **OK** to confirm your selection.

Return the User to the Calling Page Because this Create/Edit page will be called from several places, when users finish with the display, they should return to the calling page. To accomplish this, you create an item and change the branch on the Create/Edit page. Every time the Create/Edit page is called, the item must be set with the number of the calling page.

To create a hidden item:

1. Under Items, click the **Create** icon.
2. For Select Item Type, select **Hidden** and click **Next**.
3. For Display Position and Name:

a. Item Name - Enter:

```
P7_PREV_PAGE
```

b. Region - Select **Issue Identification**.

c. Click **Next**.

4. Click **Create Item**.

   The Page Definition for page 7 appears.

   Next, edit the Cancel button.

To edit the Cancel button:

1. Under Buttons, click **Cancel**.
2. Scroll down to Optional URL Redirect.
3. In Page, enter:

   ```
   &P7_PREV_PAGE.
   ```

   Note the period at the end.

4. Click **Apply Changes**.

   Next, edit the branch.

To edit the branch:

1. Under Branches, select the After Processing branch, **Go to Page**.
2. Under Action, enter the following in Page (be sure to include the period):

   ```
   &P7_PREV_PAGE.
   ```

3. Click **Apply Changes**.

Add Functionality to Support Adding Multiple Issues Sequentially Next, you add functionality that enables users to add more than one issue at a time. To accomplish this, you first add a new button and then create a new branch.

To add a new button:

1. Under Buttons, click the **Copy** icon.
2. For Button to copy, click **CREATE**.
3. For Target Page, accept the default, **7**, and click **Next**.
4. For New Button:

   a. Button Name - Enter `CREATE_AGAIN`.

   b. Label - Enter `Create and Create Another`.

   c. Click **Copy Button**.

Functionally, the Copy Button currently works the same as the CREATE button. Next, create a branch that keeps the user on the Create page.

Note that this branch also resets P7_PREV_PAGE because the value of that item will be lost when the cache of the page is cleared. The sequence of this new branch will be 0. Setting the sequence to 0 makes the branch fire before the default branch but only when the Create and Create Another button is used.

To create a branch that keeps the user on the create page:

1. Under Page Processing, Branches, click the **Create** icon.
2. For Point and Type, accept the defaults and click **Next**.
3. For Target:
   a. Page - Enter 7.
   b. Clear Cache - Enter 7.
   c. Set these items - Enter the following:

      ```
      P7_PREV_PAGE
      ```

   d. With these values - Enter the following (be sure to include the period):

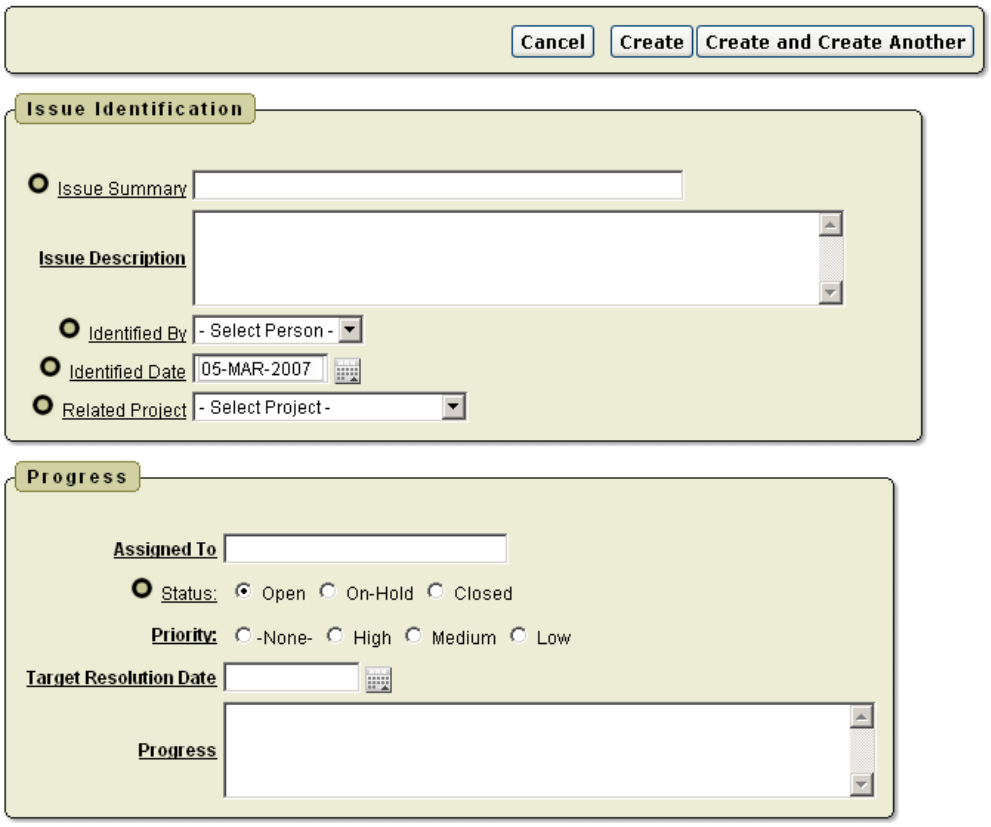
      ```
      &P7_PREV_PAGE.
      ```

   e. Click **Next**.
4. For Branch Conditions:
   a. Sequence - Enter 0.
   b. When Button Pressed - Select **CREATE_AGAIN**.
5. Click **Create Branch**.

   The Page Definition for page 7 appears.
6. Under Branches, select the newly created branch, **Go to Page**, next to 7.
7. Under Action, select the **include process success message** check box.
8. Click **Apply Changes**.

Run the Page To see the changes, click the **Run Page** icon. The new form appears as shown in Figure 14–9.

Figure 14–9 Create/Edit Issues Form



The branch you just created is looking for a value in P7_PREV_PAGE. Since the page was not called from another page, the value has not been set. You need to fix that next.

Refine the Issues Report

Next, you refine the Issues report page to support dynamic modification of the query. To accomplish this, you must:

- Move the Create button to a new region and edit the label
- Create new items that enable the user to restrict the query
- Add a `WHERE` clause to reference those new items
- Alter the report column attributes to display each person's name and the project
- Modify headings

Move Create Button to a New Region To create a new region for the Create button:

1. Go to the Page Definition for page 6, Issues.
2. Under Regions, click the **Create** icon.
3. Select **HTML** and click **Next**.
4. Select **HTML** for region container and click **Next**.
5. For Display Attributes:
   a. Title - Enter `Buttons`.
   b. Region Template - Select **Button Region without Title**.

c. Display Point - Select **Page Template Body (2. items below region content)**.

d. Click **Next**.

6. Click **Create Region**.

To move the Create button to the Buttons region:

1. Under Buttons, click the **Create** link.
2. Under Name, enter the following in the Text Label/Alt field:

```
Add a New Issue
```

3. Under Displayed, select **Buttons** in the Display in Region field.
4. Under Optional URL Redirect:

a. Set These Items - Enter:

```
P7_PREV_PAGE
```

b. For With These Values, enter `6`.

5. Click **Apply Changes**.

Change the Query and Display Next, change the query to display the actual values for people and projects instead of the ID and then clean up the report display.

To edit column attributes for ISSUE_ID:

1. Under Regions, select **Report** next to Issues.

   The Report Attributes page appears.
2. Click the **Edit Icon** to the left of ISSUE_ID.
3. Scroll down to Column Link. Under Column Link:

a. For Item 2, Name, enter:

```
P7_PREV_PAGE
```

b. For Item 2, Value, enter 6.

4. Click **Apply Changes**.

To edit column attributes for `IDENTIFIED_BY`, `RELATED_PROJECT` and `ASSIGNED_TO`:

1. Click the **Edit Icon** to the left of `IDENTIFIED_BY`.
2. Scroll down to Tabular Form Element. From Display As, select **Display as Text (based on LOV, does not save state).**
3. Scroll down to Lists of Values. From Named LOV, select **PEOPLE**.
4. Return to the top of the page and click the **Next** (>) icon.

   The Column Attributes page for RELATED_PROJECT appears.
5. Scroll down to Tabular Form Element. From Display As, select **Display as Text (based on LOV, does not save state).**
6. Scroll down to Lists of Values. Under List of Values:

a. Named LOV - Select **PROJECTS**.

b. Display Null - Select **Yes**.

   c. Null Text - Enter a hyphen (-).

7. Return to the top of the page and click the **Next** (>) icon.

   The Column Attributes page for ASSIGNED_TO appears.

8. Scroll down to Tabular Form Element. From Display As, select **Display as Text (based on LOV, does not save state).**

9. Scroll down to Lists of Values. Under List of Values:

   a. Named LOV - Select **PEOPLE**.

   b. Display Null - Select **Yes**.

   c. Null Text - Enter a hyphen (-).

10. Click **Apply Changes**.

    The Report Attributes page appears.

Next, you customize how the report displays by changing report attributes.

To alter the report display:

1. From Headings Type (the radiogroup row at the top of Column Attributes), accept the default, **Custom**.
2. For ISSUE_ID, delete the Heading text.
3. For ISSUE_SUMMARY, change the Heading to `Summary`.
4. For TARGET_RESOLUTION_DATE:

   a. Force the heading to wrap. In Heading, enter:

   ```
   Target<br>Resolution<br>Date
   ```

   b. For Column Alignment, select **center**.

   c. For Heading Alignment, select **center**.

5. To sort on ISSUE_ID:

   a. For all columns except ISSUE_ID, check **Sort**.

   b. For ISSUE_SUMMARY, select **1** for Sort Sequence.

6. Scroll down to Layout and Pagination. Specify the following:

   a. Show Null Values as - Enter a hyphen (-).

   b. Number of Rows - Enter `5`.

7. Under Sorting, select the light gray arrow for Ascending and Descending Image.
8. Under Messages, enter the following in When No Data Found Message:

   ```
   No issues found.
   ```

9. Click **Apply Changes**.

Add Support for Filtering Although the report now displays nicely, it does not support filtering by the end user. To add this functionality, you first create items that enable the user to set values to query against. You will store these new items in a new region that will display above the report.

To create a new region:

1. Under Regions, click the **Create** icon.

2. Select **HTML** and then click **Next**.
3. Select **HTML** for region container and click **Next**.
4. For Display Attributes:
   a. For Title, enter `Issue Report Parameters`.
   b. For Region Template, accept the default, **Reports Region**.
   c. For Sequence, enter `5`.
   d. Click **Next**.
5. Click **Create Region**.

Next, create the items.

To create the item for Identified By:

1. Under Items, click the **Create** icon.
2. For Select Item Type, select **Select List** and click **Next**.
3. For Select List Control Type, accept the default selection, **Select List**, and click **Next**.
4. For Display Position and Name:
   a. For Item Name - Enter `P6_IDENTIFIED_BY`.
   b. For Region, select **Issue Report Parameters**.
   c. Click **Next**.
5. For Identify List of Values:
   a. For Named LOV, select **PEOPLE**.
   b. For Null Text, enter:

      ```
      - All -
      ```

   c. For Null Value, enter:

      ```
      -1
      ```

   d. Click **Next**.
6. For Item Attributes, accept the defaults and click **Next**.
7. For Source, Default, enter:

   ```
   -1
   ```

8. Click **Create Item**.

To create an item for Assigned To:

1. Under Items, click the **Create** icon.
2. For Select Item Type, select **Select List** and click **Next**.
3. For Select List Control Type, accept the default selection, **Select List**, and click **Next**.
4. For Display Position and Name:
   a. For Item Name, enter `P6_ASSIGNED_TO`.
   b. For Region, select **Issue Report Parameters**.

c. Click **Next**.

5. For Identify List of Values:
   a. Named LOV - Select **PEOPLE**.
   b. For Null Text - Enter:

      ```
      - All -
      ```

   c. For Null Value - Enter:

      ```
      -1
      ```

   d. Click **Next**.
6. For Item Attributes, accept the defaults and click **Next**.
7. For Source, Default, enter:

   ```
   -1
   ```

8. Click **Create Item**.

To create an item for Status:

1. Under Items, click the **Create** icon.
2. For Select Item Type, select **Select List** and click **Next**.
3. For Select List Control Type, accept the default selection, **Select List**, and then click **Next**.
4. For Display Position and Name:
   a. Item Name - Enter `P6_STATUS`.
   b. For Region - Select **Issue Report Parameters**.
   c. Click **Next**.
5. For Identify List of Values:
   a. For Named LOV - Select **STATUS**.
   b. For Null Text - Enter:

      ```
      - All -
      ```

   c. For Null Value - Enter:

      ```
      -1
      ```

   d. Click **Next**.
6. For Item Attributes, accept the defaults and click **Next**.
7. For Source, Default, enter:

   ```
   -1
   ```

8. Click **Create Item**.

To create an item for Priority:

1. Under Items, click the **Create** icon.
2. For Select Item Type, select **Select List** and click **Next**.

3. For Select List Control Type, accept the default selection, **Select List**, and click **Next**.
4. For Identify Item Name and Display Position:
   a. For Item Name - Enter `P6_PRIORITY`.
   b. For Region - Select **Issue Report Parameters**.
   c. Click **Next**.
5. For List of Values:
   a. For Named LOV - Select **PRIORITIES**.
   b. For Null Text - Enter:

      ```
      - All -
      ```

   c. For Null Value - Enter:

      ```
      -1
      ```

   d. Click **Next**.
6. For Identify Item Attributes, accept the defaults and click **Next**.
7. For Source, for Default, enter:

   ```
   -1
   ```

8. Click **Create Item**.

To create an item for Related Project:

1. Under Items, click the **Create** icon.
2. For Select Item Type, select **Select List** and click **Next**.
3. For Select List Control Type, accept the default selection, **Select List** and click **Next**.
4. For Display Position and Name:
   a. For Item Name - Enter `P6_RELATED_PROJECT`.
   b. For Region - Select **Issue Report Parameters**.
   c. Click **Next**.
5. For Identify List of Values:
   a. For Named LOV - Select **PROJECTS**.
   b. For Null Text - Enter:

      ```
      - All -
      ```

   c. For Null Value - Enter:

      ```
      -1
      ```

   d. Click **Next**.
6. For Item Attributes, accept the defaults and click **Next**.
7. For Source, for Default, enter:

   ```
   -1
   ```

8. Click **Create Item**.

Next, create a Go button. This button enables the user to execute the query once they select report parameters. Buttons can be created in region positions or displayed among items.

To create Go button:

1. Under Buttons, click the **Create** icon.
2. For Button Region, select **Issue Report Parameters** and click **Next**.
3. For Button Position, select **Create a button displayed among this region's items**.

   This selection displays the button to the right of the last report parameter.
4. Click **Next**.
5. For Button Attributes:
   a. Button Name - Enter `P6_GO`.
   b. Button Style - Select **Template Based Button**.
   c. **Template** - Select **Button**.
6. Click **Create Button**.

   The Page Definition for page 6 appears.

Currently the items display stacked on top of one another. To use space more efficiently, change the position of P6_RELATED_PROJECT, P6_STATUS, and P6_PRIORITY so they display next to each other. Place P6_RELATED_PROJECT, P6_STATUS on the first line and P6_PRIORITY on the second line.

To change the position of P6_RELATED_PROJECT, P6_STATUS, and P6_PRIORITY:

1. Under Items, click the **Edit All** icon.
2. For P6_RELATED_PROJECT, P6_STATUS, and P6_PRIORITY, select **No** for New Line.
3. Click **Apply Changes**.
4. Click the Edit Page icon in the upper right corner to return to the Page Definition for page 6. The Edit Page icon resembles a small green piece of paper and pencil.

Next, you need to modify the report to react to the parameters. To accomplish this, you need to modify the query's `WHERE` clause as follows:

```
WHERE (IDENTIFIED_BY = :P6_IDENTIFIED_BY OR
       :P6_IDENTIFIED_BY = '-1')
   AND (RELATED_PROJECT = :P6_RELATED_PROJECT OR
       :P6_RELATED_PROJECT = '-1')
   AND (ASSIGNED_TO = :P6_ASSIGNED_TO OR
       :P6_ASSIGNED_TO = '-1')
   AND (STATUS = :P6_STATUS OR
       :P6_STATUS = '-1')
   AND (PRIORITY = :P6_PRIORITY OR
       :P6_PRIORITIY = '-1')
```

To use the preceding `WHERE` clause, you must convert the Issues region into a PL/SQL Function Body Returning a SQL Query.

To turn the Issues region into a PL/SQL Function Body Returning a SQL Query:

1. Under Regions, click **Issues**.

2. Under Name, for Type, select **SQL Query (PL/SQL function body returning SQL query)**.

3. For Region Source, replace the existing statements with the following:

```
DECLARE

   q VARCHAR2(32767); -- query
   w VARCHAR2(4000) ; -- where clause
   we VARCHAR2(1) := 'N'; -- identifies if where clause exists

BEGIN

   q := 'SELECT "ISSUE_ID", '||
        ' "ISSUE_SUMMARY", '||
        ' "IDENTIFIED_BY", '||
        ' "RELATED_PROJECT", '||
        ' "ASSIGNED_TO", '||
        ' "STATUS", '||
        ' "PRIORITY", '||
        ' "TARGET_RESOLUTION_DATE", '||

        ' "ACTUAL_RESOLUTION_DATE" '||
        ' FROM "#OWNER#"."HT_ISSUES" ';

   IF :P6_IDENTIFIED_BY != '-1'
      THEN
      w := ' IDENTIFIED_BY = :P6_IDENTIFIED_BY ';
      we := 'Y';
   END IF;

   IF :P6_RELATED_PROJECT != '-1'
      THEN
      IF we = 'Y'
         THEN
         w := w || ' AND RELATED_PROJECT = :P6_RELATED_PROJECT ';
      ELSE
         w := ' RELATED_PROJECT = :P6_RELATED_PROJECT ';
         we := 'Y';
      END IF;
   END IF;

   IF :P6_ASSIGNED_TO != '-1'
      THEN
      IF we = 'Y'
         THEN
         w := w || ' AND ASSIGNED_TO = :P6_ASSIGNED_TO ';
    ELSE
         w := ' ASSIGNED_TO = :P6_ASSIGNED_TO ';
         we := 'Y';
    END IF;
   END IF;

   IF :P6_STATUS != '-1'
      THEN
      IF we = 'Y'
         THEN
         w := w || ' AND STATUS = :P6_STATUS ';
      ELSE
         w := ' STATUS = :P6_STATUS ';
         we := 'Y';
```

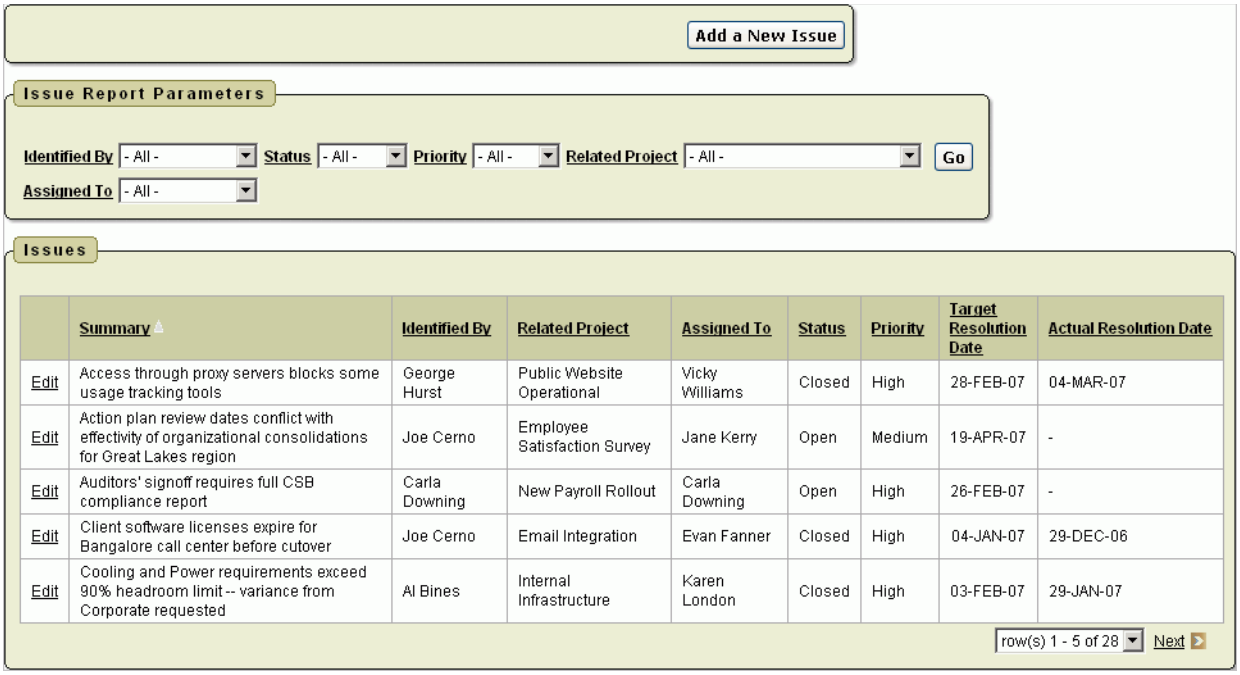
```
      END IF;
END IF;

   IF :P6_PRIORITY != '-1'
       THEN
       IF we = 'Y'
          THEN
          w := w || ' AND PRIORITY = :P6_PRIORITY ';
       ELSE
          w := ' PRIORITY = :P6_PRIORITY ';
          we := 'Y';
       END IF;
   END IF;

   IF we = 'Y'
      THEN q := q || ' WHERE '|| w;
   END IF;

RETURN q;

END;
```

4. Click **Apply Changes**.

Note that this function first sets the variable `q` to the original `SELECT` statement. It then builds a `WHERE` clause (`w`) composed of just the variables set by the user. If any variables have been set, it appends the `WHERE` clause to the original `SELECT` and passes that new `SELECT` to the database.

Run the Page The report is now complete. Click the **Run Page** icon. The revised report appears and should resemble Figure 14–10 on page 14-43.

Figure 14–10 Issues Report



To change the report parameters, make new selections under Issue Report Parameters and then click **Go**.

Add a Page to Support Assigning Multiple Issues Simultaneously

Currently, you can assign an issue by editing it. Next, you add a new page that enables users to assign multiple issues at once and modify the Related Project, Status, and Priority.

Create a Tabular Form

To add a new page to support assigning multiple issues:

1. Go to the Application home page.
2. Click **Create Page**.
3. Select **Form** and click **Next**.
4. Select **Tabular Form** and click **Next**.
5. For Table/View Owner:
   a. Table/View Owner- Select the appropriate schema.

      For this exercise, assume that the purpose of this form is to enable users to assign issues, or update existing records, and not create or delete issues.

   b. Allowed Operations - Select **Update Only**.
   c. Click **Next**.
6. For Table/View Name, select **HT_ISSUES** and click **Next**.
7. For Displayed Columns:
   a. Press **CTRL** and select the following columns:
      – ISSUE_SUMMARY
      – IDENTIFIED_BY
      – IDENTIFIED_DATE
      – RELATED_PROJECT
      – ASSIGNED_TO
      – STATUS
      – PRIORITY
   b. Click **Next**.
8. For Primary Key, accept the default, **ISSUE_ID**, and click **Next**.
9. For Primary Key Source, accept the default, **Existing trigger,** and click **Next**.
10. For Updatable Columns:
    a. Press **CTRL** and select the following columns:
       – RELATED_PROJECT
       – ASSIGNED_TO
       – STATUS
       – PRIORITY
    b. Click **Next**.
11. For Page and Region Attributes:
    a. Page - Enter `8`.

   b. Page Name - Enter `Assign Issues`.
   c. Region Title - Enter `Assign Issues`.
   d. Click **Next**.
12. For Tab Options, accept the default, **Do not use tabs**, and then click **Next**.
13. For Button Labels:
   a. For Cancel Button Label, accept the default.
   b. For Submit Button Label, enter `Apply Changes`.
   c. Click **Next**.
14. For Branching, accept the defaults and click **Next**.
15. Review your selections and then click **Finish**.

Add Lists of Values Once you have created the initial tabular form, you need to add lists of values to make it easier to select issues. Additionally, you need to restrict the query to display only unassigned issues.

To add lists of values:

1. From the Success page, click **Edit Page**.

   The Page Definition for page 8, Assign Issues, appears.
2. Under Regions, click **Assign Issues**.
3. Under Source, for Region Source, replace the existing statements with the following:

```
SELECT "ISSUE_ID",
       "ISSUE_SUMMARY",
       "IDENTIFIED_BY",
       "IDENTIFIED_DATE",
       "RELATED_PROJECT",
       "ASSIGNED_TO",
       "STATUS",
       "PRIORITY"
  FROM "#OWNER#"."HT_ISSUES"
WHERE assigned_to IS NULL
```

To edit report attributes:

1. Select the **Report Attributes** tab at the top of the page.

   Edit the heading for ISSUE_SUMMARY.
2. For ISSUE_SUMMARY, enter the following in the Heading field:

```
Summary
```

3. To sort by ISSUE_ID:
   a. For all columns except ISSUE_ID, select **Sort**.
   b. For IDENTIFIED_DATE, for **Sort Sequence**, select **1**.
4. Edit the following attributes for IDENTIFIED_BY:
   a. Click the **Edit** icon to the left of IDENTIFIED_BY.
   b. Under Tabular Form Element, for Display As, select **Display as Text (based on LOV, does not save state)**.

c. Scroll down to Lists of Values.

d. For Named LOV, select **PEOPLE**.

e. Click the Next button (>) at the top of the page to go to IDENTIFIED_DATE.

5. Edit the following attributes for IDENTIFIED_DATE:

a. Under Column Formatting, for Number/Date Format, enter `DD-MON-YYYY`.

b. Click the Next button (>) at the top of the page to go to the RELATED_ PROJECT column.

6. Edit the following attributes for RELATED_PROJECT:

a. Under Tabular Form Element, for Display As, select **Select List (named LOV)**.

b. Under List of Values, for Named LOV, select **PROJECTS**.

c. Click the Next button (>) at the top of the page to go to the ASSIGNED_TO column.

7. Edit the following attributes for ASSIGNED_TO:

a. Under Tabular Form Element, for Display As, select **Select List (named LOV)**.

b. Under List of Values:

– Named LOV - Select **PEOPLE**.

– Display Null - Select **Yes**.

– Null Text - Enter a hyphen (-).

c. Click the Next button (>) at the top of the page to go to the STATUS column.

8. Edit the following attributes for STATUS:

a. Under Tabular Form Element, for Display As, select **Select List (named LOV)**.

b. Under List of Values, for Named LOV, select **STATUS**.

c. Click the Next button (>) at the top of the page to go to the PRIORITY column.

9. Edit the following attributes for PRIORITY:

a. Under Tabular Form Element, for Display As, select **Select List (named LOV)**.

b. Under List of Values:

– From Named LOV, select **PRIORITIES**.

– For Display Null, select **Yes**.

– For Null Text, enter a hyphen (-).

c. Click **Apply Changes**.

The Report Attributes page appears.

10. Scroll down to Sorting. Under Ascending and Descending Image, select the light gray arrow.

11. Under Messages, enter the following in When No Data Found Message:

```
No Unassigned Issues.
```

12. Click **Apply Changes**.

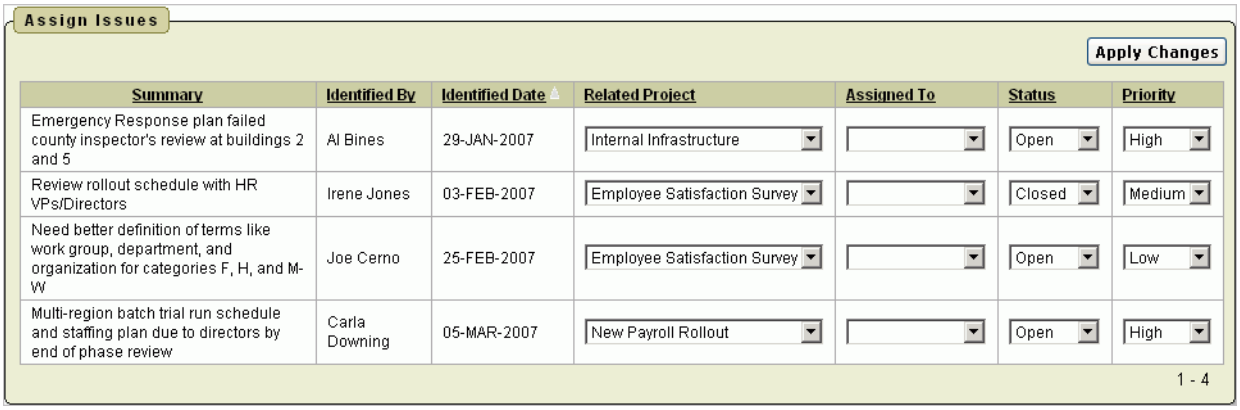
Delete the Unnecessary Cancel Button The wizard created an unnecessary Cancel button.

To delete the Cancel button:

1. On the Page Definition for page 8, click **CANCEL** in the Buttons section.
2. Click **Delete**.
3. Click **OK** to confirm your selection.

Run the Page The tabular form is now complete. To view the new form, click the **Run Page** icon. The Assign Issues form appears as shown in Figure 14–11.

Figure 14–11 Assign Issues



To assign an issue, make a selection from the Assigned To list and then click **Apply Changes**. Notice that once an issue has been assigned, the issue no longer displays.

Create Summary Reports

Lastly, you need to add four summary reports.

Topics in this section include:

- Add an Issue Summary by Project Report
- Add Resolved by Month Identified Report
- Add a Calendar to Display Issues
- Add a Bar Chart that Displays Average Days to Resolve

Add an Issue Summary by Project Report

The Issue Summary report enable users to select a project and then see a summary of issues related to that project. This report includes the following summary information:

- Date first issue identified
- Date last issue closed
- Total number of issues
- Number of issues by status
- Number of open issues by priority
- Assignments by status

To create this report, you code the information in two SQL statements. The first statement gathers information having a singular result and the second statement gathers information having multiple results.

To add an Issue Summary by Project report:

1. Go to the Application home page.
2. Click **Create Page**.
3. Select **Report** and click **Next**.
4. Select **SQL Report** and click **Next**.
5. For Page Attributes:
   a. Page - Enter `9`.
   b. Page Name - Enter `Issue Summary by Project`.
   c. Click **Next**.
6. For Tab Options, accept the default, **Do not use tabs**, and click **Next**.
7. For SQL Query:
   a. Enter the following SQL SELECT statement:

```
SELECT MIN(identified_date) first_identified,
       MAX(actual_resolution_date) last_closed,
       COUNT(issue_id) total_issues,
       SUM(DECODE(status,'Open',1,0)) open_issues,
       SUM(DECODE(status,'On-Hold',1,0)) onhold_issues,
       SUM(DECODE(status,'Closed',1,0)) closed_issues,
       SUM(DECODE(status,
                  'Open',decode(priority,null,1,0),
                  0))                    open_no_prior,
       SUM(DECODE(status,
                  'Open',decode(priority,'High',1,0),
                  0))                    open_high_prior,
       SUM(DECODE(status,
                  'Open',decode(priority,'Medium',1,0),
                  0))                    open_medium_prior,
       SUM(DECODE(status,
                  'Open',decode(priority,'Low',1,0),
                  0))                    open_low_prior
   FROM ht_issues
WHERE related_project = :P9_PROJECT
```

   b. Click **Next**.
8. For Report Attributes:
   a. Report Template - Select **default: vertical report, look 1 (include null columns)**.
   b. For Region Name - Enter `Issue Summary by Project`.
   c. Accept the remaining defaults and click **Next**.
9. Review your selections and click **Finish**.

Now that you have the first query, you need to edit the headings and create the item to control the related project. First, create a region to display above the report to contain the Project parameter.

Create a New Region To create a new region to display above the report:

1. From the Success page, click **Edit Page**.

   The Page Definition for page 9, Issue Summary by Project, appears.

2. Under Regions, click the **Create** icon.
3. Select **HTML** and click **Next**.
4. Select **HTML** for region container and click **Next**.
5. For Display Attributes:
   a. Title - Enter `Issue Summary Report Parameters`.
   b. Display Point - Select **Page Template Body (2. items below region content)**.
   c. For Sequence, enter `5`.
   d. Accept the remaining defaults and click **Next**.
6. Click **Create Region**.

Create the Project Item To create the Project item:

1. Under Items, click the **Create** icon.
2. For Select Item Type, select **Select List** and click **Next**.
3. For Select List Control Type, accept the default, **Select List**, and click **Next**.
4. For Display Position and Name:
   a. Item Name - Enter `P9_PROJECT`.
   b. Region - Select **Issue Summary Report Parameters**.
   c. Click **Next**.
5. For List of Values:
   a. Named LOV - Select **PROJECTS**.
   b. Null Text - Enter:

      ```
      - Select -
      ```

   c. Null Text - Enter:

      ```
      - 1
      ```

   d. Click **Next**.
6. For Item Attributes, accept the defaults and click **Next**.
7. For Source, for Default, enter `-1`.
8. Click **Create Item**.

Create a Go Button To create a Go button to execute the query:

1. Under Buttons, click the **Create** icon.
2. For Button Region, select **Issue Summary Report Parameters** and click **Next**.
3. For Button Position, select **Create a button displayed among this region's items** and click **Next**.
4. For Button Attributes:
   a. Button Name - Enter `P9_GO`.
   b. Button Style - Select **Template Based Button**.
   c. Template - Select **Button**.

5. Click **Create Button**.

Edit Headings and Report Settings Next, you need to edit the headings and report setting for the report region. You also need to set the report regions to conditionally display when the user has selected a project.

To edit the headings and report settings:

1. Under Regions, click **Report** next to Issue Summary by Project.
2. For Headings Type, select **Custom**.
3. Under Column Attributes:
   a. Change the Heading for FIRST_IDENTIFIED to:

      ```
      First Issue Identified:
      ```

   b. Change the Heading for LAST_CLOSED to:

      ```
      Last Issue Closed:
      ```

   c. Change the Heading for TOTAL_ISSUES to:

      ```
      Total Issues:
      ```

   d. Change the Heading for OPEN_ISSUES to:

      ```
      Open Issues:
      ```

   e. Change the Heading for ONHOLD_ISSUES to:

      ```
      On-Hold Issues:
      ```

   f. Change the Heading for CLOSED_ISSUES to:

      ```
      Closed Issues:
      ```

   g. Change the Heading for OPEN_NO_PRIOR to:

      ```
      Open Issues with No Priority:
      ```

   h. Change the Heading for OPEN_HIGH_PRIOR:

      ```
      Open Issues of High Priority:
      ```

   i. Change the Heading for OPEN_MEDIUM_PRIOR to:

      ```
      Open Issues of Medium Priority:
      ```

   j. Change the Heading for OPEN_LOW_PRIOR:

      ```
      Open Issues of Low Priority:
      ```

4. Scroll down to Layout and Pagination. Specify the following:
   a. For Show Null Values as, enter a hyphen (-).
   b. For Pagination Scheme, select **- No Pagination Selected -**.
5. Select the **Region Definition** tab at the top of the page.
   a. Scroll down to Conditional Display.
   b. For Condition Type, select **Value of Item in Expression 1 Is NOT Contained within Colon Delimited List in Expression 2.**

c. In Expression 1, enter `P9_PROJECT`.

d. For Expression 2, enter `-1`.

6. Click **Apply Changes**.

Create a Query to Retrieve Assignments To create a query to retrieve assignments by status.

1. Under Regions, click the **Create** icon.
2. Select **Report** and then click **Next**.
3. For Report Implementation, select **SQL Report** and click **Next**.
4. For Display Attributes:
   a. Title - Enter `Assignments by Status`.
   b. Column - Select **2**.
   c. Accept the remaining defaults and click **Next**.
5. For Source:
   a. SQL Query or PL/SQL function returning a SQL Query - Enter the following:

```
SELECT p.person_name,
       i.status,
       COUNT(i.issue_id) issues
  FROM ht_issues i,
       ht_people p
WHERE i.related_project = :P9_PROJECT
  AND i.assigned_to = p.person_id
GROUP BY person_name, status
```

   b. Rows Per Page - Enter `20`.
   c. Break Columns - Select **Column 1**.
   d. Accept the remaining defaults and click **Next**.
6. For Conditional Display:
   a. Condition Type - Select **Value of Item in Expression 1 Is NOT Contained within Colon Delimited List in Expression 2**.
   b. In Expression 1 enter:

```
P9_PROJECT
```

   c. For Expression 2 enter `-1`.
7. Click **Create Region**.

To edit headings and report settings:

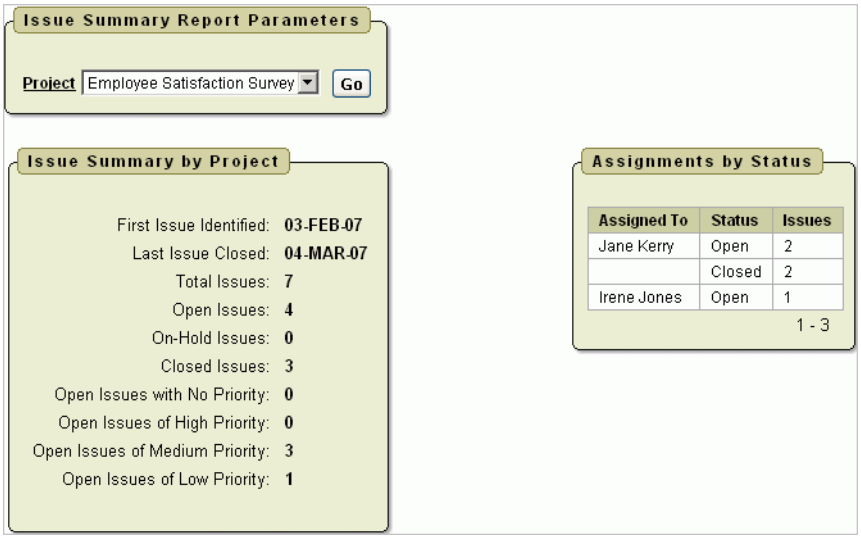
1. Under Regions, click **Report** next to Assignments by Status.
2. For Headings Type, select **Custom**.
3. For PERSON_NAME, change Heading to `Assigned To`.
4. Scroll down to Layout and Pagination. From Pagination Scheme, select **Row Ranges 1-15 16-30 in select list (with pagination)**.
5. Scroll down to Messages. In When No Data Found Message, enter:

```
No issues found.
```

6. Click **Apply Changes**.

Run the Page To see your newly created report, click the **Run Page** icon. Note that initially no data displays since no project is selected. Select a project and click **Go**. Your report should resemble Figure 14–12.

Figure 14–12 Issue Summary by Project Report



Add Resolved by Month Identified Report

The Resolved by Month Identified report is a line chart. This report first calculates the number of days it took to resolve each closed issue, averaged by the month the issue was identified, and finally displayed by the month.

To add a Resolved by Month Identified report:

1. Go to the Application home page.
2. Click **Create Page**.
3. Select **Chart** and click **Next**.
4. Select **SVG Chart** and click **Next**.
5. Select **Line** and click **Next**.
6. For Page Attributes:
   a. Page Number - Enter `10`.
   b. Page Name and Region Name - Enter `Resolved by Month Identified`.
   c. Accept the remaining defaults and click **Next**.
7. For Tab Options, accept the default, **Do not use tabs**, and click **Next**.
8. For Query:
   a. Series Name - Enter `Resolved`.
   b. SQL - Enter the following:

```
SELECT NULL l,
       TO_CHAR(identified_date,'Mon YYYY') month,
       AVG(actual_resolution_date-identified_date) days
```

```
 FROM ht_issues
WHERE status = 'Closed'
GROUP BY TO_CHAR(identified_date,'Mon YYYY')
```

Note that this query has no link (that is, the l column). It extracts the month from the identified date so that the data can be grouped by month. Lastly, it calculates the average number of days it took for the issues to be closed that were identified in that month.

c. For When No Data Found Message, enter:

```
No Closed Issues found.
```

d. Click **Next**.

9. Review your selections and click **Finish**.

Edit the Chart Next, add a correct axis label and turn off the Chart Title and legend.

To edit the chart:

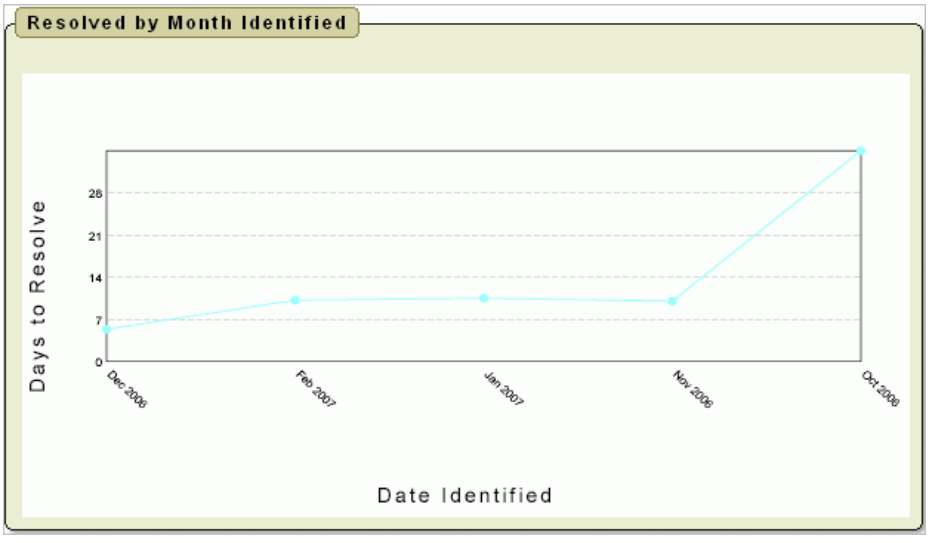
1. From the Success page, select **Edit Page**.

   The Page Definition for page 10, Resolved by Month Identified, appears.

2. Under Regions, click **SVG Chart**, next to Resolved by Month Identified.
3. Under Chart Settings:
   a. Chart Height - Enter `300`.
   b. Deselect **Show Legend**.
4. Scroll to Axes Setting:
   a. For X Axis Title, enter `Date Identified`.
   b. For Y Axis Title, enter `Days to Resolve`.
5. Click **Apply Changes**.

Run the Page To view your newly created line chart, click the **Run Page** icon. Your line chart should resemble Figure 14–13.

Figure 14–13 Resolved by Month Identified Line Chart



Add a Calendar to Display Issues

The Target Resolution Dates report is a calendar that displays issues that have not yet closed along with the assigned person on the day that corresponds to the issue target resolution date.

Create a Calendar To create a calendar of target resolution dates:

1. Go to the Application home page.
2. Click **Create Page**.
3. Select **Calendar** and click **Next**.
4. Select **SQL Calendar** and click **Next**.
5. For Page Attributes:
   a. Page Number - Enter `11`.
   b. Page Name and Region Name - Enter `Target Resolution Dates`.
   c. Click **Next**.
6. For Tab Options, accept the default, **Do not use tabs**, and click **Next**.
7. For Table/View Owner:
   a. In Enter SQL Query, enter the following:

```
SELECT I.TARGET_RESOLUTION_DATE,
       I.ISSUE_SUMMARY ||
          ' ('||nvl(P.PERSON_NAME,'Unassigned') ||') ' disp,
       I.ISSUE_ID
  FROM HT_ISSUES I,
       HT_PEOPLE P
  WHERE I.ASSIGNED_TO = P.PERSON_ID (+)
   AND (I.RELATED_PROJECT = :P11_PROJECT OR
        :P11_PROJECT = '-1')

AND I.STATUS != 'Closed'
```

   b. Click **Next**.

   Note that:

   – The `target_resolution_date` is the date on which the issue displays
   – The `issue_summary` is concatenated with the person assigned
   – The `issue_id` does not display, but is used to create a link to enable the user to view and edit the issue

8. For Date/Display Columns:
   a. Date Column - Select **TARGET_RESOLUTION_DATE**.
   b. For Display Column - Select **DISP**.
   c. Click **Next**.
9. Review your selections and click **Finish**.

Add an Item to Support Project Look Up To enable the user to look up one project or all projects, you need to add an item.

To add an item to support project look up:

1. From the Success page, select **Edit Page**.

   The Page Definition for page 11, Target Resolution Dates, appears.
2. Under Items, click the **Create** icon.
3. For Item Type, select **Select List** and click **Next**.
4. For Select List Control Type, select **Select List** and click **Next**.
5. For Display Position and Name:
   a. Item Name - Enter `P11_PROJECT`.
   b. Accept the remaining defaults and click **Next**.
6. For List of Values:
   a. Named LOV - Select **PROJECTS**.
   b. Null Text - Enter:

      ```
      - All Projects -
      ```

   c. For Null Value - Enter:

      ```
      -1
      ```

   d. Click **Next**.
7. For Item Attributes, accept the defaults and click **Next**.
8. For Source, Default, enter:

   ```
   -1
   ```

9. Click **Create Item**.

Create a Go Button To create a Go button to execute the query:

1. Under Buttons, click the **Create** icon.
2. For Button Region, select **Target Resolution Dates** and click **Next**.
3. For Button Position, select **Create a button displayed among this region's items** and click **Next**.
4. For Button Attributes:
   a. Button Name - Enter `P11_GO`.
   b. Button Style - Select **Template Based Button**.
   c. Template - Select **Button**.
5. Click **Create Button**.

Modify Calendar Attributes Lastly, you need to modify the Calendar Attributes to add link support for viewing and editing the displayed issues. To accomplish this, you need to call page 7, View/Edit Issues, clear any data from the page and pass in the current issue ID along with the fact that page 11 was the calling page. Then, you need to add a note that displays when the query excludes Closed issues.

To modify the Calendar Attributes:

1. Under Regions, click **Calendar** to the right of Target Resolution Dates.
2. Scroll down to Column Link, enter the following:

a. Page - Enter 7.

b. Clear Cache - Enter 7.

c. Set these items - Enter:

```
P7_ISSUE_ID,P7_PREV_PAGE
```

d. For With these values - Enter:

```
#ISSUE_ID#,11
```

3. Select the **Region Definition** tab at the top of the page.
4. Scroll down to Header and Footer.
5. In Region Footer, enter the following:

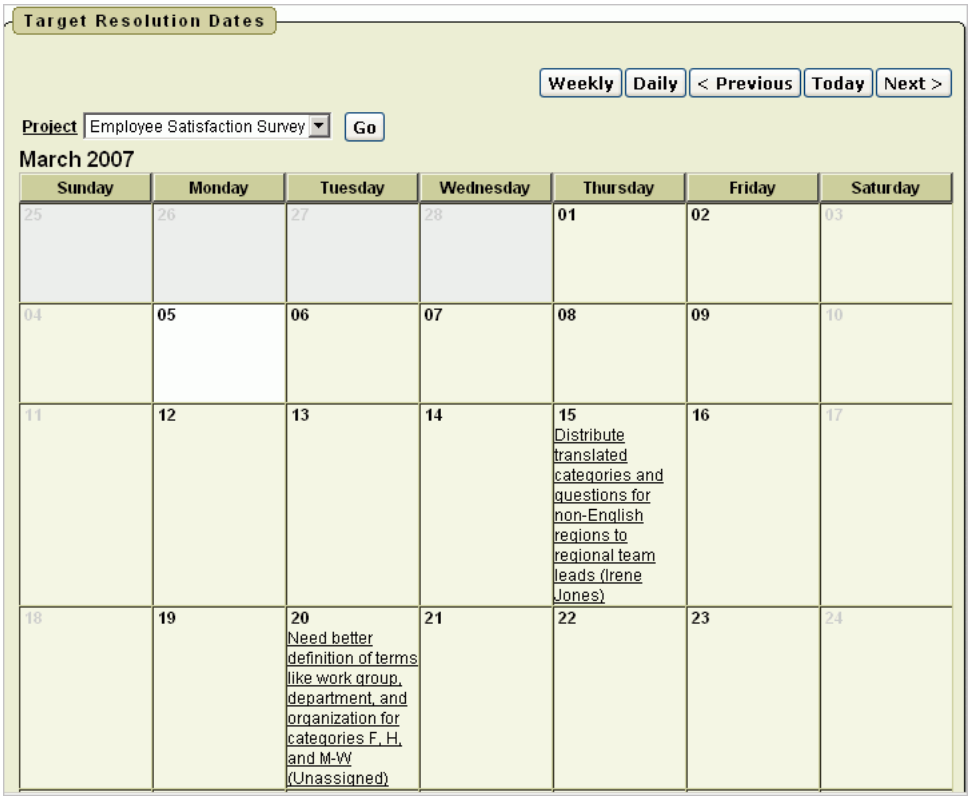
```
This excludes Closed issues.
```

6. Click **Apply Changes**.

Run the Page To see your newly created calendar, click the **Run Page** icon. Your report should resemble Figure 14–14 on page 14-56. Note that you can click **Weekly** or **Daily** to see the corresponding calendar views.

Note that you can also click the text displayed for an issue to display the Edit Issue page. To return to the calendar, click **Cancel**.

Figure 14–14 Target Resolution Dates Report



Add a Bar Chart that Displays Average Days to Resolve

The Average Days to Resolve report is a bar chart that calculates the number of days it takes to resolve each closed issue and then averages that number by assigned person.

To add the Average Days to Resolve report:

1. Go to the Application home page.
2. Click **Create Page**.
3. Select **Chart** and click **Next**.
4. Select **HTML Chart** and click **Next**.
5. Select **Bar (HTML)** and click **Next**.
6. For Page Attributes:
   a. Page - Enter `12`.
   b. Page Name and Region Name - Enter `Average Days to Resolve`.
   c. Accept the remaining defaults and click **Next**.
7. For Tab Options, accept the default, **Do not use tabs**, and click **Next**.
8. For Chart Definition:
   a. Chart SQL - Replace the existing statements with the following:

```
SELECT NULL l,
       NVL(p.person_name,'None Assigned') person,
       AVG(i.actual_resolution_date-i.identified_date) days
  FROM ht_issues i,
       ht_people p
 WHERE i.assigned_to = p.person_id (+)
   AND i.status = 'Closed'
 GROUP BY p.person_name
```

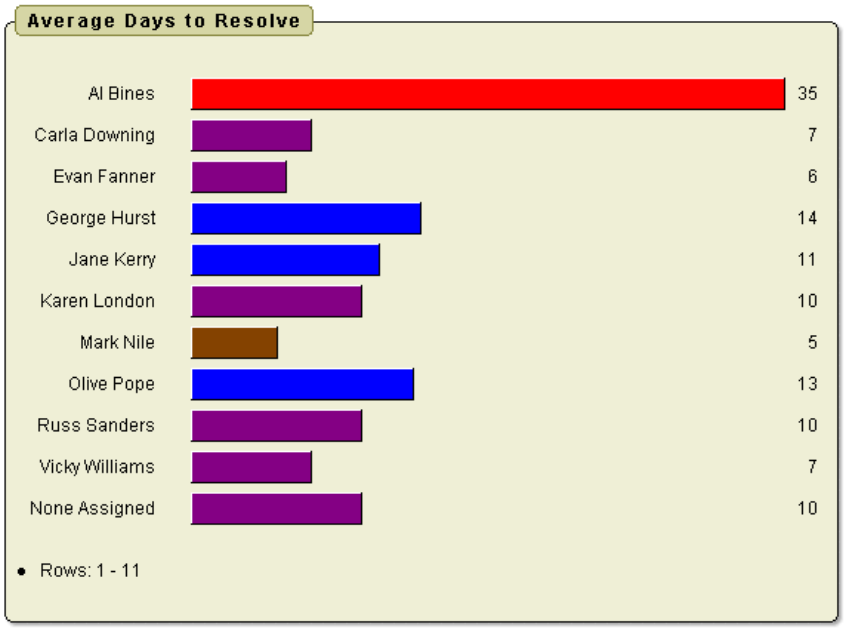
   In the above SELECT statement:

   – The first item selected is the link. This report does not link to any other page, and so `NULL` was selected.
   – The second item is the person's name, or `None Assigned` if `assigned_to` is `NULL`.
   – The third item selected is the average number of days it took for that person to resolve all their issues so the issues have a status of closed.

   b. For Include in summary, select only **Number of data points**. Deselect all other options.
   c. Accept the remaining defaults and click **Next**.
9. Review your selections and click **Finish**.

Run the Page To view your newly created bar chart, click **Run Page**. Your report should resemble Figure 14–15.

Figure 14–15 Average Days to Resolve Report



Add Content to the Home Page

Now that you have completed all the detail pages, you next need to add content to the home page and tie all the pages together. In this section, you modify the home page to display the following information:

- A menu of all available reports
- Navigation to the maintenance pages
- A button to **Add a New Issue**
- Overdue Issues
- Recently Opened Issues
- Open Issues by Project as a chart
- Unassigned Issues

Topics in this section include:

- Add a List
- Add Maintenance Navigation
- Add a New Issues Button
- Add An Overdue Issues Report
- Add an Unassigned Issues Report
- Add a Recently Opened Issues Report
- Add an Open Issues by Project Pie Chart

Add a List

First, you add a menu implemented as a list.

To add a menu:

1. Go to the Application home page.
2. Click the **Shared Components** icon.
3. Under Navigation, click **Lists**.
4. Click **Create**.
5. For Name, enter `Main Menu`.
6. For List Template, select **Vertical Sidebar List**.
7. Click **Create**.

Create List Entries Now that the list has been created, you add list items to it. You need to add one list item for each report page.

To add a list item for Assign Issues:

1. Click the **Create List Entry** button on the far right of the page.
2. For List Entry Label, enter `Assign Issues`.
3. Under Target, specify the following
   a. Page - Select **8**.
   b. Select **reset pagination for this page**.
4. Click **Create**.

   The List Entries page appears.

Now you need to create four more list items, one for each of the reports in your application.

To add four more list items:

1. On the List Entries page, click **Create List Entry**.
2. To define a list item for Issues:
   a. For Sequence, enter `20`.
   b. For List Entry Label, enter `Issues`.
   c. Under Target, specify the following:
      – Page - Select `6`.
      – Select **reset pagination for this page**.
      – Clear Cache - Enter **6**.

        This clears any selections for page 6 from the session state.
3. Click **Create and Create Another**.
4. To define a list item for Issue Summary:
   a. For Sequence, enter `30`.
   b. For List Entry Label, enter `Issue Summary by Project`.
   c. Under Target, specify the following:
      – Page - Select **9**.
      – Select **reset pagination for this page**.
      – Clear Cache - Enter **9**.

5. Click **Create and Create Another**.
6. To define a list item for Resolved by Month Identified:
   a. For Sequence, enter `40`.
   b. For List Entry Label, enter `Resolved by Month Identified (chart).`
   c. Under Target, select **10** for Page.
7. Click **Create and Create Another**.
8. To define a list item for Target Resolution Dates:
   a. For Sequence, enter `50`.
   b. For List Entry Label, enter `Target Resolution Dates (calendar).`
   c. Under Target, specify the following:
      – Page - Select **11**.
      – Select **reset pagination for this page**.
9. Click **Create and Create Another**.
10. To define a list item for Average Days to Resolve:
    a. For Sequence, enter `60`.
    b. For List Entry Label, enter `Average Days to Resolve (chart).`
    c. Under Target, select **12** for Page.
11. Click **Create**.

Include the List on the Home Page Now that the list is created, you need to include it on the home page. To display the list in the left margin, you need to change the page template to one that supports the appropriate region position.

To change the page template on the home page:

1. Click the **Edit Page** icon in the upper right corner.

   The Page Definition for page 12, Average Days to Resolve, appears.
2. In the Page field, enter `1` and click **Go**.
3. Under Page, click the **Edit page attributes** icon.

   The Page attributes page appears.
4. Under Display Attributes, for Page Template, select **No Tabs with Sidebar**.
5. Click **Apply Changes**.

Next, create a region to contain your menu.

To create a new region:

1. Under Regions, click the **Create** icon.
2. Select **List** and click **Next**.
3. For Display Attributes:
   a. Title - Enter `Menu`.
   b. Region Template - Select **No Template**.
   c. Display Point - Select **Page Template Region Position 2** (or select the quick link **[Pos. 2]**).

   d. Click **Next**.

4. For Source, select **Main Menu**.
5. Click **Create List Region**.

Add Maintenance Navigation

Next, you need to add maintenance navigation as a list. This list will display just below the reports in the left margin.

1. Go to the Application home page.
2. Click the **Shared Components** icon.
3. Under Navigation, click **Lists**.
4. Click **Create**.
5. For Name, enter `Maintenance`.
6. For List Template, select **Vertical Sidebar List**.
7. Click **Create**.

Next, create three list items. The first list item acts as a separator between the two navigation regions. The other two enable users to view people and projects.

To add list items:

1. Click **Create List Entry**.
2. To define the first list item:
   a. For List Entry Label, enter:

      ```
       
      ```

   b. Under Target, select **1** for Page.
3. Click **Create and Create Another**.
4. To define a list item for Projects:
   a. For Sequence, enter `20`.
   b. For List Entry Label, enter:

      ```
      Projects
      ```

   c. Under Target:
      – For Page, select **2**.
      – Check **reset pagination for this page**.
5. Click **Create and Create Another**.
6. To define a list item for People:
   a. For Sequence, enter `30`.
   b. For List Entry Label, enter:

      ```
      People
      ```

   c. Under Target:
      – For Page, select **4**.
      – Check **reset pagination for this page**.

7. Click **Create**.

To create a region to display the new list.

1. Click the **Edit Page** icon in the upper right corner.
2. Under Regions, click the **Create** icon.
3. Select **List** and click **Next**.
4. For Display Attributes:
   a. Title - Enter `Maintenance`.
   b. Region Template - Select **No Template**.
   c. Display Point - Select **Page Template Region Position 2** (or select the quick link **[Pos. 2]**).
   d. Click **Next**.
5. For Source, select **Maintenance**.
6. Click **Create List Region**.

Add a New Issues Button

Next, you create a button to navigate the user to page 7, Create/Edit Issue.

To create a region to contain the button:

1. Under Regions, click the **Create** icon.
2. Select **HTML** and click **Next**.
3. Select **HTML** for the region container and click **Next**.
4. For Display Attributes:
   a. Title - Enter `Buttons`.
   b. Region Template - Select **No Template**.
   c. Display Point - Select **Page Template Region Position 1** (or select the quick link **[Pos. 1]**).
   d. Click **Next**.
5. Click **Create Region**.

To add a button:

1. Under Buttons, click the **Create** icon.
2. For Button Region, select **Buttons** and click **Next**.
3. For Button Position, accept the default, **Create a button in a region position**, and then click **Next**.
4. For Button Attributes:
   a. Button Name - Enter `ADD`.
   b. Label - Enter:

      ```
      Add a New Issue
      ```

   c. Action - Select **Redirect to URL without submitting page**.
   d. Click **Next**.

5. For Button Template, select **Button** and click **Next**.
6. For Position, select **Top of Region**, accept the remaining defaults, and click **Next**.

   On the Branching page, you need to call the correct page, clear the cache, and specify that the Create and Cancel buttons return the user to the home page.
7. For Branching, specify the following:
   a. Page - Select **7**.
   b. Clear Cache - Enter `7`.
   c. Set these items - Enter:

      ```
      P7_PREV_PAGE
      ```

   d. With these values - Enter `1`.
8. Click **Create Button**.

Add An Overdue Issues Report

Next, add some content to the home page. In this exercise, you add a report to display overdue issues. The query for this report retrieves all unclosed issues with a past target resolution date.

Add a Report of Overdue Issues To add a report to display overdue issues:

1. Under Regions, click the **Create** icon.
2. Select **Report** and then click **Next**.
3. For Report Implementation, select **SQL Report** and click **Next**.
4. For Display Attributes, enter `Overdue Issues` in the Title field and click **Next**.
5. For Source, enter the following in Enter SQL Query:

   ```
   SELECT i.issue_id,
          i.priority,
          i.issue_summary,
          p.person_name assignee,
          i.target_resolution_date,
          r.project_name
     FROM ht_issues i,
          ht_people p,
          ht_projects r
    WHERE i.assigned_to = p.person_id (+)
      AND i.related_project = r.project_id
      AND i.target_resolution_date < sysdate
      AND i.status != 'Closed'
   ```

   The outer join is necessary because the assignment is optional.
6. Click **Create Region**.

Edit the Overdue Issues Report Now that the region has been created, you need to edit the headings and turn the summary into a link to display the issue details.

To edit the column headings:

1. Under Regions, click **Report** next to Overdue Issues.
2. For Headings Type, select **Custom**.

3. For ISSUE_ID, remove the Heading.
4. For ISSUE_SUMMARY, enter the following for Heading:

   ```
   Summary
   ```

5. For ASSIGNEE, change the Heading to:

   ```
   Assigned To
   ```

6. For TARGET_RESOLUTION_DATE:
   a. For Heading, enter:

      ```
      Target<br>Resolution<br>Date
      ```

   b. For Column Alignment, select **center**.
   c. For Heading Alignment, select **center**.
7. For **ISSUE_ID**, deselect **Show**.

   This enables the query to pass in the link, but not display it.
8. To sort by issue ID, select **Sort** for all columns except ISSUE_ID.
9. For TARGET_RESOLUTION_DATE, select **1** for Sort Sequence.
10. For ISSUE_SUMMARY, select **2** for Sort Sequence.

To edit column attributes for ISSUE_SUMMARY:

1. Click the **Edit** icon to the left of ISSUE_SUMMARY.
2. Scroll down to Column Link:
   a. For Link Text, click the quick link of **[Icon 3]**.
   b. For Page, select **7**.
   c. For Clear Cache, select **7**.
   d. For Item 1, enter the Name:

      ```
      P7_ISSUE_ID
      ```

   e. For Item 1, enter the Value:

      ```
      #ISSUE_ID#
      ```

   f. For Item 2, enter the Name:

      ```
      P7_PREV_PAGE
      ```

   g. For Item 2, enter the Value:

      ```
      1
      ```

3. Click **Apply Changes**.

To select layout and pagination attributes:

1. Under Layout and Pagination:
   a. For Pagination Scheme, select **Search Engine 1,2,3,4 (set based pagination)**.
   b. For Number of Rows, enter `5`.
2. Under Sorting, select the light gray arrow for Ascending and Descending Image.

3. Under Messages, enter the following in When No Data Found Message:

```
No Overdue Issues.
```

4. Click **Apply Changes**.

Add an Unassigned Issues Report

The next report you add displays unassigned, open issues. This report is very similar to Overdue Issues. Rather than creating it manually, you can copy the Overdue Issues report and modify it.

To create a new report by copying an existing report:

1. Under Regions, click the **Copy** icon.
2. In the Name column, click **Overdue Issues**.
3. For To Page, accept the default **1**, accept the remaining defaults, and click **Next**.
4. For Region Name, enter `Unassigned Issues`.
5. Click **Copy Region**.

To modify the query and edit the report region:

1. Under the Regions section, click **Unassigned Issues**.
2. For Region Source, replace the existing statements with the following:

```
SELECT i.issue_id,
       i.priority,
       i.issue_summary,
       i.target_resolution_date,
       r.project_name,
       p.person_name identifiee
  FROM ht_issues i,
       ht_people p,
       ht_projects r
 WHERE i.assigned_to IS NULL
   AND i.status != 'Closed'
   AND i.related_project = r.project_id
   AND i.identified_by = p.person_id
```

3. Select the **Report Attributes** tab at the top of the page.

   Note that previously defined columns have retained their modified attributes.

4. For IDENTIFIEE, enter the following Heading:

```
Identified By
```

5. Under Messages, enter the following in When No Data Found Message:

```
No Unassigned Issues.
```

6. Click **Apply Changes**.

Add a Recently Opened Issues Report

Lastly, you add a report of recently opened issues. The underlying query displays the five most recently opened issues.

To create a report of recently opened issues by copying an existing report:

1. Under Regions, click the **Copy** icon.

2. Under Name, select **Unassigned Issues**.
3. For To Page, accept the default **1**, accept the remaining defaults, and click **Next**.
4. For Region Name, enter `Recently Opened Issues`.
5. Click **Copy Region**.

To modify the query and edit the report region:

1. Under Regions, click **Report** next to Recently Opened Issues.
2. For all columns:
   a. Disable sorting by deselecting **Sort**.
   b. Set Sequence to **-**.
3. Select the **Region Definition** tab at the top of the page.
4. For Region Source, replace the existing statements with the following:

```
SELECT issue_id,
       priority,
       issue_summary,
       assignee,
       target_resolution_date,
       project_name,
       identifiee
FROM
(
SELECT i.issue_id,
       i.priority,
       i.issue_summary,
       p.person_name assignee,
       i.target_resolution_date,
       r.project_name,
       p2.person_name identifiee
  FROM ht_issues i,
       ht_people p,
       ht_people p2,
       ht_projects r
 WHERE i.assigned_to = p.person_id (+)
   AND i.related_project = r.project_id
   AND i.identified_by = p2.person_id
   AND i.created_date > (sysdate - 7)
 ORDER BY i.created_date desc
)
 WHERE rownum < 6
```

5. Select the **Report Attributes** tab at the top of the page.
6. For ASSIGNEE, click the up arrow to the right of the Sort Sequence column until ASSIGNEE appears after ISSUE_SUMMARY.
7. For ASSIGNEE, change Heading to:

```
Assigned To
```

8. Scroll down to the Layout and Pagination section. From Pagination Scheme, select **- No Pagination Selected -**.
9. Under Messages, enter the following in When No Data Found Message:

```
No Recently Opened Issues.
```

10. Click **Apply Changes**.

Add an Open Issues by Project Pie Chart

Next, add a pie chart displaying Open Issues by Project.

To add a pie chart:

1. Under Regions, click the **Create** icon.
2. Select **Chart** and click **Next**.
3. Select **SVG Chart** and click **Next**.
4. Select **Pie** and click **Next**.
5. For Display Attributes, enter `Open Issues by Project` in Title and then click **Next**.
6. For Source, enter the following in SQL:

```
SELECT NULL LINK,
       NVL(r.project_name,'No Project') label,
       COUNT(r.project_name) value
  FROM ht_issues i,
       ht_projects r
 WHERE i.status = 'Open'
   AND i.related_project = r.project_id
GROUP BY NULL, r.project_name
ORDER BY r.project_name
```

   Note that this query does not include a link, the label is the `Project Name`, and the value calculated and used for the pie chart is the total number of open issues by project.

7. Click **Create Region**.

To edit the chart.

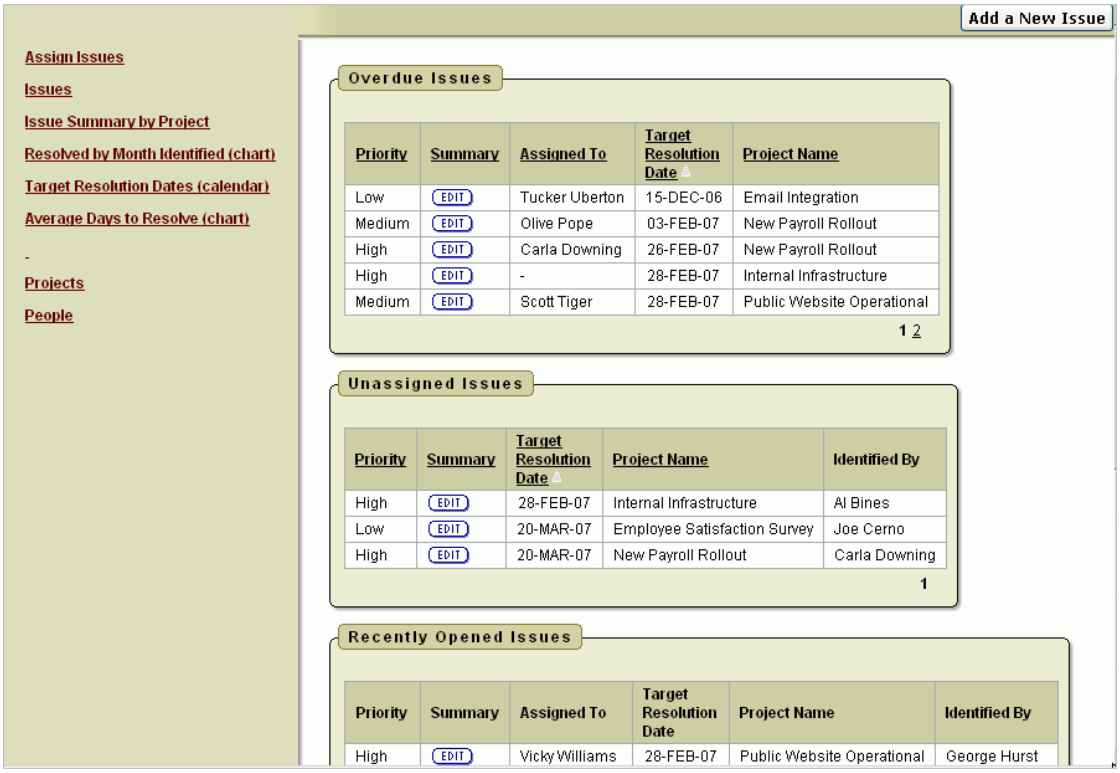
1. Under Regions, click **SVG Chart** next to Open Issues by Project.
2. For Chart Width, enter `500`.
3. For Chart Height, enter `200`.
4. For Chart Title, remove the title.
5. Under Chart Query, enter the following in When No Data Found Message:

```
No Open Issues.
```

6. Under Font Settings, for Legend, select **14** for the Font Size.
7. Click **Apply Changes**.

Run the Page To view the revised page, click the **Run Page** icon. Your home page should resemble Figure 14–16.

Figure 14–16 Revised Home Page



Add a Breadcrumb Menu

In the previous exercise, you created menus on the home page to enable users to navigate to various pages within your application. Next, you need to provide users with a way to navigate to the home page. You can accomplish this by utilizing breadcrumbs. You can add breadcrumbs when you create a new page, or manually after you create your pages.

In the next exercise, you add breadcrumb entries and then include that breadcrumb within a region on page 0. Adding components to page 0 makes them display on all pages with an application.

Topics in this section include:

- Navigate to the Breadcrumbs Page
- Add Breadcrumb Entries
- Create a Page 0
- Create a Region to Contain the Breadcrumb

Navigate to the Breadcrumbs Page

To navigate to the Breadcrumbs page:

1. Go to the Application home page.
2. Click **Shared Components**.
3. Under Navigation, click **Breadcrumbs**.
4. Click the **Breadcrumb** icon.

The Breadcrumb Entries page appears.

Add Breadcrumb Entries

Next, you edit the existing breadcrumb entry and add breadcrumb entries for other pages.

Edit Breadcrumb Entry for Page 1 To edit the breadcrumb entry for page 1:

1. Under Hierarchical View, click the **Home** link.
2. Under Breadcrumb, enter `1` in Page.
3. Under Entry, enter `Home` for Short Name.
4. Under Target, enter `1` in Page.
5. Click **Apply Changes**.

Create Breadcrumb Entry for Page 2 To add a breadcrumb entry for page 2:

1. Click **Create Breadcrumb Entry**.
2. Under Breadcrumb, enter `2` in Page.
3. Under Entry:
   a. Parent Entry - Select **Home**.
   b. Short Name - Enter `Projects`.
4. Under Target, enter `2` in Page.
5. Click **Create**.

Create Breadcrumb Entry for Page 3 To create a breadcrumb entry for page 3:

1. Click **Create Breadcrumb Entry**.
2. Under Breadcrumb, enter `3` in Page.
3. Under Entry:
   a. Parent Entry - Select **Projects**.
   b. For Short Name - Enter `Create/Edit Projects`.
4. Under Target, enter `3` in Page.
5. Click **Create**.

Create Breadcrumb Entry for Page 4 To create a breadcrumb entry for page 4:

1. Click **Create Breadcrumb Entry**.
2. Under Breadcrumb, enter `4` in Page.
3. Under Entry:
   a. For Parent Entry - Select **Home**.
   b. For Short Name - Enter `People`.
4. Under Target, enter 4 in Page.
5. Click **Create**.

   Note that the People breadcrumb entry is a sibling to Projects.

Create Breadcrumb Entry for Page 5 To create a breadcrumb entry for page 5:

1. Click **Create Breadcrumb Entry**.
2. Under Breadcrumb, enter `5` in Page.
3. Under Entry:
   a. For Parent Entry - Select **People**.
   b. For Short Name - Enter `Create/Edit Person Information`.
4. Under Target, enter 5 in Page.
5. Click **Create**.

Create Breadcrumb Entry for Page 6 To create a breadcrumb entry for page 6:

1. Click **Create Breadcrumb Entry**.
2. Under Breadcrumb, enter `6` in Page.
3. Under Entry:
   a. For Parent Entry - Select **Home**.
   b. For Short Name - Enter `Issues`.
4. Under Target, enter 6 in Page.
5. Click **Create**.

Create Breadcrumb Entry for Page 7 To create a breadcrumb entry for page 7:

1. Click **Create Breadcrumb Entry**.
2. Under Breadcrumb, enter `7` in Page.
3. Under Entry:
   a. For Parent Entry - Select **Home**.
   b. For Short Name - Enter `Create/Edit Issue`.
4. Under Target, enter 7 in Page.
5. Click **Create**.

Create Breadcrumb Entry for Page 8 To create a breadcrumb entry for page 8:

1. Click **Create Breadcrumb Entry**.
2. Under Breadcrumb, enter `8` in Page.
3. Under Entry:
   a. For Parent Entry - Select **Home**.
   b. For Short Name - Enter `Assign Issues`.
4. Under Target, enter `8` in Page.
5. Click **Create**.

Create Breadcrumb Entry for Page 9 To create a breadcrumb entry for page 9:

1. Click **Create Breadcrumb Entry**.
2. Under Breadcrumb, enter `9` in Page.
3. For Entry:

   a. For Parent Entry - Select **Home**.
   b. For Short Name - Enter `Issue Summary by Project`.
4. Under Target, enter `9` in Page.
5. Click **Create**.

Create Breadcrumb Entry for Page 10 To create a breadcrumb entry for page 10:

1. Click **Create Breadcrumb Entry**.
2. Under Breadcrumb, enter `10` in Page.
3. Under Menu Option:
   a. For Parent Entry- Select **Home**.
   b. For Short Name - Enter `Resolved by Month Identified`.
4. Under Target, enter `10` in Page.
5. Click **Create**.

Create Breadcrumb Entry for Page 11 To create a breadcrumb entry option for page 11:

1. Click **Create Breadcrumb Entry**.
2. Under Breadcrumb, enter `11` in Page.
3. Under Entry:
   a. For Parent Entry - Select **Home**.
   b. For Short Name - Enter `Target Resolution Dates`.
4. Under Target, enter `11` in Page.
5. Click **Create**.

Create Breadcrumb Entry for Page 12 To create a breadcrumb entry for page 12:

1. Click **Create Breadcrumb Entry**.
2. Under Breadcrumb, enter 12 in Page.
3. Under Entry:
   a. For Parent Entry - Select **Home**.
   b. For Short Name - Enter `Average Days to Resolve`.
4. Under Target, enter `12` in Page.
5. Click **Create**.

Create a Page 0

Now that the breadcrumb exists, you need to create page 0 and then create a region to contain your Breadcrumb menu. Adding components to page 0 makes them display on all pages with an application.

To create a page 0:

1. Go to the Application home page.
2. Click **Create Page**.
3. Select **Blank Page** and click **Next**.

4. For Page Attributes, enter 0 for Page Number and click **Next**.
5. For Page Name, enter **Breadcrumbs** for Name and click **Next**.
6. On Tabs, accept the default, **No**, and then click **Next**.
7. Review your selections and click **Finish**.

Create a Region to Contain the Breadcrumb

To create a region to contain your breadcrumb:

1. From the Success page, select **Edit Page**.

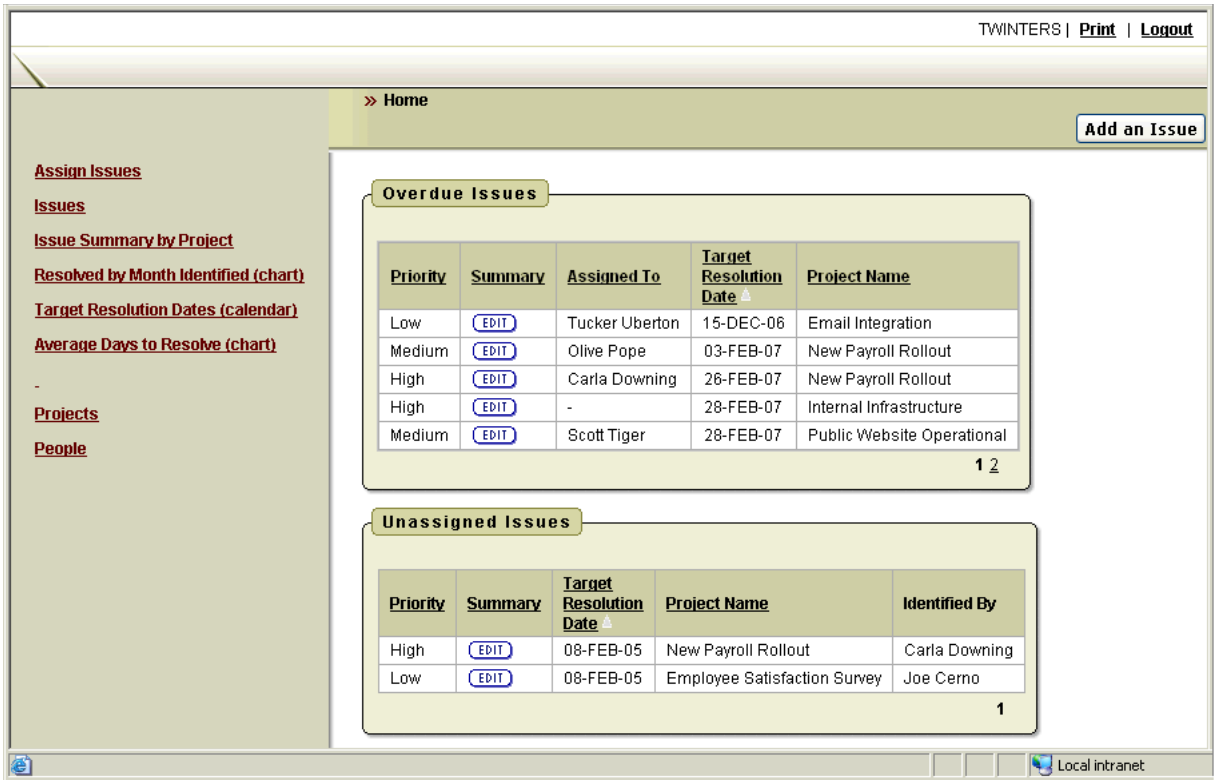
   The Page Definition for page 0 appears.
2. Under Regions, click the **Create** icon.
3. For Identify the type of region to add to this page, select **Breadcrumb** and click **Next**.
4. For Breadcrumb Container Region:
   a. Region Title - Enter **Breadcrumbs**.
   b. Region Template - Select **No Template**.
   c. Display Point - Select **Page Template Region Position 1**.

      This selection displays the breadcrumb above all other content on the page.
   d. Click **Next**.
5. For Breadcrumb, accept the defaults and click **Next**.
6. For Breadcrumb Entry, accept the defaults and click **Next**.
7. Click **Finish**.

Return to the home page by clicking **Edit Page**. When the Page Definition for page 0 appears, click the Next Page button (>). The Page Definition for page 1 appears.

Run the Page To see your completed home page, click the **Run Page** icon. Your home page should resemble Figure 14–17 on page 14-73.

Figure 14–17 Revised Home Page with Breadcrumb Menu



Notice the Breadcrumb in the top bar. Click one of the items on the Maintenance menu on the left side of the page. Notice how the breadcrumb changes.

At this stage your application is fully functional, but is missing the security and email notification. Those topics are discussed in the next section.

Adding Advanced Features

Once your application is fully functional you can focus on adding advanced features outlined during the planning and project analysis phase.

Topics in this section include:

- Add Support for Email Notification
- Add Application Security

Add Support for Email Notification

The planning and project analysis phase produced two email requirements:

- Notify people when an issue is assigned to them
- Notify the project lead when any issue becomes overdue

Topics in this section include:

- How Email Notification Works
- Add Notification of New Assignments
- Add Notification for Overdue Issues

How Email Notification Works

To send mail from within an Oracle Application Express application, you create a PL/SQL process that calls the supplied APEX_MAIL package.

Email is not sent immediately, but is stored in a temporary queue until a DBMS_JOB pushes the queue. The DBMS_JOB utilizes two preferences, SMTP Host Address and SMTP Host Port, to send mail in the queue. By default, these preferences are set to localhost and 25. If Oracle Application Express is not configured for SMTP services, you need to change your Email Environment Settings.

> **See Also:** "How to Send Email from an Application" in *Oracle Application Express Application Builder User's Guide* to learn about configuring Email Environment settings.

The following is a description of the SEND procedure of the APEX_MAIL package.

```
PROCEDURE SEND
Argument Name                  Type                    In/Out Default?
------------------------------ ----------------------- ------ --------
P_TO                           VARCHAR2                IN
P_FROM                         VARCHAR2                IN
P_BODY                         VARCHAR2                IN
P_BODY_HTML                    VARCHAR2                IN     DEFAULT
P_SUBJ                         VARCHAR2                IN     DEFAULT
P_CC                           VARCHAR2                IN     DEFAULT
P_BCC                          VARCHAR2                IN     DEFAULT
```

Add Notification of New Assignments

First, you add a notification to a person when the person has a new assignment. An assignment can be made or changed from two different pages: Create/Edit Issue and Assign Issues.

On the Create/Edit Issue page, you can store the initial values and then check them against any changes to see if an assignment has been made or changed. The Assign Issues is a tabular form, so there is no way to check the old values against the new values. For that reason, the best way to implement the notification is with a before insert and update trigger on HT_ISSUES. You can create this trigger programmatically using SQL Workshop.

To create a before insert and update trigger on HT_ISSUES:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.
2. Click **Create**.
3. For Select the type of database object you want to create, click **Trigger**.
4. For Table Name, select **HT_ISSUES** and click **Next**.
5. For Define:
   a. For Trigger Name, enter BIU_HT_ISSUES_NOTIFY_ASSIGNEE.
   b. For Firing Point, select **AFTER**.
   c. For Options, select **insert, update**.
   d. For Trigger Body, enter the following:

```
IF (INSERTING AND
    :new.assigned_to IS NOT NULL)
   OR
   (UPDATING AND
```

```
 (:old.assigned_to IS NULL OR
  :new.assigned_to != :old.assigned_to) AND
:new.assigned_to IS NOT NULL)
THEN
   FOR c1 IN
      (SELECT person_name, person_email
         FROM ht_people
        WHERE person_id = :new.assigned_to)
   LOOP

   IF c1.person_email IS NOT NULL
      THEN
         FOR c2 IN
             (SELECT project_name
                FROM ht_projects
               WHERE project_id = :new.related_project)
         LOOP

        -- APEX_MAIL.SEND(
        --    p_to => c1.person_email,
        --    p_from => c1.person_email,
        --    p_body =>
        --       'You have been assigned a new issue.
        --       'The details are below. ' ||chr(10)||
        --        chr(10)||
        --       ' Project: '|| c2.project_name ||chr(10)||
        --       ' Summary: '||:new.issue_summary ||chr(10)||
        --       ' Status: '||:new.status ||chr(10)||
        --       'Priority: '||nvl(:new.priority,'-'),
        --   p_subj => 'New Issue Assignment');
         END LOOP;
   END IF;

  END LOOP;
END IF;
```

e. To make this work within your environment, uncomment the APEX_MAIL.SEND and replace the `p_to` and `p_from` with your own valid email address.

f. Click **Next**.

6. To review the code, expand the **SQL** arrow.

7. Click **Finish**.

Add Notification for Overdue Issues

The second email notification notifies the project lead whenever an issue becomes overdue. An issue becomes overdue when the target resolution date has passed, but the issue is not yet closed. There is no human interaction to determine if an issue is overdue, so you cannot check for it on a page or in a trigger.

The best way to check for overdue issues is to write a package that queries the `HT_ISSUES` table. If it finds any overdue issues, the package initiates an email to the Project Lead. This procedure checks for issues by project so that the project lead can receive just one email with all overdue issues rather than an email for each issue. The package will be called once a day by a `dbms_job`.

You can use the Create Object function as follows:

- Create the package and package body from within the SQL Workshop

- Use SQL Command Processor to run the create commands

To create the package:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Commands**.

   SQL Commands appears.

2. Enter the following in the field provided:

```
CREATE OR REPLACE package ht_check_overdue_issues
AS
   PROCEDURE email_overdue;
END;
/
```

3. Click **Run**.

To create the package body:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Commands**.

   SQL Commands appears.

2. Enter the following in the field provided:

```
CREATE OR REPLACE PACKAGE BODY ht_check_overdue_issues
AS

PROCEDURE email_overdue
IS
   l_msg_body varchar2(32000) := null;
   l_count number             := 0;
BEGIN

FOR c1 IN
   (SELECT pr.project_id,
           pr.project_name,
           pe.person_name,
           pe.person_email
      FROM ht_projects pr,
           ht_people pe
     WHERE pr.project_id = pe.assigned_project
       AND pe.person_role = 'Lead')
LOOP
   FOR c2 IN
      (SELECT i.target_resolution_date,
              i.issue_summary,
              p.person_name,
              i.status,
              i.priority
         FROM ht_issues i,
              ht_people p
        WHERE i.assigned_to = p.person_id (+)
          AND i.related_project = c1.project_id
          AND i.target_resolution_date < SYSDATE
          AND i.status != 'Closed'
     ORDER BY i.target_resolution_date, i.issue_summary)
LOOP
   IF l_count = 0
      THEN
         l_msg_body :=
           'As of today, the following issues '||
           'are overdue:'||chr(10)||
```

```
            chr(10)||
           ' Project: '|| c1.project_name ||chr(10)||
            chr(10)||
          '     Target: '||c2.target_resolution_date ||chr(10)||
          '    Summary: '||c2.issue_summary ||chr(10)||
          ' Status:     '||c2.status ||chr(10)||
          ' Priority:   '||c2.priority ||chr(10)||
          'Assigned to: '||c2.person_name;
 ELSE
        l_msg_body := l_msg_body ||chr(10)||
            chr(10)||
         '     Target: '||c2.target_resolution_date ||chr(10)||
         '    Summary: '||c2.issue_summary ||chr(10)||
         '     Status: '||c2.status ||chr(10)||
         ' Priority:   '||c2.priority ||chr(10)||
         'Assigned to: '||c2.person_name;
END IF;
l_count := l_count + 1;
END LOOP;

 IF l_msg_body IS NOT NULL
   THEN
   -- APEX_MAIL.SEND(
   --    p_to => c1.person_email,
   --    p_from => c1.person_email,
   --    p_body => l_msg_body,
   --   p_subj => 'Overdue Issues for Project '||
                 c1.project_name);
 END IF;
 l_count := 0;

END LOOP;

END email_overdue;

END ht_check_overdue_issues;
/
```

To make this work within your environment, uncomment the `APEX_MAIL.SEND` and replace the `p_to` and `p_from` with your own valid email address.

3. Click **Run**.

   Next, you want to update the demonstration data to include your employees' valid email addresses.

To update demonstration data to include valid email addresses:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.
2. From the Object list on the left side of the page, select **Tables**.
3. Select the **HT_PEOPLE** table.
4. Select the **Data** tab.
5. For each person, edit the email address:
   a. Click the **Edit** icon.
   b. Change Person Email to a valid email address.
   c. Click **Apply Changes**.

6. Repeat step 5 for all people within the HT_PEOPLE table.
7. Return to the Workspace home page by clicking the **Home** breadcrumb link.

   Next, you want to create a DBMS_JOB that executes your newly created package at a time interval you specify.

To create the DBMS_JOB:

The following is an example of a DBMS_JOB that executes your newly created package. To make this a valid DBMS_JOB, however, you need to set the interval appropriately and execute it using SQL Commands within the SQL Workshop.

```
DECLARE
   jobno number;
BEGIN
   DBMS_JOB.SUBMIT(
      job => jobno,
     what => 'BEGIN
              ht_check_overdue_issues.email_overdue;
              END;',
next_date => SYSDATE,
 interval => desired_interval
             );
   COMMIT;
END;
/
```

For this DBMS_JOB, replace *desired_interval* with the appropriate interval. For example, to have this job execute once each day, you would replace *desired_interval* with the following:

```
'TRUNC(SYSDATE)+(25/24)'
```

> **See Also:** *Send email from Application Express applications* How To on OTN at:
>
> http://www.oracle.com/technology/products/database/application_express/howtos/index.html

Add Application Security

The planning and project analysis phase produced two security requirements:

- Only the CEO and Managers can define and maintain projects and people
- Once assigned, only the person assigned or a project lead can change data about the issue

Within Oracle Application Express, you can define authorization schemes. Authorization controls user access to specific controls or components based on user privileges. Once defined, you can associate an authorization scheme with any page, region, or item to restrict access. Each authorization schema is run only when needed and is defined to validate either once for each page view or once for each session.

Topics in this section include:

- Restrict Project and People Definition
- Restrict Issue Modification

Restrict Project and People Definition

The first requirement states that only the CEO and Managers may define and maintain projects and people. To address this requirement, you:

- Create an authorization scheme to check the current user's role
- Associate the authorization scheme with the items on the Projects and People report that navigate to the Create/Edit pages
- Associate the authorization scheme with the Create/Edit pages themselves so that a user cannot bypass the security by manually editing the URL to the target page

To reference the current user, use the session variable `:APP_USER`. This session variable is compared with the person's email address (which is the same as their workspace or workspace name). Whenever coding this type of security, you should always code in a user that can pass all security. You may find this user very useful for development and testing. If you do not take this approach, you may not be able to access the restricted pages unless you define yourself as the CEO or Manager.

Create the Authorization Scheme

To create the authorization scheme:

1. On the Workspace home page, click **Application Builder**.
2. Select the **Issue Tracker** application.
3. Click **Shared Components**.
4. Under Security, click **Authorization Schemes**.
5. Click **Create**.
6. For Create Authorization Scheme, accept the default, **From Scratch**, and click **Next**.
7. Under Authorization Scheme, enter the following in Name:

   ```
   USER_CEO_OR_MANAGER
   ```

8. Under Authorization Scheme:
   a. Scheme Type - Select **Exists SQL Query**.
   b. Expression 1 - Enter:

      ```
      SELECT '1'
        FROM ht_people
       WHERE (upper(person_email) = UPPER(:APP_USER) AND
              person_role IN ('CEO','Manager'))
          OR (UPPER(:APP_USER) = 'HOWTO')
      ```

   c. Identify error message displayed when scheme violated - Enter:

      ```
      You are not authorized to access this function.
      ```

9. Scroll down to Evaluation Point. For Validate authorization scheme, select **Once per session**.

   This selection is sufficient in this instance as the assigned role typically does not change within a given session.

10. Click **Create**.

Next, you need to associate the authorization scheme with the appropriate objects.

Associate Objects on the Projects Report

To associate the authorization scheme with the Projects report:

1. Click the **Edit Page** icon in the upper right corner. The Edit Page icon resembles a small green piece of paper and pencil.
2. In Page, enter `2` and then click **Go**.

   The Page Definition for page 2, Projects, appears.
3. Under Regions, click **Report** next to Projects.
4. Click the **Edit** icon to the left of `PROJECT_ID`.
5. Under Authorization, select the Authorization Scheme **USER_CEO_OR_MANAGER**.
6. Click **Apply Changes**.
7. Click **Cancel**.

To associate the authorization scheme with the Create button on the Projects report:

1. Under Buttons on the Page Definition for page 2, click the **Create** link (not the icon).

   The Edit Page Buttons page appears.
2. Under Authorization, select the Authorization Scheme **USER_CEO_OR_MANAGER**.
3. Click **Apply Changes**.

Associate Objects with the Create/Edit Report

To associate the authorization scheme with the Create/Edit Project page:

1. Go to page 3 by clicking the Next Page (>) button.

   The Page Definition for page 3, Create/Edit Project, appears.
2. Under Page, click the **Edit page attributes** icon.

   The Page attributes page appears.
3. Under Security, select the Authorization Scheme **USER_CEO_OR_MANAGER**.
4. Click **Apply Changes**.

Associate Objects with the People Report

To associate the authorization scheme with the People report.

1. Go to page 4 by clicking the Next Page (>) button.

   The Page Definition for page 4, People, appears.
2. Under Regions, click **Report** next to People.
3. Click the **Edit** icon to the left of PERSON_ID.
4. Under Authorization, select the Authorization Scheme **USER_CEO_OR_MANAGER**.
5. Click **Apply Changes**.
6. Click **Cancel**.

To associate the authorization scheme with the Create button on the People report:

1. Go to page 5 by clicking the Next Page (>) button.

   The Page Definition for page 5 appears.

2. Under Buttons, click the **Create** link (not the icon).

   The Edit Page Buttons page appears.

3. Under Authorization, select the Authorization Scheme **USER_CEO_OR_MANAGER**.

4. Click **Apply Changes**.

To associate the authorization scheme with the Create/Edit Person Information page:

1. Under Page, click the **Edit page attributes** icon.

   The Page attributes page appears.

2. Under Security, select the Authorization Scheme **USER_CEO_OR_MANAGER**.

3. Click **Apply Changes**.

You can test this by creating a user with the user name of HOWTO. The HOWTO user should be able to see the edit link. Then, create another user, HOWTO2. This user should not be able to see the link.

See Also: "Create Users" on page 14-88

Restrict Issue Modification

The second requirement states that once an issue has been assigned, only the person assigned (or a project lead) can change data about the issue. This requirement is a little trickier since it changes for every issue.

Currently, there are two pages that enable users to modify an issue: the Create/Edit Issue page and the Assign Issues page. On the Assign Issues page, the only issues that are displayed are those that are unassigned. As the issues are unassigned, security is not necessary.

There are many places that a user can navigate to edit an issue:

- Three regions on the home page display issues or have edit links
- The Issues report has links to edit each issue
- The Target Resolution Dates report enables users to select an issue to edit.

Although other users are not allowed to change the data, you do want to enable users to view all the detailed data about an issue so that they can view the progress and resolution. Given this requirement, the best approach is to create an authorization scheme to be evaluated once for each page view.

The authorization scheme will be associated with both the Apply Changes and Delete buttons on the Create/Edit Issue page. This way, unauthorized users can view all the details, but if they do change something, they have no way of saving that change.

For added security, you can also associate the authorization scheme with the process that performs the insert, update and delete on `HT_ISSUES`. This protects your application against someone changing the URL to call the Apply Changes process. To let users know why they are not able to make changes, you can add an HTML region that displays an explanation when the authorization fails. The SQL for this scheme must be specific to the Create/Edit Issues page because it needs to reference P7_ISSUE_ID. It also needs to retrieve data from the database because at the time it is

evaluated, the necessary data will not be available in the session state. The only item that will be available will be P7_ISSUE_ID because it will be passed by the link.

Create the Authorization Scheme

To create the authorization scheme:

1. Go to the Application home page.
2. Click **Shared Components**.
3. Under Security, click **Authorization Schemes**.
4. Click **Create**.
5. For Creation Method, accept the default **From Scratch** and then click **Next**.
6. Under Authorization Scheme, enter the following in Name:

   ```
   P7_ASSIGNED_OR_PROJECT_LEAD
   ```

7. Under Authorization Scheme:
   a. For Scheme Type, select **PL/SQL Function Returning Boolean**.
   b. For Expression 1, enter:

   ```
   DECLARE
      l_related_project   integer;
      l_assigned_to       integer;
      l_person_id         integer;
      l_person_role       varchar2(7);
      l_assigned_project  integer;
   BEGIN

   -- User is HOWTO or new Issue
   IF :APP_USER = 'HOWTO' or
      :P7_ISSUE_ID IS NULL
      THEN RETURN TRUE;
   END IF;

   FOR c1 IN (SELECT related_project,
                     assigned_to
                FROM ht_issues
               WHERE issue_id = :P7_ISSUE_ID)
   LOOP
      l_related_project := c1.related_project;
      l_assigned_to     := c1.assigned_to;
   END LOOP;

   -- Issue not yet assigned
   IF l_assigned_to IS NULL
   THEN RETURN TRUE;
   END IF;

   FOR c2 IN (SELECT person_id,
              person_role,
              assigned_project
         FROM ht_people
        WHERE upper(person_email) = upper(:APP_USER))
   LOOP
      l_person_id        := c2.person_id;
      l_person_role      := c2.person_role;
      l_assigned_project := c2.assigned_project;
   ```

```
END LOOP;

-- User is lead of related project
IF l_person_role = 'Lead' and
   l_assigned_project = l_related_project
   THEN RETURN TRUE;

-- User is assigned to issue
ELSIF l_assigned_to = l_person_id
   THEN RETURN TRUE;
ELSE
   RETURN FALSE;
END IF;
END;
```

c. For Identify error message displayed when scheme violated, enter:

```
This issue is not assigned to you, nor are you the Project Lead. Therefore
you are not authorized to modify the data.
```

8. Under Evaluation Point, for Validate authorization scheme, select **Once per page view**.

   This selection is necessary since each issue may have a different result.

9. Click **Create**.

Now you need to associate the authorization scheme with the appropriate objects on the Create/Edit Issue page.

Associate Objects with the Create Edit Issues Report

To associate the authorization scheme with buttons and processes:

1. Go to the Application home page.
2. Select page **7 - Create/Edit Issues**.
3. Under Buttons, click **Delete**.
   a. Under Authorization, select the Authorization Scheme **P7_ASSIGNED_OR_PROJECT_LEAD**.
   b. Click **Apply Changes**.
4. Under Buttons, click **Apply Changes**.
   a. Under Authorization, select the Authorization Scheme **P7_ASSIGNED_OR_PROJECT_LEAD**.
   b. Click **Apply Changes**.
5. Under Buttons, click **Create**.
   a. Under Authorization, select the Authorization Scheme **P7_ASSIGNED_OR_PROJECT_LEAD**.
   b. Click **Apply Changes**.
6. Under Buttons, click **Create and Create Another**.
   a. Under Authorization, select the Authorization Scheme **P7_ASSIGNED_OR_PROJECT_LEAD**.
   b. Click **Apply Changes**.
7. Under Page Processing, Processes, select **Process Row of HT_ISSUES**.

a. Under Authorization, select the Authorization Scheme **P7_ASSIGNED_OR_PROJECT_LEAD**.

b. Click **Apply Changes**.

Create an HTML Region

Lastly, create a new region to display an explanation when the authorization fails

To create a new region:

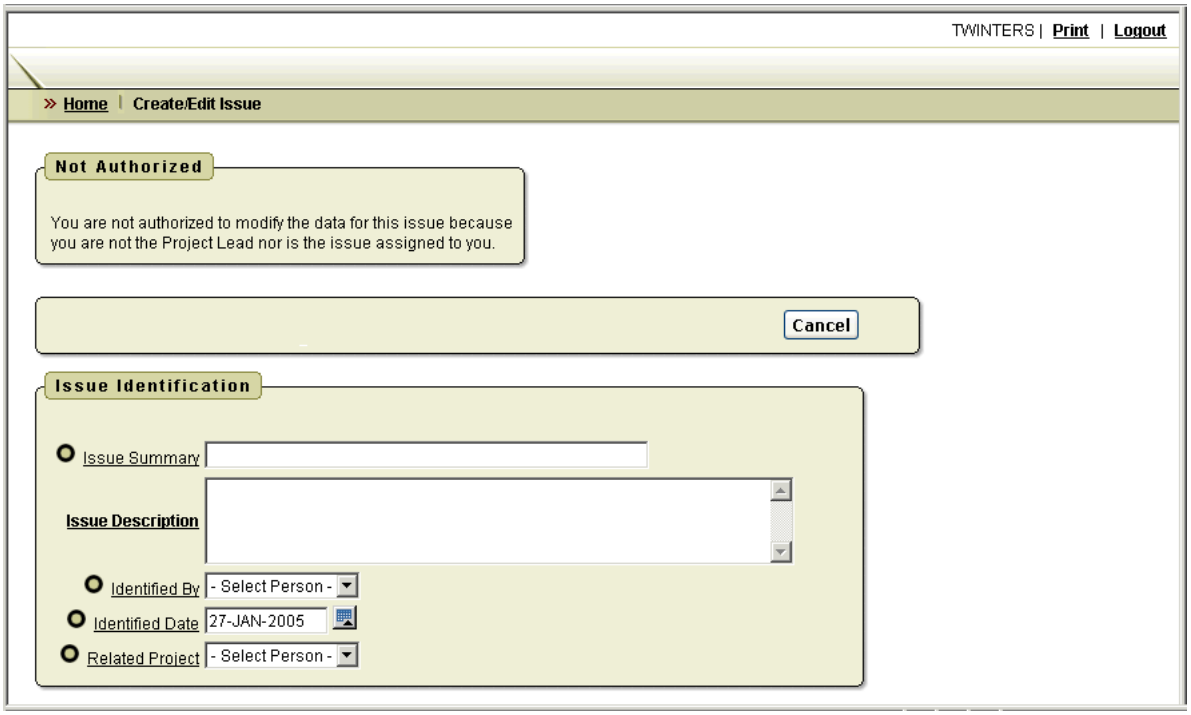
1. Under Regions, click the **Create** icon.
2. On Region, accept the default **HTML** and click **Next**.
3. Select **HTML** for region container and click **Next**.
4. For Display Attributes:
   a. For Title, enter `Not Authorized`.
   b. For Display Point, select **Page Template Body (2. items below region content)**.
   c. Click **Next**.
5. For Source, enter the following in Enter HTML Text Region Source and then click **Next**:

```
You are not authorized to modify the data for this issue because<br>you are not
the Project Lead nor is the issue assigned to you.
```

6. For Authorization Scheme, select **{Not P7_ASSIGNED_OR_PROJECT_LEAD}**. This selection makes the region only display when the Authorization Scheme fails.
7. Click **Create Region**.

Figure 14–18 on page 14-85 displays the Create/Edit Issue page being run by a person for whom the Authorization fails. Notice a new region displays at the top of the page and that the only button being displayed is Cancel.

Figure 14–18 New Region Displaying Authorization Failure



A more elegant solution to this security requirement would be to create a different page for viewing the details of an issue. You would need to have a procedure that would take in the issue_id and current user and pass back a flag for view only or edit. Then you could dynamically build the link for all the reports to call either the View page or the Edit page based upon a call to that procedure. You would still want to protect against someone accessing the edit page without using a link so you would also check permission before firing the insert, update and delete process.

Deploying Your Application

Now that your application is complete, the next step is to deploy it. Typically, developers create applications on one server and deploy it on another. Although this approach is not required, it enables you to resolve bugs without impacting the production instance.

> **Note:** To deploy an application on another server, you need to install and configure another Oracle Application Express instance.

Topics in this section include:

- Move the Application Definition
- Alternate Authentication Mechanisms to Consider
- Create Users
- Publish the URL

Move the Application Definition

The definition for your application lives within the Oracle database. The application definition includes everything that makes up the application, including the templates, but it does not include database object definitions or the underlying data. To move an application to another Oracle Application Express instance, you must export the application definition from your development server and import it into your production server.

Topics in this section include:

- Export the Application Definition
- Create the Required Objects to Support the Application
- Import the Application Definition into the Production Instance
- Load the Data

Export the Application Definition

To export the application definition from your development server:

1. On the Workspace home page, click the arrow on the **Application Builder** icon and select the application you just created.
2. Click the **Export/Import** icon.
3. For Export/Import, click **Export** and then **Next**.
4. For Application, make sure the application created in this exercise is selected.
5. Click **Export Application**.
6. When prompted, click to **Save** the file.
7. Specify a location on your local hard drive and then click **Save**.

Create the Required Objects to Support the Application

On your production instance, you need to create the objects necessary to support the application. Log in to the production instance and follow the directions in "About Building Database Objects" on page 14-6.

Note: Although the supporting objects do not need to exist for you to import the application definition, be aware you cannot test the code until they exist.

Import the Application Definition into the Production Instance

Log in to the production instance of the Workspace home page:

1. On the Workspace home page, click the arrow on the **Application Builder** icon and select the application you just created.
2. On the Application home page, click **Export/Import**.
3. On the Export/Import page, click **Import** and then click **Next**.
4. For Import File:
   a. Import file - Click the **Browse** button and then locate your exported file.
   b. File Type - Select **Application, Page, or Component Export**.
   c. File Character Set - Accept the default and click **Next**.

Once the success message appears, the next step is to install the file.

5. Click **Install**.
6. On Application Install:
   a. Parse As Schema - Select the schema on your production server that contains your application objects.
   b. Build Status, - Select **Run and Build Application**.

      This option enables other users to run the application and enables you to log in and change the code if necessary. Alternatively, you can select **Run Application Only**. Be aware that if you select this option you will not be able to access the source code for the application.
   c. Install As Application - You can select:
      – **Reuse Application ID from Export File** - Only select this option if the application ID is not being used on the production instance.
      – **Auto Assign New Application ID** - Select this option to assign a new application ID.
      – **Change Application ID** - Select this option to change the existing application ID. If you select this option, you will be prompted to enter a new application ID.

        When you install an application having the same ID as an existing application in the current workspace, the existing application is deleted and then the new application is installed. If you attempt to install an application having the same ID as an existing application in a different workspace, an error message appears.

        If all statements are successful the install commits and becomes permanent. If any errors are encountered, the install is rolled back, resulting in no permanent changes.
   d. Click **Install Application**.

   If the install is successful, the Post-App Install Utility Options page appears. From here, you can select one of the following:

   - Select **Run Application** to see the application running
   - Select **Application Attributes** to view the application definition within Application Builder

Load the Data

The next step in deploying your application is to load the data. At a minimum, you would need to populate the `project` and `people` tables.

Note there are various mechanisms you could use to accomplish this task, including:

- Use the application itself to create data.
- Use the Data Loader to load data copied from a spreadsheet.
- Use SQL Scripts and run scripts to create data.
- If you have data existing already within an Oracle database, use either export/import to move data between machines or use SQL to retrieve and transform existing data and load it into the application tables.

> **See Also:** "Loading Demonstration Data" on page 14-8 and "Importing, Exporting, Loading, and Unloading Data" in *Oracle Database Express Edition 2 Day DBA*

Alternate Authentication Mechanisms to Consider

When the application login page calls the login API with a user name and password, the Application Express engine calls the credentials verification method specified in the application's current authentication scheme. You have three choices as to how credentials are verified from within the login API:

- Implement the method yourself as a PL/SQL function returning Boolean and put it in your application's schema.
- Use the built-in LDAP authentication method, which checks user name and password against the LDAP directory that you specify.
- Use the built-in Oracle Application Express authentication method, which checks the user name and password against the Oracle Application Express workspace repository.

Your application is currently using the built-in Oracle Application Express authentication method.

> **See Also:** "Establishing User Identity Through Authentication" in *Oracle Application Express Application Builder User's Guide*

> **See Also:** Security How To documents on OTN:
>
> `http://www.oracle.com/technology/products/database/application_express/howtos/howtos.html`

Create Users

In order for your application to be accessible, you need to create users. If you are still using Oracle Application Express authentication, the simplest way to create users it to access the Manage Users page.

To create a new user:

1. Go to the Workspace home page.
2. From the Administration list on the right side of the page, click **Manage Application Express Users**.
3. From the Tasks list on the right side of the page, click **Create End User**.
4. Under User Identification, enter the required information.
5. Click **Create User** or **Create and Create Another**.

Publish the URL

Now that you have deployed your application, loaded data, and created users, you can publish your production URL.

You can determine the URL to your application by positioning the mouse over the **Run** icon on the Application home page. The URL appears in the status bar at the bottom of the page.

The Run icon gets its value from the Home link attribute on the Edit Security Attributes page. This link is only referenced by this icon and by applications that do not use the Oracle Application Express Login API. Consider the following example:

```
http://apex.oracle.com/pls/otn/f?p=11563:1:3397731373043366363
```

Where:

- `apex.oracle.com` is the URL of the server
- `pls` is the indicator to use the `mod_plsql` cartridge
- `otn` is the data access descriptor (DAD) name
- `f?p=` is a prefix used by Oracle Application Express
- `11563` is the application being called
- `1` is the page within the application to be displayed
- `3397731373043366363` is the session number

To run this example application, you would use the URL:

```
http://apex.oracle.com/pls/otn/f?p=11563:1
```

When users log in, they receive a unique session number.

As you may recall, you created the Issue Tracker application using the Create Application wizard. This wizard creates a process on the Login page (page 101) that controls authentication. The contents of the process are:

```
WWV_FLOW_CUSTOM_AUTH_STD.LOGIN(
    P_UNAME => :P101_USERNAME,
    P_PASSWORD => :P101_PASSWORD,
    P_SESSION_ID => :FLOW_SESSION,
    P_FLOW_PAGE => :APP_ID||':1'
    );
```

Note that the Page is hard coded into this process. Because of this, the page you pass in to the URL is overwritten and does not need to be included. You can access the application by using the following URL:

```
http://apex.oracle.com/pls/otn/f?p=11563:1
```

As you can see from the example used, the URL has no meaning and can be rather long. The host name can be changed to make it more symbolic. You can also configure Apache to rewrite your URL so that you can publish an abbreviated format and a URL that would be more intuitive to your users. See your Apache documentation for details.

A

DDLs and Scripts

This appendix contains DDLs (data definition language) and scripts necessary to complete a number of tutorials in *Oracle Application Express Advanced Tutorials*.

Topics in this section include:

- Creating Application Database Objects DDL
- Creating Issues Script

Creating Application Database Objects DDL

The following DDL creates all the required database objects for the issue tracking application in "How to Build and Deploy an Issue Tracking Application" on page 14-1.

```
--
-- This DDL creates all the database objects used by the
-- Issue Management Application featured in
--
-- HT_PROJECTS
--
-- The HT_PROJECTS DDL:
--    + creates the projects table with the necessary columns,
--      including a new column for a system generated primary key
--    + declares the new primary key
--    + implements the real primary key, project name, as a unique key
--    + implements a sequence to generate project id
--    + assigns the sequence to populate the project id
--      whenever a new record is created
--    + declares table and column comments
--
```

```
CREATE TABLE ht_projects (
   project_id              INTEGER           NOT NULL,
   project_name            VARCHAR2(100)     NOT NULL,
   start_date              DATE              NOT NULL,
   target_end_date         DATE              NOT NULL,
   actual_end_date         DATE

)
/
ALTER table ht_projects
```

```
   ADD CONSTRAINT ht_projects_pk
   PRIMARY KEY (project_id)
/
ALTER TABLE ht_projects
   ADD CONSTRAINT ht_projects_uk
   UNIQUE (project_name)
/
CREATE SEQUENCE ht_projects_seq
   INCREMENT BY 1
   START WITH 40
/
CREATE OR REPLACE TRIGGER bi_ht_projects
     BEFORE INSERT ON ht_projects
     FOR EACH ROW
  BEGIN
     IF :new.project_id is null
        THEN SELECT ht_projects_seq.nextval
               INTO :new.project_id
               FROM DUAL;
     END IF;
  END;
/

COMMENT ON table ht_projects IS
   'All projects currently underway.'
/
COMMENT ON column ht_projects.project_id IS
   'The system generated unique identifier for the project.'
/
COMMENT ON column ht_projects.project_name IS
   'The unique name of the project.'
/
COMMENT ON column ht_projects.start_date IS
   'The start date of the project.'
/
COMMENT ON column ht_projects.target_end_date IS
   'The targeted end date of the project.'
/
COMMENT ON column ht_projects.actual_end_date IS
   'The actual end date of the project.'
/

--
-- HT_PEOPLE
--
-- The HT_PEOPLE DDL:
--    + creates the people table with the necessary columns,
--      including a new column for a system generated primary key
--    + declares the new primary key
--    + implements the real primary key, person name, as a unique key
--    + implements a check constraint to validate the roles that people
--      can be assigned

--    + implements a foreign key to validate that people are assigned to
--      valid projects
--    + implements a check constraint to enforce that all project leads
--      and team members are assigned to projects
--    + implements a sequence to generate person id
--    + assigns the sequence to populate the person id whenever a
--      new record is created
```

```
--    + declares table and column comments
--

CREATE TABLE ht_people (
   person_id              INTEGER          NOT NULL,
   person_name            VARCHAR2(100)    NOT NULL,
   person_email           VARCHAR2(100)    NOT NULL,
   person_role            VARCHAR2(7)      NOT NULL,
   assigned_project       INTEGER
)
/
ALTER TABLE ht_people
   ADD CONSTRAINT ht_people_pk
   PRIMARY KEY (person_id)
/
ALTER TABLE ht_people
   ADD CONSTRAINT ht_people_uk
   UNIQUE (person_name)
/
ALTER TABLE ht_people
   ADD CONSTRAINT ht_people_role_cc
   CHECK (person_role in ('CEO','Manager','Lead','Member'))
/
ALTER TABLE ht_people
   ADD CONSTRAINT ht_people_project_fk
   FOREIGN KEY (assigned_project)
   REFERENCES ht_projects
/
ALTER TABLE ht_people
   ADD CONSTRAINT ht_people_assignment_cc
   CHECK ( (person_role in ('Lead','Member') and assigned_project is not null)
   or (person_role in ('CEO','Manager') and assigned_project is null) )
/
CREATE SEQUENCE ht_people_seq
   INCREMENT BY 1
   START WITH 40
/
CREATE OR REPLACE TRIGGER bi_ht_people
      BEFORE INSERT on ht_people
      FOR EACH ROW
   BEGIN
      IF :new.person_id IS NULL
          THEN SELECT ht_people_seq.nextval
                 INTO :new.person_id
                 FROM DUAL;
      END IF;

   END;
/

COMMENT ON table ht_people IS
   'All people within the company.'
/
COMMENT ON column ht_people.person_id IS
   'The system generated unique identifier for the person.'
/
COMMENT ON column ht_people.person_name IS
   'The unique name of the person.'
/
COMMENT ON column ht_people.person_role IS
```

```
   'The role the person plays within the company.'
/
COMMENT ON column ht_people.assigned_project IS
   'The project that the person is currently assigned to.'
/

--
-- HT_ISSUES
--
-- The HT_ISSUES DDL:
--    + creates the table with the necessary columns, including a new column
--      for a system generated primary key
--    + declares the new primary key
--    + implements a foreign key to validate that the issue is identified by a
--      valid person
--    + implements a foreign key to validate that the issue is assigned to a
--      valid person
--    + implements a foreign key to validate that the issue is associated with
--      a valid project
--    + implements a check constraint to validate the status that is assigned
--      to the issue
--    + implements a check constraint to validate the priority that is assigned
--      to the issue
--    + implements a sequence to generate issue id
--    + assigns the sequence to populate the issue id and the creation date
--      whenever a new record is created, records the user creating the
--      row and also assigns status of 'Open' if no status is provided
--    + records the current date and the user whenever an issue is edited and
--      sets the status to 'Closed' if an ACTUAL_RESOLUTION_DATE is
--      provided
--    + declares table and column comments
--

create table ht_issues (
   issue_id                INTEGER           not null,
   issue_summary           VARCHAR2(200)     not null,
   issue_description       VARCHAR2(2000),
   identified_by           INTEGER NOT NULL,
   identified_date         DATE              not null,
   related_project         INTEGER           not null,
   assigned_to             INTEGER,
   status                  VARCHAR2(8)       not null,
   priority                VARCHAR2(6),
   target_resolution_date  DATE,
   progress                VARCHAR2(2000),
   actual_resolution_date  DATE,
   resolution_summary      VARCHAR2(2000),
   created_date            DATE              not null,
   created_by              VARCHAR2(60)      not null,
   last_modified_date      DATE,
   last_modified_by        VARCHAR2(60)
)
/
ALTER TABLE ht_issues
   ADD CONSTRAINT ht_issues_pk
   PRIMARY KEY (issue_id)
/
ALTER TABLE ht_issues
   ADD CONSTRAINT ht_issues_identified_by_fk
   FOREIGN KEY (identified_by)
```

```
   REFERENCES ht_people
/
ALTER TABLE ht_issues
   ADD CONSTRAINT ht_issues_assigned_to_fk
   FOREIGN KEY (assigned_to)
   REFERENCES ht_people
/
ALTER TABLE ht_issues
   ADD CONSTRAINT ht_issues_project_fk
   FOREIGN KEY (related_project)
   REFERENCES ht_projects
/
ALTER TABLE ht_issues
   ADD CONSTRAINT ht_issues_status_cc
   CHECK (status in ('Open','On-Hold','Closed'))
/
ALTER TABLE ht_issues
   ADD CONSTRAINT ht_issues_priority_cc
   CHECK (priority in ('High','Medium','Low'))

/

CREATE SEQUENCE ht_issues_seq
   INCREMENT BY 1
   START WITH 40
/
CREATE OR REPLACE TRIGGER bi_ht_issues
      BEFORE INSERT on ht_issues
      FOR EACH ROW
  BEGIN
      IF :new.issue_id IS NULL
         THEN SELECT ht_issues_seq.nextval
                INTO :new.issue_id
                FROM DUAL;
      END IF;
      IF :new.status IS NULL
         THEN :new.status := 'Open';
      END IF;
      :new.created_date := sysdate;
      :new.created_by := nvl(wwv_flow.g_user,user);
  END;
/
CREATE OR REPLACE TRIGGER bu_ht_issues
      BEFORE UPDATE ON ht_issues
      FOR EACH ROW
  BEGIN
      IF :new.actual_resolution_date IS NOT NULL
         THEN :new.status := 'Closed';
      END IF;

      :new.last_modified_date := sysdate;
      :new.last_modified_by := nvl(wwv_flow.g_user,user);
  END;
/

COMMENT ON table ht_issues IS
   'All issues related to the projects being undertaken by the company.'
/
COMMENT ON column ht_issues.issue_id IS
   'The system generated unique identifier for the issue.'
```

```
/
COMMENT ON column ht_issues.issue_summary IS
   'A brief summary of the issue.'
/
COMMENT ON column ht_issues.issue_description IS
   'A full description of the issue.'
/
COMMENT ON column ht_issues.identified_by IS
   'The person who identified the issue.'
/
COMMENT ON column ht_issues.identified_date IS
   'The date the issue was identified.'
/
COMMENT ON column ht_issues.related_project IS
   'The project that the issue is related to.'
/
COMMENT ON column ht_issues.assigned_to IS
   'The person that the issue is assigned to.'
/
COMMENT ON column ht_issues.status IS
   'The current status of the issue.'
/
COMMENT ON column ht_issues.priority IS
   'The priority of the issue. How important it is to get resolved.'
/
COMMENT ON column ht_issues.target_resolution_date IS
   'The date on which the issue is planned to be resolved.'
/
COMMENT ON column ht_issues.actual_resolution_date IS
   'The date the issue was actually resolved.'
/
COMMENT ON column ht_issues.progress IS
   'Any progress notes on the issue resolution.'
/
COMMENT ON column ht_issues.resolution_summary IS
   'The description of the resolution of the issue.'
/
COMMENT ON column ht_issues.created_date IS
   'Audit Column: Date the record was created.'
/
COMMENT ON column ht_issues.created_by IS
   'Audit Column: The user who created the record.'
/
COMMENT ON column ht_issues.last_modified_date IS
   'Audit Column: Date the record was last modified.'
/
COMMENT ON column ht_issues.last_modified_by IS
   'Audit Column: The user who last modified the record.'
/
```

Creating Issues Script

The following script populates Issues table for the issue tracking application in "How to Build and Deploy an Issue Tracking Application" on page 14-1.

```
--
-- Email Integration Issues
--
INSERT INTO ht_issues
```

```
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
     (1, 'Midwest call center servers have no failover due to Conn Creek plant
fire','',
     6, sysdate-80,
     3, 6, 'Closed', 'Medium', sysdate-73,
     'Making steady progress.', sysdate-73, '')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
       identified_by, identified_date,
       related_project, assigned_to, status, priority, target_resolution_date,
       progress, actual_resolution_date, resolution_summary)
  VALUES
      (2, 'Timezone ambiguity in some EMEA regions is delaying bulk forwarding to
mirror sites','',
      6, sysdate-100,
      3, 14, 'Open', 'Low', sysdate-80,
      '','','')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
       identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (3, 'Some vendor proposals lack selective archiving and region-keyed
retrieval sections','',
      6, sysdate-110,
      3, 13, 'Closed', 'Medium', sysdate-90,
      '', sysdate-95, '')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (4, 'Client software licenses expire for Bangalore call center before
cutover','',
      1, sysdate-70,
      3, 6, 'Closed', 'High', sysdate-60,
      '',sysdate-66,'Worked with HW, applied patch set.')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (5, 'Holiday coverage for DC1 and DC3 not allowed under union contract, per
acting steward at branch 745','',
      1, sysdate-100,
      3, 13, 'Closed', 'High', sysdate-90,
      '',sysdate-95, 'Worked with HW, applied patch set.')
/
--
```

```
-- Employee Satisfaction Survey Issues
--
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (6, 'Review rollout schedule with HR VPs/Directors','',
      8, sysdate-30,
      5, null, 'Closed', 'Medium', sysdate-15,
      '',sysdate-20,'')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (7, 'Distribute translated categories and questions for non-English regions
to regional team leads','',
      8, sysdate-2,
      5, 8, 'Open', 'Medium', sysdate+10,
      'currently beta testing new look and feel','','')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (8, 'Provide survey FAQs to online newsletter group','',
      1, sysdate-10,
      5, 11, 'Open', 'Medium', sysdate+20,
      '','','')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (9, 'Need better definition of terms like work group, department, and
organization for categories F, H, and M-W','',
      1, sysdate-8,
      5, null, 'Open', 'Low', sysdate+15,
      '','','')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (10, 'Legal has asked for better definitions on healthcare categories for
Canadian provincial regs compliance','',
      1, sysdate-10,
      5, 11, 'Closed', 'Medium', sysdate+20,
      '',sysdate-1,'')
/
```

```
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (11, 'Action plan review dates conflict with effectivity of organizational
consolidations for Great Lakes region','',
      1, sysdate-9,
      5, 11, 'Open', 'Medium', sysdate+45,
      '','','')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (12, 'Survey administration consulting firm requires indemnification release
letter from HR SVP','',
      1, sysdate-30,
      5, 11, 'Closed', 'Low', sysdate-15,
      '', sysdate-17, '')
/
--
-- Internal Infrastructure Issues
--
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
 VALUES
      (13, 'Facilities, Safety health-check reports must be signed off before
capital asset justification can be approved','',
      4, sysdate-145,
      1, 4, 'Closed', 'Medium', sysdate-100,
      '',sysdate-110,'')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (14, 'Cooling and Power requirements exceed 90% headroom limit -- variance
from Corporate requested','',
      4, sysdate-45,
      1, 9, 'Closed', 'High', sysdate-30,
      '',sysdate-35,'')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (15, 'Local regulations prevent Federal contracts compliance on section
3567.106B','',
      4, sysdate-90,
```

```
      1, 10, 'Closed', 'High', sysdate-82,
      '',sysdate-85,'')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (16, 'Emergency Response plan failed county inspector''s review at buildings
2 and 5','',
      4, sysdate-35,
      1, null, 'Open', 'High', sysdate-5,
      '','','')
/
--
-- New Payroll Rollout Issues
--
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (17, 'Training for call center 1st and 2nd lines must be staggered across
shifts','',
      5, sysdate-8,
      2, 5, 'Closed', 'Medium', sysdate+10,
      '',sysdate-1,'')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (18, 'Semi-monthly ISIS feed exceeds bandwidth of Mississauga backup
site','',
      5, sysdate-100,
      2, 12, 'On-Hold', 'Medium', sysdate-30,
      'pending info from supplier','','')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
     (19, 'Expat exception reports must be hand-reconciled until auto-post
phaseout complete','',
      5, sysdate-17,
      2, 12, 'Closed', 'High', sysdate+4,
      '',sysdate-4,'')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
```

```
      (20, 'Multi-region batch trial run schedule and staffing plan due to
directors by end of phase review','',
      5, sysdate,
      2, null, 'Open', 'High', sysdate+15,
      '','','')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (21, 'Auditors'' signoff requires full CSB compliance report','',
      5, sysdate-21,
      2, 5, 'Open', 'High', sysdate-7,
      '','','')
/
--
-- Public Website Operational Issues
--
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (22, 'Review security architecture plan with consultant','',
      1, sysdate-60,
      4, 7, 'Closed', 'High', sysdate-45,
      '',sysdate-40,'')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (23, 'Evaluate vendor load balancing proposals against capital budget','',
      7, sysdate-50,
      4, 7, 'Closed', 'High', sysdate-45,
      '',sysdate-43,'')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (24, 'Some preferred domain names are unavailable in registry','',
      7, sysdate-55,
      4, 15, 'Closed', 'Medium', sysdate-45,
      '',sysdate-50,'')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (25, 'Establish grid management capacity-expansion policies with ASP','',
```

```
      7, sysdate-20,
      4, 16, 'Open', 'Medium', sysdate-5,
      '','','')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (26, 'Access through proxy servers blocks some usage tracking tools','',
      7, sysdate-10,
      4, 15, 'Closed', 'High', sysdate-5,
      '',sysdate-1,'')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (27, 'Phase I stress testing cannot use production network','',
      7, sysdate-11,
      4, 17, 'Open', 'High', sysdate,
      '','','')
/
INSERT INTO ht_issues
      (issue_id, issue_summary, issue_description,
      identified_by, identified_date,
      related_project, assigned_to, status, priority, target_resolution_date,
      progress, actual_resolution_date, resolution_summary)
  VALUES
      (28, 'DoD clients must have secure port and must be blocked from others','',
      7, sysdate-20,
      4, 17, 'On-Hold', 'High', sysdate,
      'Waiting on Security Consultant, this may drag on.','','')
/
```