

**Oracle® Database Express Edition**  
2 Day Plus Application Express Developer Guide  
Release 2.1  
**B25310-01**

February 2006

Oracle Database Express Edition 2 Day Plus Application Express Developer Guide, Release 2.1

B25310-01

Copyright © 2003, 2006, Oracle. All rights reserved.

Primary Author: Terri Winters

Contributors: Carl Backstrom, Sharon Kennedy, Sergio Leunissen, Raj Mattamal, and Simon Watt

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software—Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

---

---

# Contents

<b>Preface</b> .....	ix
Audience .....	ix
Documentation Accessibility .....	ix
Related Documents .....	x
Conventions .....	x
<b>1 How to Create a Tabular Form</b>	
<b>About the HR Account</b> .....	1-1
Unlocking the HR Account .....	1-1
Logging in to the HR User Account .....	1-2
<b>Creating an Application</b> .....	1-2
<b>Creating a Tabular Form Using a Wizard</b> .....	1-3
<b>Changing the Department Id Column to a Select List</b> .....	1-5
<b>2 How to Create a Parameterized Report</b>	
<b>Sample Report Utilizing a Form Input</b> .....	2-1
<b>About Sample Application</b> .....	2-2
<b>Creating a New Page</b> .....	2-2
<b>Creating the Query Region</b> .....	2-3
<b>Adding an Item</b> .....	2-4
<b>Adding a Button to Submit the Page</b> .....	2-4
Run the Page .....	2-5
<b>3 How to Create a Drill Down Report</b>	
<b>About Sample Application</b> .....	3-1
<b>Creating a New Application</b> .....	3-1
<b>Creating Reports for DEMO_ORDERS and DEMO_ORDER_ITEMS</b> .....	3-3
Create a Report for DEMO_ORDERS .....	3-3
Create a Report for DEMO_ORDER_ITEMS .....	3-5
<b>Customizing the DEMO_ORDER_ITEMS Report</b> .....	3-6
Add an Item to Hold the Value of ORDER_ID .....	3-6
Add a Condition .....	3-7
Modify the Region Title .....	3-7
<b>Linking the DEMO_ORDERS Report to the DEMO_ORDER_ITEMS Report</b> .....	3-7

## 4 How to Control Form Layout

<b>About Sample Application</b> .....	4-1
<b>Creating a Table and Data Input Form</b> .....	4-2
Create the HT_EMP Table .....	4-2
Create a New Page Containing a Input Form.....	4-3
Run the Page .....	4-4
<b>Changing the Appearance of a Page by Altering Region Attributes</b> .....	4-6
<b>Understanding How Item Attributes Effect Page Layout</b> .....	4-7
Edit Item Attributes .....	4-7
Fix Item Alignment .....	4-9
Add a Stop and Start HTML Table Item.....	4-9
Change Items to Display-only.....	4-10
<b>Adding a Region Header and Footer</b> .....	4-12
<b>Making a Region Conditional</b> .....	4-12
<b>Adding Another Region for HTML Text</b> .....	4-13
<b>Changing Item Types</b> .....	4-13
Change an Item to a Radio Group .....	4-14
Change an Item to a Select List .....	4-15
Change an Item to a Check Box .....	4-16
<b>About Label Templates</b> .....	4-16
<b>Changing Buttons</b> .....	4-17
<b>Running the Page for Update</b> .....	4-17
<b>Making Data Bold</b> .....	4-17

## 5 How to Work with Check Boxes

<b>About Sample Application</b> .....	5-1
<b>Creating a Single Value Check Box on a Form</b> .....	5-1
Change Product Available Radio Group to a Check Box.....	5-2
Alter the Check Box Position.....	5-3
Change Default Check Box Behavior .....	5-4
Add a Computation.....	5-5
<b>Creating Multi Value Check Box to Filter Content</b> .....	5-6
Create a Multi Value Check Box .....	5-6
Alter Check Box Display Values .....	5-7
Change Where the Check Boxes Display.....	5-7
Create a Go Button to Submit the Page.....	5-8
Adjust Default Check Box Behavior .....	5-9
<b>Adding Check Boxes to Each Row in the Report</b> .....	5-10
Call HTMLDB_ITEM.CHECKBOX .....	5-10
Add a Button to Submit Check Box Array Values .....	5-11
Add a Process .....	5-12

## 6 How to Implement a Web Service

<b>About Creating Web Service References</b> .....	6-1
<b>Creating a New Application</b> .....	6-2
<b>Specifying an Application Proxy Server Address</b> .....	6-3

<b>Searching a UDDI Registry for a Business Name</b> .....	6-3
Create a Form to Display a Stock Quote.....	6-4
<b>Searching a UDDI Registry for a Service Name</b> .....	6-5
Create a Form and Report.....	6-5
<b>7 How to Create a Stacked Bar Chart</b>	
<b>About Sample Application</b> .....	7-1
<b>Creating a Stacked Bar Chart</b> .....	7-2
<b>Adding Additional Series</b> .....	7-3
<b>Changing the Chart Format</b> .....	7-5
<b>Viewing the Chart</b> .....	7-6
<b>8 How to Upload and Download Files in an Application</b>	
<b>Creating an Application</b> .....	8-1
<b>Creating an Upload Form</b> .....	8-2
Create an HTML Region .....	8-2
Create an Upload Item .....	8-3
Create a Button .....	8-3
<b>Creating a Report with Download Links</b> .....	8-4
Create a Report on HTMLDB_APPLICATION_FILES .....	8-4
Add Link to Download Documents .....	8-5
<b>Storing Additional Attributes About the Document</b> .....	8-7
Create a Table to Store Document Attributes .....	8-7
Create an Item to Capture the Document Subject .....	8-7
Create a Process to Insert Information.....	8-8
Showing Additional Attributes in the Report Region.....	8-9
<b>Store the Document in a Custom Table</b> .....	8-10
<b>Downloading Documents from the Custom Table</b> .....	8-11
<b>9 How to Incorporate JavaScript into an Application</b>	
<b>Understanding How to Incorporate JavaScript Functions</b> .....	9-1
Incorporating JavaScript in the HTML Header Attribute.....	9-2
Including JavaScript in a .js File Referenced by the Page Template.....	9-2
<b>About Referencing Items Using JavaScript</b> .....	9-2
<b>Calling JavaScript from a Button</b> .....	9-3
<b>Creating a Client Side JavaScript Validation</b> .....	9-4
About Accessing the HR Account .....	9-4
Create an Application on the EMPLOYEES Table .....	9-4
Add a Function to the HTML Header Attribute .....	9-6
Edit an Item to Call the Function.....	9-6
<b>Enabling and Disabling Form Elements</b> .....	9-7
Add a Function the HTML Header Attribute.....	9-7
Edit an Item to Call the Function.....	9-8
Change P2_DEPARTMENT_ID to a Select List.....	9-8
Create a Call to the disFormItems Function.....	9-8
<b>Changing the Value of Form Elements</b> .....	9-9

## 10 How to Build and Deploy an Issue Tracking Application

<b>Planning and Project Analysis</b> .....	10-1
Planning and Project Analysis .....	10-2
Gathering the Necessary Data.....	10-2
Defining Security Requirements.....	10-2
Selecting Data Management Functions .....	10-3
Selecting Data Presentation Functions.....	10-3
Defining Special Function Requirements .....	10-3
<b>Designing the Database Objects</b> .....	10-3
About the Projects Table .....	10-4
About the People Table .....	10-4
About the Issues Table .....	10-5
<b>Implementing Database Objects</b> .....	10-6
Build the Database Objects .....	10-6
View the Created Database Objects.....	10-7
<b>Loading Demonstration Data</b> .....	10-8
Load Projects Data .....	10-8
Update Dates to Make the Projects Current.....	10-9
Load People Data .....	10-10
Load Issues Data .....	10-12
<b>Building a Basic User Interface</b> .....	10-12
Create the Application.....	10-12
Add Pages to Maintain Projects .....	10-14
Create Pages for Maintaining Projects .....	10-14
Refine the Appearance of the Projects Report Page.....	10-16
Refine the Create/Edit Project Page.....	10-17
Add Pages to Track People.....	10-19
Create Pages for Maintaining People.....	10-19
Modify the People Report Page .....	10-20
Refine the Create/Edit People Page.....	10-22
Add Pages to Track Issues .....	10-25
Create a Report for HT_ISSUES.....	10-25
Refine the Create/Edit Issues Page.....	10-26
Refine the Issues Report.....	10-35
Add a Page to Support Assigning Multiple Issues Simultaneously .....	10-43
Create Summary Reports .....	10-46
Add an Issue Summary by Project Report .....	10-47
Add Resolved by Month Identified Report .....	10-52
Add a Target Resolution Dates Report .....	10-53
Add an Average Days to Resolve Report.....	10-56
Add Content to the Home Page .....	10-57
Add a Reports Menu .....	10-58
Add Maintenance Navigation.....	10-60
Add a New Issues Button .....	10-61
Add An Overdue Issues Report.....	10-62
Add an Unassigned Issues Report .....	10-64
Add a Recently Opened Issues Report .....	10-65

Add an Open Issues by Project Pie Chart .....	10-66
Add a Breadcrumb Menu .....	10-67
Navigate to the Breadcrumbs Page .....	10-68
Add Breadcrumb Entries .....	10-68
Create a Page 0 .....	10-71
Create a Region to Contain the Breadcrumb .....	10-71
<b>Adding Advanced Features</b> .....	10-72
Add Support for Email Notification.....	10-72
How Email Notification Works .....	10-73
Add Notification of New Assignments .....	10-73
Add a Notification for Overdue Issues.....	10-75
Add Application Security .....	10-77
Restrict Project and People Definition .....	10-78
Restrict Issue Modification .....	10-80
<b>Deploying Your Application</b> .....	10-84
Move the Application Definition .....	10-84
Export the Application Definition .....	10-84
Create the Required Objects to Support the Application.....	10-85
Import the Application Definition into the Production Instance.....	10-85
Load the Data .....	10-86
Alternate Authentication Mechanisms to Consider.....	10-86
Create Users .....	10-86
Publish the URL.....	10-87

## **A DDLs and Scripts for Issue Tracking Application**

Create Application Database Objects DDL .....	A-1
Create Issues Script.....	A-6





---

---

# Preface

*Oracle Database Express Edition 2 Day Plus Application Express Developer Guide* contains tutorials with step-by-step instructions that explain how to create a variety application components and complete applications using the Oracle Application Express client within Oracle Database Express Edition (Oracle Database XE).

This preface contains these topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

## Audience

*Oracle Database Express Edition 2 Day Plus Application Express Developer Guide* is intended for application developers who wish to learn how to build database-centric Web applications using the Oracle Application Express client. 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 client, and Application Builder.

What is Application Builder? 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:

- "Application Builder Concepts" in *Oracle Database Express Edition Application Express User's Guide*.
- "Using Application Builder" in *Oracle Database Express Edition Application Express User's Guide*.

## 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.

## **Related Documents**

For more information, see these Oracle resources:

- *Oracle Database Express Edition Installation Guide for Linux*
- *Oracle Database Express Edition Installation Guide for Microsoft Windows*
- *Oracle Database Express Edition 2 Day DBA Guide*
- *Oracle Database Express Edition 2 Day Developer Guide*
- *Oracle Database Express Edition Application Express User's Guide*

For the most recent version of the Oracle Database Express Edition documentation, see the Oracle Database XE online library:

<http://www.oracle.com/technology/xe/documentation>

## **Conventions**

The following text conventions are used in this document:

<b>Convention</b>	<b>Meaning</b>
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	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.

---

---

# How to Create a Tabular Form

A tabular form enables users to update multiple rows in a table at once using 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.

This section contains the following topics:

- [About the HR Account](#)
- [Creating an Application](#)
- [Creating a Tabular Form Using a Wizard](#)
- [Changing the Department Id Column to a Select List](#)

## About the HR Account

To create the tabular form in this exercise, you need to log in to the sample database user account HR. Oracle Database Express Edition (Oracle Database XE) comes with a sample database user account called HR. This user owns a number of database tables in a sample schema that can be used to create applications for a fictional Human Resources department.

Topics in this section include:

- [Unlocking the HR Account](#)
- [Logging in to the HR User Account](#)

## Unlocking the HR Account

For security reasons, the HR user account is locked after installation. You need to unlock this account before you can build a new application. If you have already unlocked the HR account, go to "[Creating an Application](#)" on page 1-2.

To unlock the sample user account:

1. Open the Database Home Page login window:
  - On Windows, from the **Start** menu, select **Programs** (or **All Programs**), then **Oracle Database 10g Express Edition**, and then **Go To Database Home Page**.
  - On Linux, click the **Application** menu (on Gnome) or the **K** menu (on KDE), then point to **Oracle Database 10g Express Edition**, and then **Go To Database Home Page**.

2. At the Database Home Page login window, enter the following information:
  - **Username:** Enter `system` for the user name.
  - **Password:** Enter the password that was specified when Oracle Database XE was installed.
3. Click **Login**.  
The Database Home Page appears.
4. Click the **Administration** icon, and then click **Database Users**.
5. Click the **HR** schema icon to display the user information for HR. (See [Figure 1–1](#).)

**Figure 1–1 HR Schema icon**



6. Under Manage Database User, enter the following settings:
  - **Password** and **Confirm Password:** Enter `hr` for the password.
  - **Account Status:** Select **Unlocked**.
  - **Roles:** Ensure that both **CONNECT** and **RESOURCE** are enabled.
7. Click **Alter User**.

## Logging in to the HR User Account

To log in to the HR user account:

1. Log out from the database administrator account by clicking **Logout** in the upper right corner of the page.
2. In the window, click **Login**.
3. In the Login window, enter `hr` for both the user name and password.
4. Click **Login**.  
The Database Home Page appears.

## Creating an Application

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

To create an application using the Create Application Wizard:

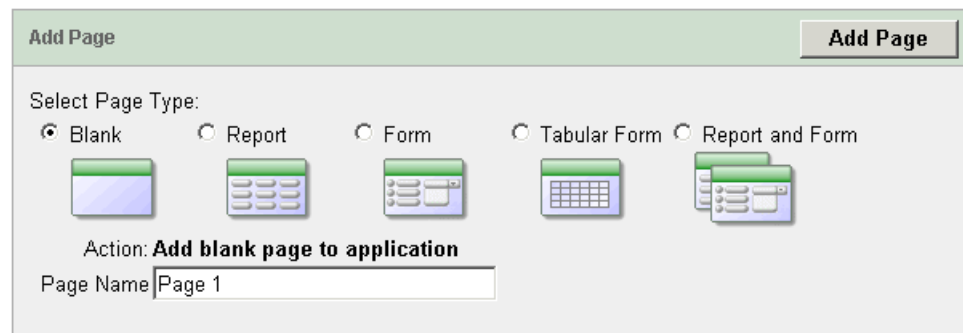
1. On the Database Home Page, click the **Application Builder** icon.  
The Application Builder home page appears.
2. Click **Create**.
3. Select **Create Application** and click **Next**.
4. Specify the page name.
  - a. For Name, enter `Tabular Form`.
  - b. For Application, accept the default.

- c. For Create Application, select **From scratch**.
- d. For Schema, accept the default.
- 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 one blank page.

5. Add a blank page:
  - a. Under Select Page Type, select **Blank** and click **Add Page** as shown in [Figure 1-2](#).

**Figure 1-2 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 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 in order to prevent lost updates. The following exercise creates a tabular form on the HR table.

To create a tabular form using the Tabular Form Wizard:

1. Navigate to the Application home page. On the Database Home Page, click the **Application Builder** icon.
2. Select **Tabular Form**.
3. Click **Create Page**.

4. For the page type, select **Form** and click **Next**.
5. Select **Tabular Form** and click **Next**.
6. For Table/View Owner:
  - a. From Table/View Owner, select **HR**.
  - b. From Allowed Operations, accept the default, **Update, Insert, and Delete**.
  - c. Click **Next**.
7. For Table/View Name, select **EMPLOYEES** and click **Next**.
8. 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**.
9. For Primary Key, accept the default **ID (Number)** and click **Next**.
10. For Source Type, accept the default, **Existing trigger**, and click **Next**.
11. For Updatable Columns, select all columns and click **Next**:
12. On Identify Page and Region Attributes:
  - a. For Page, accept the default.
  - b. For Page Name, enter `Tabular Form`.
  - c. For Region Title, enter `Tabular Form`.
  - d. For Region Template and Report Template, accept the defaults.
  - e. Click **Next**.
13. For Tab, accept the default, **Do not use tabs**, and click **Next**.
14. For Button Labels:
  - a. Accept the defaults for the Cancel, Delete, and Add Row buttons.
  - b. For the Submit button, enter `Apply Changes`.
  - c. Click **Next**.
15. Accept the remaining defaults, click **Next**.
16. 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 1-3](#) on page 1-5.

**Figure 1–3 Run Page Icon**

2. If prompted to enter a username and password:
  - a. For User Name, enter `hr`.
  - b. For Password, enter `hr`.
  - c. Click **Login**.

The tabular form appears as shown in [Figure 1–4](#).

**Figure 1–4 Tabular Form**

Tabular Form						Cancel	Delete	Apply Changes
<input type="checkbox"/>	First Name	Last Name	Hire Date	Salary	Department Id			
<input type="checkbox"/>	Steven	King	17-JUN-87	24000	90			
<input type="checkbox"/>	Neena	Kochhar	21-SEP-89	17000	90			
<input type="checkbox"/>	Lex	De Haan	13-JAN-93	17000	90			
<input type="checkbox"/>	Alexander	Hunold	03-JAN-90	9000	60			
<input type="checkbox"/>	Bruce	Ernst	21-MAY-91	6000	60			
<input type="checkbox"/>	David	Austin	25-JUN-97	4800	60			
<input type="checkbox"/>	Valli	Pataballa	05-FEB-98	4800	60			
<input type="checkbox"/>	Diana	Lorentz	07-FEB-99	4200	60			
<input type="checkbox"/>	Nancy	Greenberg	17-AUG-94	12000	100			
<input type="checkbox"/>	Daniel	Faviet	16-AUG-94	9000	100			

row(s) 1 - 10 of 107 [Next](#) [Add Row](#)

As shown in [Figure 1–4](#), note the tabular form contains four buttons. Cancel, Delete, and Apply Changes display in the upper left corner and Add Row displays in the bottom right corner. Additionally, a check box appears to the left of each row enabling the users to select one row. Users can also select all rows at once by selecting the check box to the left of the column headings. That same check box is also used in conjunction with the Delete button to identify the rows to be deleted.

## Changing the Department Id Column to a Select List

When the Tabular Form Wizard creates a tabular form, updatable columns display by default as text fields. You can change this default display by editing report column attributes. In the next exercise, you change the default display of the Department Id column to a select list.

To change the default display of a column to a select list:

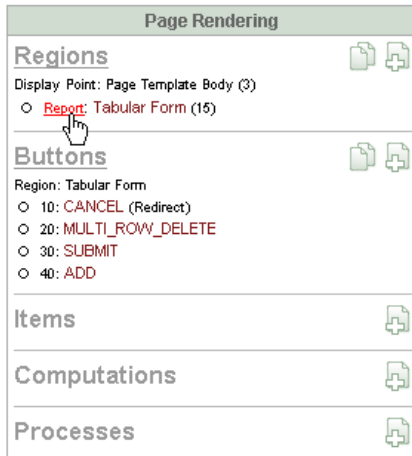
1. Navigate to the Page Definition for page 2. Click **Edit Page 2** on the Developer toolbar as shown in [Figure 1-5](#).

**Figure 1-5 Developer Toolbar**



2. Under Regions, click **Report** next to Tabular Form as shown [Figure 1-6](#).

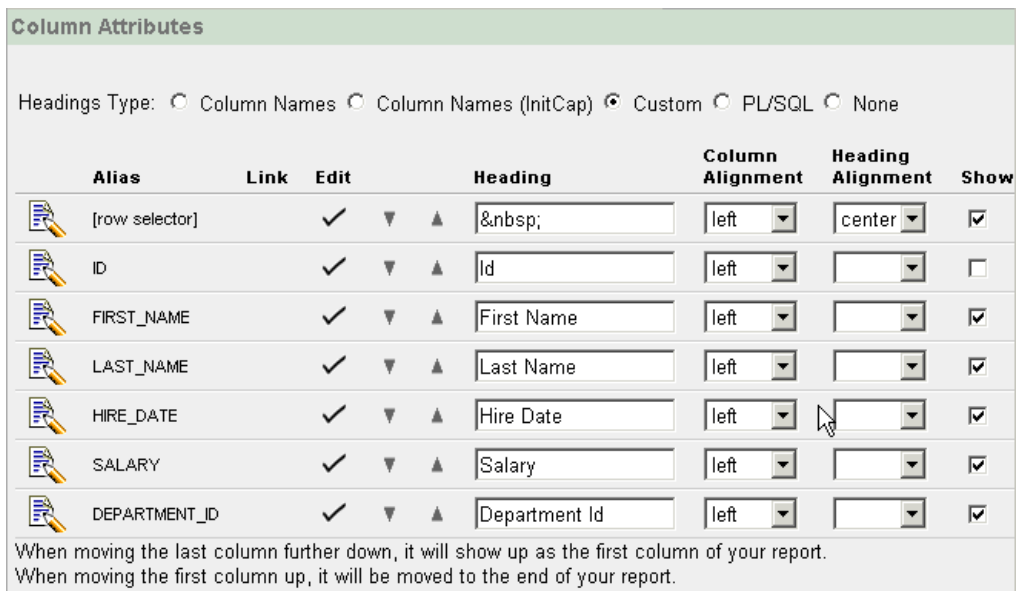
**Figure 1-6 Report Link**



The Report Attributes page appears.

3. Under Column Attributes, click the **Edit** icon next to the DEPARTMENT\_ID column as shown in [Figure 1-7](#).

**Figure 1-7 Column Attributes**



The Column Attributes page appears. Next, change the default display of this column to a select list.



4. Scroll down to Tabular Form Element.
5. From Display As, select **Select List (query based LOV)**.  
Next, create your query.
6. Scroll down to List of Values and enter the following in LOV Query:

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

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

As shown in [Figure 1-8](#), notice the Department Id column now displays as a select list.

**Figure 1-8 Tabular Form with Department Id Column Changed to a Select List**

**Tabular Form**

<input type="checkbox"/>	First Name	Last Name	Hire Date	Salary	Department Id
<input type="checkbox"/>	Steven	King	17-JUN-87	24000	90 ▼
<input type="checkbox"/>	Neena	Kochhar	21-SEP-89	17000	90 ▼
<input type="checkbox"/>	Lex	De Haan	13-JAN-93	17000	90 ▼
<input type="checkbox"/>	Alexander	Hunold	03-JAN-90	9000	60 ▼
<input type="checkbox"/>	Bruce	Ernst	21-MAY-91	6000	60 ▼
<input type="checkbox"/>	David	Austin	25-JUN-97	4800	60 ▼
<input type="checkbox"/>	Valli	Pataballa	05-FEB-98	4800	60 ▼
<input type="checkbox"/>	Diana	Lorentz	07-FEB-99	4200	60 ▼
<input type="checkbox"/>	Nancy	Greenberg	17-AUG-94	12000	100 ▼
<input type="checkbox"/>	Daniel	Faviet	16-AUG-94	9000	100 ▼

row(s) 1 - 10 of 107 ▼

---

**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 data of the form and applying updates.

Consider the following example:

```
SELECT first_name FROM employees;
```

Note that this should not be altered to:

```
SELECT lower(first_name) FROM employees
```

---



---

---

## 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, or a parameterized report. In this exercise, you create a report region based on a SQL query which references the value of a form item within the application.

This section contains the following topics:

- [Sample Report Utilizing a Form Input](#)
- [About Sample Application](#)
- [Creating a New Page](#)
- [Creating the Query Region](#)
- [Adding an Item](#)
- [Adding a Button to Submit the Page](#)

### Sample Report Utilizing a Form Input

[Figure 2-1](#) on page 2-2 is an example of a form in which the report results are based on user input. In this example, this user populates the form by making a selection from the Show list. The easiest way to create this type of report in Application Builder is to define a report region based on a SQL query.

Figure 2–1 Sample Report

Ordered Products

Show

CATEG	NAME	QUANTITY
Comp	Computer	1
Audio	Audio	1
Video	Classic Projector	2
Audio	Stereo Headphones	1
Audio	Stereo Headphones	1
Computer	Ultra Slim Laptop	1
Video	Portable DVD Player	1
Phones	PDA Cell Phone	1
Phones	Bluetooth Headset	1
Phones	PDA Cell Phone	1
Computer	3.2 GHz Desktop PC	3
Video	Classic Projector	4
Phones	Bluetooth Headset	1
Computer	512 MB DIMM	1
Phones	PDA Cell Phone	1

1 - 15

## About Sample Application

Application Builder installs with a number of demonstration applications. In this exercise, you create a new page within the demonstration application, *Sample Application*.

To see if *Sample Application* is installed:

1. Log in to the Database Home Page using your database username and password.
2. Click the down arrow on the right side of the **Application Builder** icon.
3. From the menu, select **Demonstrations**.

The Demonstration Applications page appears, displaying links to demonstration applications.

4. Locate *Sample Application* and check the Status column:
  - a. If the Status column displays **Installed**, return to the Database Home Page.
  - b. If the Status column displays **Not Installed**, select **Install** in the Action column. Follow the on-screen instructions.

## Creating a New Page

First, you create a new blank page and within *Sample Application*.

To create a new page:

1. On Database Home Page, click the **Application Builder** icon.
2. Select **Sample Application**.
3. Click the **Create Page** button.

4. For Page, select **Blank Page** and click **Next**.
5. For Page, enter 700 and click **Next**.
6. In Name, enter `Ordered Products` and click **Next**.
7. For Tabs, accept the default, **No**, and click **Next**.
8. Review your selections and click **Finish**.
9. On the Success Page, click the **Edit Page** icon.  
The Page Definition for page 700 appears.

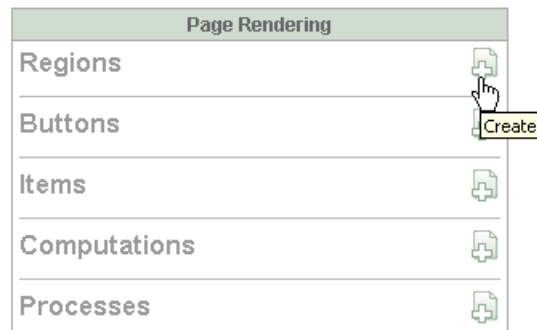
## Creating the Query Region

Next, you need to create a report.

To create a query region to contain the report:

1. Under Regions, click the **Create** icon as shown in [Figure 2-2](#).

**Figure 2-2 Create Icon**



2. Select **Report** and click **Next**.
3. For Report Implementation, select **SQL Report** and click **Next**.
4. For Display Region Attributes:
  - a. For Title, enter `Ordered Products`.
  - b. Accept the remaining default values and click **Next**.
5. Enter following SQL query:
 

```
SELECT p.category,
       p.product_name,
       i.quantity
  FROM demo_product_info p,
       demo_order_items i
 WHERE p.product_id = i.product_id
       AND ( p.category = :P700_SHOW or :P700_SHOW = 'ALL' )
```
6. Click **Create Region**.

## Adding an Item

The previous SQL query references an item named P700\_SHOW. Next, you need to create this item. 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.

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

To create the select list P700\_SHOW:

1. Under Items, click the **Create** icon as shown in [Figure 2-3](#).

**Figure 2-3 Create Icon**



2. For Item Type, select **Select List** and click **Next**.
3. For Select List Control Type, accept the default of **Select List** and click **Next**.
4. For Item Name, enter P700\_SHOW, accept the remaining defaults, and click **Next**.
5. For List of Values:
  - a. For Named LOV, select **CATEGORIES**.
  - b. In Null Text, enter:  
- All Categories -
  - c. For Null Value, enter:  
ALL
  - d. Click **Next**.
6. For Item Attributes, accept the defaults and click **Next**.
7. Click **Create Item**.

## Adding a Button to Submit the Page

For the report to be driven by the Product Category select list (the form input), you need to submit the page. To accomplish this, you need add a button.

To add a button to submit the page:

1. Under Buttons, click the **Create** icon.
2. For Button Region, select **Ordered Products** 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 P700\_GO.
5. Accept the remaining defaults and click **Create Button**.

The Page Definition for page 700 appears.

**See Also:** "Creating Buttons" in *Oracle Database Express Edition Application Express User's Guide*

## Run the Page

To run the page:.

1. Click the **Run Page** icon in the upper right corner as shown in [Figure 2-4](#).

**Figure 2-4 Run Page Icon**



2. If prompted to enter a username and password:
  - a. For User Name, enter either demo or admin.
  - b. For Password, enter your database username in lowercase letters.
  - c. Click **Login**.
3. When the Order Products page appears, select **Computer** from the Show menu and click **Go**.

As shown in [Figure 2-5](#), notice that making a selection from the Show menu populates the form.

**Figure 2-5 Form Results Being Populated from a Select List**

**Ordered Products**

---

Show

CATEGORY	PRODUCT_NAME	QUANTITY
Computer	3.2 GHz Desktop PC	1
Computer	Ultra Slim Laptop	1
Computer	3.2 GHz Desktop PC	3
Computer	512 MB DIMM	1

1 - 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 a second report.

This tutorial describes how to create a report on the `DEMO_ORDERS` table with links to drill down detail data in the `DEMO_ORDER_ITEMS` table. Both tables are installed with the demonstration application, *Sample Application*.

This section contains the following topics:

- [About Sample Application](#)
- [Creating a New Application](#)
- [Creating Reports for DEMO\\_ORDERS and DEMO\\_ORDER\\_ITEMS](#)
- [Customizing the DEMO\\_ORDER\\_ITEMS Report](#)
- [Linking the DEMO\\_ORDERS Report to the DEMO\\_ORDER\\_ITEMS Report](#)

### About Sample Application

Application Builder installs with a number of demonstration applications. In this exercise you create a drilldown report using the tables that install with the demonstration application, *Sample Application*.

To see if *Sample Application* is installed:

1. Log in to the Database Home Page using your database username and password.
2. Click the down arrow on the right side of the **Application Builder** icon.
3. From the menu, select **Demonstrations**.

The Demonstration Applications page appears, displaying links to demonstration applications.

4. Locate *Sample Application* and check the Status column:
  - a. If the Status column displays **Installed**, return to the Database Home Page.
  - b. If the Status column displays **Not Installed**, select **Install** in the Action column. Follow the on-screen instructions.

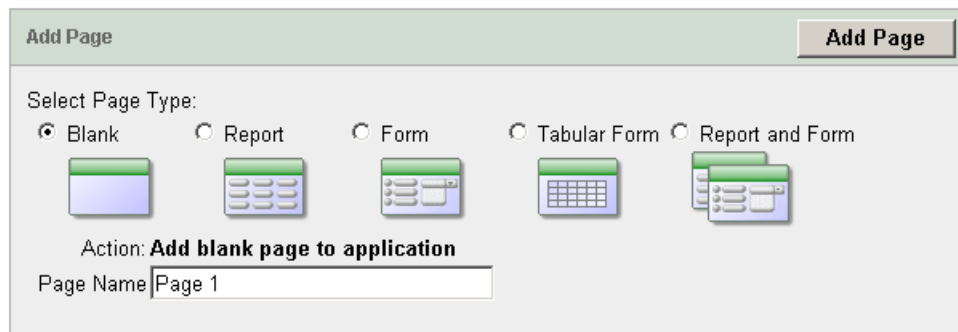
### Creating a New Application

First, create a new application.

To create an application:

1. On the Database Home Page, click the **Application Builder** icon.  
The Application Builder home page appears.
2. Click **Create**.
3. Select **Create Application** and click **Next**.
4. Specify the page name.
  - a. For Name, enter `Drilldown Reports`.
  - b. For Application, accept the default.
  - c. For Create Application, select **From scratch**.
  - d. For Schema, accept the default.
  - 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. Add the first blank page:
  - a. Under Select Page Type, select **Blank** as shown in [Figure 3-1](#).

**Figure 3-1 Add Page**



- b. In Page Name, enter `Orders`.
  - c. Click **Add Page**.
6. Add the second blank page:
  - a. Under Select Page Type, select **Blank**.
  - b. In Page Name, enter `Order Items`.
  - c. 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 your application contains three pages:

- 1 - Orders
- 2 - Order Items
- 101 - Login

## Creating Reports for DEMO\_ORDERS and DEMO\_ORDER\_ITEMS

Next, you need to create reports for the DEMO\_ORDERS and the DEMO\_ORDER\_ITEMS tables.

Topics in this section include:

- [Create a Report for DEMO\\_ORDERS](#)
- [Create a Report for DEMO\\_ORDER\\_ITEMS](#)

### Create a Report for DEMO\_ORDERS

To create a report on the DEMO\_ORDERS table:

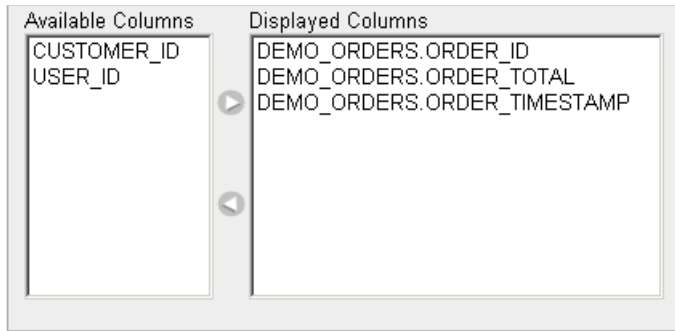
1. On the Application home page, click **Create Page**.
2. Select the page type **Report** and click **Next**.
3. Select **Wizard Report** and click **Next**.
4. For Page Attributes:
  - a. For Page, select **1 Orders**.
  - b. In Page Title and Region Title, enter *Orders*.
  - c. For Region Template, accept the default.
  - d. Click **Next**.
5. For Tabs, accept the default, **Do not use tabs**, and click **Next**.
6. For Tables and Columns:
  - a. For Table/View Owner, select the default.
  - b. For Table/View, select **DEMO\_ORDERS**.

The columns in the DEMO\_ORDERS table appear.

- c. From the Available Columns list, press **CTRL** and move the following columns to the Displayed Columns list as shown in [Figure 3–2](#) on page 3-4:

ORDER\_ID, ORDER\_TOTAL, ORDER\_TIMESTAMP

**Figure 3–2 Selected Columns**



Next, create a join with the DEMO\_CUSTOMERS table to display the customer name. First, select a new table.

- d. From the Table/View list, select **DEMO\_CUSTOMERS**.  
The columns in the DEMO\_CUSTOMERS table appear.
- e. From the Available Columns list, select **CUST\_LAST\_NAME** and move it to the Display Columns list.
- f. Click **Next**.
7. For Join Conditions, accept the join by clicking **Next**.
8. For Report Options, accept the defaults and click **Next**.
9. Click **Create Report Page**.
10. Run the page by clicking the **Run Page** icon. If prompted for a user name and password, enter your database user account credentials.

As shown in [Figure 3–3](#), a report on the DEMO\_ORDERS table appears.

**Figure 3–3 Report on DEMO\_ORDERS Table**

Orders			
Order Id	Cust Last Name	Order Total	Order Timestamp
1	Dulles	1200	23-SEP-05
2	Hartsfield	599	18-SEP-05
3	Hartsfield	1999	13-SEP-05
4	Logan	750	08-SEP-05
5	Logan	40	03-SEP-05
6	O'Hare	250	29-AUG-05
7	LaGuardia	3800	24-AUG-05
8	Lambert	40	19-AUG-05
9	Lambert	450	14-AUG-05
10	Bradley	500	09-AUG-05

1 - 10

11. Click **Edit Application** on the Developer toolbar to return to Application Builder.  
The Application home page appears.

## Create a Report for DEMO\_ORDER\_ITEMS

To create a report on the DEMO\_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. For Page, select 2 Order Items.
  - b. In Page Title and Region Title, enter Order Items.
  - c. For Region Template, accept the default.
  - d. Click **Next**.
5. For Tables and Columns:
  - a. For Table/View Owner, accept the default.
  - b. For Table/View, select **DEMO\_ORDER\_ITEMS**.  
The columns in the DEMO\_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 DEMO\_PRODUCT\_INFO table to display the product name.
  - d. For Show Only Related Tables, select **No**.  
Then, select a new table.
  - e. From the Table/View list, select **DEMO\_PRODUCT\_INFO**.  
The columns in the DEMO\_PRODUCT\_INFO table appear.
  - f. From the Available Columns list, select **PRODUCT\_NAME** and move it to the Display Columns list.
  - g. Click **Next**.
6. For Join Conditions:
  - a. Under the first Column, select **DEMO\_ORDER\_ITEMS.PRODUCT\_ID**.
  - b. Under the second Column, select **DEMO\_PRODUCT\_INFO.PRODUCT\_ID**.
  - c. 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 3-4](#) on page 3-6, a report on the DEMO\_ORDER\_ITEMS table appears.

**Figure 3–4 Report on DEMO\_ORDER\_ITEMS Table**

Order Items				
Order Item Id	Order Id	Product Name	Unit Price	Quantity
1	1	3.2 GHz Desktop PC	1200	1
11	7	3.2 GHz Desktop PC	1200	3
2	2	MP3 Player	199	1
9	5	Bluetooth Headset	40	1
13	8	Bluetooth Headset	40	1
8	4	PDA Cell Phone	250	1
15	9	PDA Cell Phone	250	1
10	6	PDA Cell Phone	250	1
7	4	Portable DVD Player	500	1
16	10	Portable DVD Player	500	1
14	9	512 MB DIMM	200	1
3	2	Classic Projector	50	2
12	7	Classic Projector	50	4
6	3	Ultra Slim Laptop	1999	1
4	2	Stereo Headphones	150	1

row(s) 1 - 15 of 16 Next >

## Customizing the DEMO\_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 condition 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 Condition](#)
- [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. Navigate to the Page Definition of page 2. Click **Edit Page 2** on the Developer toolbar.  
The Page Definition appears.
2. Under Items, click the **Create** icon as shown in [Figure 3–5](#).

**Figure 3–5 Create Icon**



3. For Item Type, select **Hidden** and click **Next**.
4. For Display Position and Name:
  - a. In Item Name, enter P2\_ORDER\_ID.

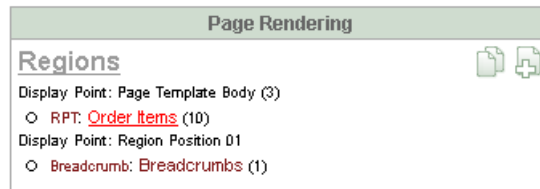
- b. In Sequence, accept the default.
  - c. For Region, select **Order Items**.
  - d. Click **Next**.
5. Click **Create Item**.

## Add a Condition

To add a condition to the DEMO\_ORDER\_ITEMS report:

1. Under Regions, select **Order Items** as shown in [Figure 3-6](#).

**Figure 3-6 Order Items**



2. Select the **Query Definition** tab.
3. Click **Modify Join Conditions**.
4. On the Modify Join Conditions page:
  - a. For first Column, select `DEMO_ORDER_ITEMS.ORDER_ID`. Note that you may need to click **Search** to view available columns.
  - b. Enter the following condition in the second Column field:  
`= :P2_ORDER_ID`
5. Click **Apply Changes**.

## Modify the Region Title

To modify the region title of the DEMO\_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 DEMO\_ORDERS Report to the DEMO\_ORDER\_ITEMS Report

Lastly, you link the DEMO\_ORDERS report to the DEMO\_ORDER\_ITEMS report. To accomplish this, you edit the attributes of the ORDER\_ID column on the DEMO\_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 Orders report to the Order Items report:

1. Navigate to the Page Definition of page 1, Orders. On the Page Definition, enter 1 in the Page field and click **Go**.

2. Under Regions, select **Orders**.
3. Click the **Report Attributes** tab.
4. Click the **Edit** icon adjacent 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 3-7](#).

**Figure 3-7 Column Link Attributes for the Order Id column**

The screenshot shows the 'Column Link' configuration window. The 'Link Text' field contains '#ORDER\_ID#'. Below it, a preview shows '[ORDER\_ID] [Icon 1] [Icon 2] [Icon 3] [Icon 4]'. The 'Link Attributes' field is empty. The 'Target' dropdown is set to 'Page in this Application'. The 'Page' field contains '2'. There is a 'Reset Pagination' checkbox. The 'Request' field is empty. The 'Clear Cache' field is empty. Below these are three rows for 'Item Name' and 'Item Value':

Name	Value
Item 1: P2_ORDER_ID	#ORDER_ID#
Item 2:	
Item 3:	

6. Scroll back 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 3-8](#) on page 3-9, you can link to page 2 by clicking on an Order Id.



**Figure 3–8 DEMO\_ORDERS Report with Link to Page 2**

Orders			
Order Id	Cust Last Name	Order Total	Order Timestamp
<a href="#">1</a>	Dulles	1200	23-SEP-05
<a href="#">2</a>	Hartsfield	599	18-SEP-05
<a href="#">3</a>	Hartsfield	1999	13-SEP-05
<a href="#">4</a>	Logan	750	08-SEP-05
<a href="#">5</a>	Logan	40	03-SEP-05
<a href="#">6</a>	OHare	250	29-AUG-05
<a href="#">7</a>	LaGuardia	3800	24-AUG-05
<a href="#">8</a>	Lambert	40	19-AUG-05
<a href="#">9</a>	Lambert	450	14-AUG-05
<a href="#">10</a>	Bradley	500	09-AUG-05

1 - 10



---

---

# 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 inside of regions using item attributes.

This tutorial describes how to create a data input form and then change its layout by changing region and item attributes.

This section contains the following topics:

- [About Sample Application](#)
- [Creating a Table and Data Input Form](#)
- [Changing the Appearance of a Page by Altering Region Attributes](#)
- [Understanding How Item Attributes Effect Page Layout](#)
- [Adding a Region Header and Footer](#)
- [Making a Region Conditional](#)
- [Adding Another Region for HTML Text](#)
- [Changing Item Types](#)
- [About Label Templates](#)
- [Changing Buttons](#)

## About Sample Application

Application Builder installs with a number of demonstration applications. In this exercise you create a data input form within the demonstration application, *Sample Application*.

To see if *Sample Application* is installed:

1. Log in to the Database Home Page using your database username and password.
2. Click the down arrow on the right side of the **Application Builder** icon.
3. From the menu, select **Demonstrations**.

The Demonstration Applications page appears, displaying links to demonstration applications.

4. Locate *Sample Application* and check the Status column:
  - a. If the Status column displays **Installed**, return to the Database Home Page.

- b. If the Status column displays **Not Installed**, select **Install** in the Action column. Follow the on-screen instructions.

## Creating a Table and Data Input Form

The first step in creating a data input form is to create the underlying data objects in SQL Scripts. 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](#)
- [Create a New Page Containing a Input Form](#)
- [Run the Page](#)

### 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 Database Home Page, click the **SQL** icon.
2. Click **SQL Scripts**.
3. When the SQL Scripts appears, click **Create**.

The Script Editor appears.

4. In Script Name, enter HT\_EMP.
5. In the Script Editor, enter the following 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;
/

```

6. Click **Save**.  
The script appears in the SQL Scripts Repository.
7. Run the HT\_EMP script:
  - a. Click the HT\_EMP script.
  - b. Click **Run**.
  - c. On the Run Script page, click **Run** again.  
Next, change the page view to Details.
8. From the View list, select **Details** and click **Go**.

**Figure 4–1 Details View in the Manage Script Results Page**

<input type="checkbox"/>	Script	Run By	Started ▼	Elapsed	Status	Statements	Bytes	View
<input type="checkbox"/>	HT_EMP	DOC2	117 seconds ago	0.78	Complete	5 of 5	0	
row(s) 1 - 1 of 1								

9. Click the **View** icon to access the Results page.  
Red text indicates errors while executing the file.

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

## Create a New Page Containing a Input Form

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

To create a data input form:

1. Navigate to the Database Home Page by clicking the Home breadcrumb link at the top of the page.
2. Click the **Application Builder** icon.
3. Select **Sample Application**.
4. Click **Create Page**.
5. For Page, select **Form** and click **Next**.
6. On Create Page, select **Form on a Table or View** and click **Next**.
7. For Table/View Owner, accept the default and click **Next**.
8. For Table/View Name, select the **HT\_EMP** table and click **Next**.
9. For Page and Region Attributes:
  - a. In Page, enter 900.
  - b. In the Page Name, enter HT\_EMP.
  - c. In the Region Title, enter How to Layout a Form.
  - d. For Region Template, accept the default.
  - e. Click **Next**.
10. For Tab, accept the default, **Do not use tabs**, and click **Next**.
11. For Primary Key, accept the default and click **Next**.

Note that the wizard reads the primary key from the database definition.

12. For Source Type, accept the default **Existing Trigger** and click **Next**.
13. For Select Columns, press SHIFT, select all columns, and click **Next**.
14. For Process Options, accept the defaults and click **Next**.
15. For Branching, in the fields provided enter 900 (the page you are creating) and click **Next**.

Since this page is just for demonstration, you will not be utilizing branching.

16. Click **Finish**.
17. View the Page Definition by clicking the **Edit Page** icon.

The Page Definition for page 900 appears.

18. Delete the following validation:
  - Under Page Processing Validations, select **P900\_REC\_CREATE\_DATE not null**.
  - Click **Delete**.

## Run the Page

Once you create the HT\_EMP table and page 900, 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 4-2](#) on page 4-5.

**Figure 4–2 Run Page Icon**

2. If prompted to enter a username and password:
  - a. For User Name, enter either `demo` or `admin`.
  - b. For Password, enter the name of your database user account in all lowercase letters.
  - c. Click **Login**.  
Sample Application appears.
  - d. Navigate to page 900:
    - Click **Edit Page 1** on the Developer toolbar at the bottom of the page.  
The Page Definition appears.
    - In the Page field, enter **900** and click **Go**.
  - e. On the Page Definition for page 900, click the **Run Page** icon in the upper right corner.

As shown in [Figure 4–3](#), the new form appears. Note that the `HT_EMP` form contains basic employee details and includes select lists, text areas, and display only items.

**Figure 4–3 HT\_EMP 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 Database Express Edition Application Express 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 for wrapping of input text.

## Changing the Appearance of a Page by Altering Region Attributes

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

To change the region title and other region level attributes:

1. Navigate to the Page Definition for page 900. Click **Edit Page 900** on the Developer toolbar.
2. Under Regions, click **How to Layout a Form** as shown in [Figure 4-4](#).

**Figure 4-4** *How to Layout a Form Link*



The Region Definition appears.

3. Change the Title to `Employee Info`.
4. Scroll down to User Interface.
5. From Display Point, temporarily move the region by selecting **Page Template Region Position 3**.

The Display Point controls the position of a region within the page template. In this instances, 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, Application Express engine 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.

6. From Template, select **Borderless Region**.
7. Click **Apply Changes** at the top of the page.
8. From the Page Definition, click the **Run Page** icon in the upper right corner.

The form appears as shown in [Figure 4-5](#) on page 4-7.



**Figure 4–5 HT\_EMP Form with New Display Point and Template**

The screenshot shows a form titled "Employee Info" with a "Cancel" and "Create" button at the top. The form contains the following fields:

- Emp First Name (text field)
- Emp Middle Initial (text field)
- Emp Last Name (text field)
- Emp Part Or Full Time (checkbox)
- Emp Salary (text field)
- Emp Dept (text field)
- Emp Hiredate (calendar icon)
- Emp Manager (text field)
- Emp Special Info (text area)
- Emp Telecommute (checkbox)
- Rec Create Date (calendar icon)
- Rec Update Date (calendar icon)

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

1. Navigate to the Page Definition for page 900. Click **Edit Page 900** 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 Effect Page Layout

An item is part of an HTML form. An item can be a text field, a text area, a password, a select list, a 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 Items to Display-only](#)

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

### Edit Item Attributes

To edit item attributes:

1. Navigate to the Page Definition for page 900.
2. On the Page Definition, select the title **Items**.

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. You can reorder items by changing the sequence. You can reorder items by entering 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 4-1](#):

**Table 4-1 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
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. (See [Figure 4-6](#) on page 4-9.)

Figure 4–6 HT\_EMP After Editing the Prompt, New Line, Width Attributes

The screenshot shows a form titled "Employee Info" with a blue header bar containing "Cancel" and "Create" buttons. The form contains several input fields: "First Name", "Middle Initial", "Last Name", "Part or Full Time", "Salary", "Department", "Hire Date", "Special Information" (a large text area), "Manager", "Telecommute", "Record Create Date", and "Record Update Date". The "Special Information" field is significantly wider than the others, which causes the "First Name", "Middle Initial", and "Last Name" fields to be pushed far to the right, creating an uneven layout.

Note that some items are pushed too far to the right. This display is caused by the width of the Special Information item. Oracle Application Express lays out regions as tables and of a 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

To add a Stop and Start HTML Table item and reset the column width:

1. Navigate to the Page Definition for page 900. Click **Edit Page 900** on the Developer toolbar.
2. Under Items, click the **Create** icon as shown in [Figure 4-7](#).

**Figure 4-7 Create Icon**



3. For Item Type, select **Stop and start table** and click **Next**.  
The Create Item Wizard appears.
4. For Display Position and Name:
  - a. For Item Name, accept the default.
  - b. In Sequence, enter 4 . 5.
  - c. From Region, select **Employee Info**.
  - d. Click **Next**.
5. On Item Attributes:
  - a. Remove the X in the Label field by clicking **Clear**.
  - b. Accept all other defaults and click **Next**.
6. Click **Create Item**.  
The Page Definition appears.
7. Edit the column width for Special Information:
  - a. Click the title **Items**.
  - b. For Special Information, change the Column Span to 3 and click **Apply Changes**.
8. Click the **Run Page** icon in the upper right corner.

## Change Items 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 describes how to make items display-only and then how to add them to their own region.

To make an item P900\_REC\_CREATE\_DATE display-only:

1. Navigate to the Page Definition for page 900.
2. Under Items, select the item **P900\_REC\_CREATE\_DATE**.
3. From Display As, select **Text Field (Disabled, saves state)**.
4. Click **Apply Changes**.

To make an item P900\_REC\_UPDATE\_DATE display-only:

1. Navigate to the Page Definition for page 900.
2. Under Items, select the item **P900\_REC\_UPDATE\_DATE**.
3. From Display As, select **Text Field (Disabled, saves state)**.

#### 4. Click **Apply Changes**.

Next, create a new region and then move the audit items into that region.

To create a new region:

1. Navigate to the Page Definition for page 900.
2. Under Regions, click the **Create** icon as shown in [Figure 4–8](#).

**Figure 4–8 Create Icon**



3. For the region type, select **HTML** and click **Next**.
4. Specify the type of region container. Select **HTML** and click **Next**.
5. For the Title, enter `Audit Information`, accept the remaining defaults, and click **Next**.
6. Click **Create Region**.

To move the items to the new region:

1. Navigate to the Page Definition for page 900.
2. Select the heading **Items**.  
A summary page appears.
3. For `P900_REC_CREATE_DATE` and `P900_REC_UPDATE_DATE`:
  - a. Under Prompt, add a colon to the end of the label name.
  - b. Under Region, select **Audit Information**.
  - c. Click **Apply Changes**.
4. Click the **Run Page** icon in the upper right corner.

[Figure 4–9](#) demonstrates how these items would display in a running page.

**Figure 4–9 Audit Information Region**

Audit Information	
* Record Create Date:	<input type="text"/>
Record Update Date:	<input type="text"/>

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. Navigate to the Page Definition for page 900. Click **Edit Page 900** on the Developer toolbar.
2. Under Regions, select **Audit Information**.
3. Scroll down to User Interface

4. From the Template list, select **Hide and Show Region**.
5. Click **Apply Changes**.

## Adding a Region Header and Footer

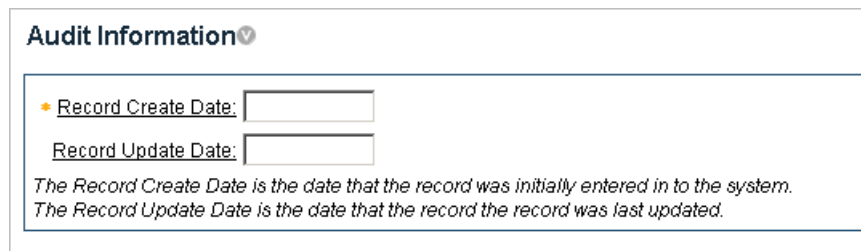
Regions can have headers and footers. Headers and footers typically contain text or HTML that displays at either the top or bottom of the region.

To add a region footer:

1. Navigate to the Page Definition for page 900.
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 the record was last updated.</i>
```
5. Click **Apply Changes**.
6. Run the page by clicking the **Run Page** icon in the upper right corner.
7. Expand the Audit Information region as shown in [Figure 4–10](#).

**Figure 4–10 Audit Information Region with Footer**



As shown in [Figure 4–10](#), the text of the footer is wrapped with the italic HTML tag and there is an imbedded a 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, P900\_EMP\_ID)

To display the Audit Information region conditionally:

1. Navigate to the Page Definition for page 900. Click **Edit Page 900** on the Developer toolbar.
2. Under Regions, select **Audit Information**.
3. When the Region Definition appears, 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:
 

```
P900_EMP_ID
```
5. Click **Apply Changes**.

## Adding Another Region for HTML Text

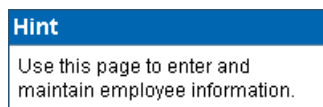
You have the option of displaying regions in columns as well as in rows. This exercise explains how to create another region to display hint text to the user.

To create a region to display hint text:

1. Navigate to the Page Definition for page 900.
2. Under Regions, click the **Create** icon.
3. For the region type, select **HTML** and click **Next**.
4. Select the type of HTML region container. Select **HTML** and click **Next**.
5. For Display Attributes:
  - a. In Title, enter `Hint`.
  - b. From Region Template, select **Sidebar Region**.
  - c. From Display Point, select **Page Template Region Position 3**.
  - d. For Sequence, accept the default.
  - e. From Column, select **3**.
  - f. Click **Next**.
6. In Enter HTML Text Region Source, enter the following:
 

```
Use this page to enter and<br/>
maintain employee information.
```
7. Click **Create Region**.
8. Click the **Run Page** icon. Note the new region on the left of the page as shown in [Figure 4–11](#).

**Figure 4–11** *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, navigate 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](#)

## 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. Navigate to the Page Definition for page 900. Click **Edit Page 900** on the Developer toolbar.
2. Under Items, select **P900\_EMP\_PART\_OR\_FULL\_TIME**.
3. From Display As, select **Radiogroup**.
4. Scroll down to Label. For Label, remove the label text. (It will be redundant.)
5. Scroll down to List of Values.
6. Under List of Values, create a static list of values:
  - a. From Named LOV, select **Select Named LOV**.
  - b. In 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 that being inserted into the database will be either F or P.

7. Scroll back to the top of the page.
8. Click **Apply Changes**.
9. Run the page by clicking the **Run Page** icon in the upper right corner. The modified form appears as shown in [Figure 4-12](#).

**Figure 4-12** Part or Full-time item Changed to a Radio Group

The screenshot shows a web form with the following elements:

- Buttons: **Cancel** and **Create**
- Field: **\* First Name** (text input)
- Radio Group:
  - Full-time
  - Part-time
- Field: **Salary** (text input)
- Field: **Department** (text input)
- Field: **Hire Date** (calendar icon)

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

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

1. Navigate to the Page Definition for page 900. Click **Edit Page 900** on the Developer toolbar.
2. Under Items, select **P900\_EMP\_PART\_OR\_FULL\_TIME**.
3. Scroll down to Lists of Values.



4. In Columns, enter 2.
5. Scroll up to the top of the page and click **Apply Changes**.

By changing this setting to match the number of valid values (that is, Full-time and Part-time), the values display side by side.

## Change an Item to a Select List

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

To change Department to a select list:

1. Navigate to the Page Definition for page 900.
2. Under Items, select **P900\_EMP\_DEPT**.
3. From Display As, 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 Lists of Values.
5. Under Lists of Values, create a static list of values:
  - a. From Named LOV, select **Select Named LOV**
  - b. In 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. Scroll up to the top of the page and click **Apply Changes**.
7. Run the page by clicking the **Run Page** icon.

The revised form appears as shown in [Figure 4-13](#) on page 4-16.

**Figure 4–13 Department Changed to a Select List**

The screenshot shows a form with several fields. The 'Department' field is a dropdown menu that is currently open, displaying a list of options: '- No Assignment -' (which is highlighted in blue), 'ACCOUNTING', 'HR', 'MANUFACTURING', and 'SALES'. Other fields visible include 'First Name' (with a red asterisk), 'Salary', 'Hire Date', 'Special Information', and 'Telecommute'.

## 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 Telecommute to a check box:

1. Navigate to the Page Definition for page 900. Click **Edit Page 900** on the Developer toolbar.
2. Under Items, select **P900\_EMP\_TELECOMMUTE**.
3. From Display As, select **Checkbox**.
4. Under Displayed:
  - a. In Sequence, enter 55.
  - b. From Begin on New Line, select **Yes**
5. Scroll down to List of Values.
6. To have the label precede the check box:
  - a. From Named LOV, select **Select Named LOV**
  - b. In 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. Scroll up to the top of the page and click **Apply Changes**.
8. Run the page by clicking the **Run Page** icon.

## About Label Templates

You can control the look of an item label by using a label template. *Sample Application* includes the following label templates:

- No Label
- Required Label

The Required Label template prepends a red asterisk (\*) to the label. You may also create your own label templates to control the look of labels using different fonts, borders, backgrounds, and images.

To change to a different label template:

1. Navigate to the Page Definition for page 900.
2. Under Items, select an item name.
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 P900\_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. Navigate to the Page Definition for page 900.
2. Click the heading **Buttons**.
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 `:::P900_EMP_ID:1` to the end of the URL as shown in the following example:

```
http://apex.oracle.com/pls/otn/f?p=9659:900:1284393467360777225:::::P900_EMP_ID:1
```

This will pass the value 1 to the item P9000\_EMP\_ID. If you run the page note that the Delete and Apply Changes buttons now display. The Create button appeared previously because the page was expecting a new record to be created. Also note a Record Create Date now appears.

**See Also:** "Understanding URL Syntax" in *Oracle Database Express Edition Application Express 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. Navigate to the Page Definition.
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.

---

---

## How to Work with Check Boxes

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

This tutorial illustrates the different ways you can create check boxes within the demonstration application, *Sample Application* and explains how to reference and process the values of checked boxes.

This section contains the following topics:

- [About Sample Application](#)
- [Creating a Single Value Check Box on a Form](#)
- [Creating Multi Value Check Box to Filter Content](#)
- [Adding Check Boxes to Each Row in the Report](#)

### About Sample Application

Application Builder installs with a number of demonstration applications. In this exercise you create check boxes within the demonstration application, *Sample Application*.

To see if *Sample Application* is installed:

1. Log in to the Database Home Page using your database username and password.
2. Click the down arrow on the right side of the **Application Builder** icon.
3. From the menu, select **Demonstrations**.

The Demonstration Applications page appears, displaying links to demonstration applications.

4. Locate *Sample Application* and check the Status column:
  - a. If the Status column displays **Installed**, return to the Database Home Page.
  - b. If the Status column displays **Not Installed**, select **Install** in the Action column. Follow the on-screen instructions.

### Creating a Single Value Check Box on a Form

As shown in [Figure 5-1](#) on page 5-2, the Add/Modify Products form enables users to add new products to the database.

**Figure 5–1 Add/Modify Products Form in Sample Application**

In this exercise you change the Product Available radio group to a check box and then change the position of the label.

Topics in this section include:

- [Change Product Available Radio Group to a Check Box](#)
- [Alter the Check Box Position](#)
- [Change Default Check Box Behavior](#)

## Change Product Available Radio Group to a Check Box

To change the Product Available radio group to a check box:

1. On the Database Home Page, click the **Application Builder** icon.
2. Click **Sample Application**.

The Application home page appears.

3. Navigate to the Page Definition for page 6:
  - a. In the Page field, enter 6 and click **Go**.
  - b. Click **6 - Add/Modify**.

The Page Definition for page 6 appears. The Product Available radio group is an item named P6\_PRODUCT\_AVAIL.

4. Under Items, select **P6\_PRODUCT\_AVAIL**.
5. From Display As, select **Checkbox**.
6. Scroll down to List of Values.
7. Under List of Values:

- a. From Named LOV, select **Select Named LOV**.
- b. In List of values definition, enter:

```
STATIC; ;Y
```

This list of values is a static list. In this example, the display value is null and the return value is Y. If a display value were provided, it would appear to the left of the check box and could be used in place of the label.

8. Scroll back to the top of the page and click **Apply Changes**.

The Page Definition appears.

9. Click the **Run Page** icon in the upper right corner. (See [Figure 5–2](#)).

**Figure 5–2 Run Page Icon**



10. If prompted to enter a username and password:
  - a. For User Name, enter either demo or admin.
  - b. For Password, enter your database user name in lowercase letters.
  - c. Click **Login**.

Sample Application appears.

11. Navigate to page 6. Select **Edit Page 1** from the Developer Toolbar at the bottom of the page.

The Page Definition appears.

12. In the Page field, enter **6** and click **Go**.
13. Click the **Run Page** icon in the upper right corner.

As shown in [Figure 5–3](#), note that the Product Available item now displays as a check box.

**Figure 5–3 Product Available Item as a Check Box**

## Alter the Check Box Position

Next, you move the check box label to the right side of the check box.

To change the position of the check box label:

1. Navigate to the Page Definition for page 6. Select **Edit Page 6** from the Developer Toolbar.
2. Under Items, select **P6\_PRODUCT\_AVAIL**.
3. Scroll down to Label.
4. For Label, delete the text `Product Available`.
5. Scroll down to Lists of Values.
6. Under List of Values, change the List of values definition to:

STATIC:Product Available;Y

7. Scroll back to the top of the page and click **Apply Changes**.
8. Run the page by clicking the **Run Page** icon.

As shown in [Figure 5–4](#), note that the label Product Available now displays to the right of the check box.

**Figure 5–4 Product Available Label Moved to the Right**

Removing the label and adding the display value to the LOV causes the Application Express engine to render the check box first and then the display value, Product Available.

## Change Default Check Box Behavior

In certain circumstances, you may want a check box to be enabled by default. You can accomplish this by setting the default value attribute of the check box item. One disadvantage of this approach is that you need to perform some extra steps to disable it. Because of the way you defined the Product Available check box, it is virtually impossible to disable it.

Consider the following example:

1. Within Sample Application, navigate to the Home page by selecting the **Home** tab.
2. From the Tasks list, click **Add a New Product**.
3. On the Add/Modify Products page:
  - a. Fill in the required fields (fields marked with an asterisk).
  - b. Disable the **Product Available** check box.
  - c. Click **Create**.
4. Disable the Product Available check box again and click **Apply Changes**.  
The Products page appears.
5. Select the **Edit** icon for product you just added.

Notice that the Product Available check box is enabled even though you disabled it twice when you added the product. This behavior results from the fact:

- The Product Available check box has a default value of Y.
- When Product Available is NULL, it defaults to the default value Y which enables the check box.



## Add a Computation

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. Navigate to the Page Definition for page 6. Select **Edit Page 6** from the Developer Toolbar.
2. Under Page Processing, Computations, click the **Create** icon.
3. For Location, select **Item on This Page** and click **Next**.
4. For Item:
  - a. For Compute Item, select **P6\_PRODUCT\_AVAIL**.
  - b. For Computation Point, select **After Submit**.
  - c. For Computation Type, select **Static Assignment**.
  - d. Click **Next**.
5. For Computation, enter the following and click **Next**.

N

---



---

**Note:** Since this item is a check box, set the value of the item to something other than Y or NULL.

---



---

Next, create a condition that controls when the computation executes.

6. On Condition:
  - a. For Condition Type, select **Value of Item in Expression 1 is NULL**.
  - b. In Expression 1, enter:

P6\_PRODUCT\_AVAIL

Because of these settings, this computation will only execute when the value of the check box item, P6\_PRODUCT\_AVAIL is NULL.

7. Click **Create**.

To test the new computation:

1. Click the **Run Page** icon.  
Sample Application appears.
2. Navigate to the Home page by selecting the **Home** tab.
3. From the Tasks list, select **Add a New Product**.
4. On the Add/Modify Product page:
  - a. Fill in the required fields (fields marked with an asterisk).
  - b. Disable the **Product Available** check box.
  - c. Click **Create**.
5. Disable the Product Available check box again and click **Apply Changes**.  
The Products page appears.

6. Select the **Edit** icon for product you just added.

Note that the Product Available check box is not selected.

## Creating Multi Value Check Box to Filter Content

In the next exercise, you create a multi value check box on the Product page. This check box enables users to filter the report by selecting a category.

Topics in this section include:

- [Create a Multi Value Check Box](#)
- [Alter Check Box Display Values](#)
- [Change Where the Check Boxes Display](#)
- [Create a Go Button to Submit the Page](#)
- [Adjust Default Check Box Behavior](#)

### Create a Multi Value Check Box

To a create muli value check box:

1. Navigate to the Page Definition for page 3:
  - a. Click the Edit Page link on the Developer Toolbar.
  - b. Enter 3 in the Page field and click **Go**.

The Page Definition for page 3 appears.

2. Under Items, click the **Create** icon.
3. For Item Type, select **Check Box** and click **Next**.
4. On Display Position and Name:
  - a. In Item Name, enter P3\_SHOW.
  - b. For Sequence, accept the default.
  - c. For Region, select **Products (1) 10**.
  - d. Click **Next**.
5. For List of Values:
  - a. For Named LOV, accept the default.
  - b. For Display Null Option, select **No**.
  - c. In List of Values Query, enter:

```
SELECT distinct category a, category b
FROM demo_product_info
ORDER BY 1
```

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

- d. Click **Next**.
6. On Item Attributes:
  - a. For Label, remove the default text by clicking **Clear**.
  - b. Accept the remaining defaults and click **Next**.

7. For Source:
  - a. For Item Source, select **Static Assignment**.
  - b. For Item Source Value, enter:

Audio:Computer:Phones:Video

When a multi value check box is submitted, the value of the item is a colon delimited string of values. Using this string as the source ensures all boxes are checked when Application Express engine renders the page.

- c. Click **Create Item**.

## Alter Check Box Display Values

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

1. Navigate to the Page Definition for page 3.
2. Under Items, select **P3\_SHOW**.
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**.

## Change Where the Check Boxes Display

Next, you edit attributes so that the category check boxes display above the report. To accomplish this, you need to change the Display Point attribute of the region associated with P3\_SHOW.

To change the display point of the Products region:

1. Navigate to the Page Definition for page 3.
2. Under Region, select **Products**.
3. From Display Point, select **Page Template Body (3. items above region content)**.
4. Click **Apply Changes**.

If you ran page 3 now, you would notice the category check boxes display vertically. Next, you need to change the display so that the category check boxes display horizontally.


To alter where the category check boxes display:

1. Navigate to the Page Definition for page 3.
2. Under Items, select **P3\_SHOW**.
3. Scroll down to List of Values.
4. In Columns, enter 4.
5. Scroll back to the top of the page and click **Apply Changes**.

The Page Definition for page 3 appears.

6. Click the **Run Page** icon in the upper right corner.  
 Note the check boxes display horizontally at the top of the page as shown in [Figure 5-5](#).

**Figure 5-5 Category Check Boxes Moved to Top of Page**

Products		
<input checked="" type="checkbox"/> Audio <input checked="" type="checkbox"/> Computer <input checked="" type="checkbox"/> Phones <input checked="" type="checkbox"/> Video		
	Name	Description
	3.2 GHz Desktop PC	All the options, this machine is loaded!

## Create a Go Button to Submit the Page

In order for the report to be driven by the product category check boxes, you need to submit the page.

To create a button to submit the page:

1. Navigate to the Page Definition for page 3.
2. Under Buttons, click the **Create** icon.
3. For Button Region, select **Products (1) 10** and click **Next**.
4. For Position, select **Create a button displayed among this region's items** and click **Next**.
5. For Button Attributes:
  - a. For Button Name, enter P3\_GO.
  - b. In Sequence, enter 40.
  - c. Accept the remaining defaults and click **Create Button**.

Next, you need to create a branch to tell the Application Express engine where to go after the user clicks the **Go** button.

To create a branch to page 3:

1. Navigate to the Page Definition for page 3.
2. Under Branches, click the **Create** icon.
3. For Branch Point and Branch Type, accept the defaults and click **Next**.
4. For Page, select **3**, accept the remaining defaults, and click **Next**.
5. Click **Create Branch**.

In order to generate a report based on the category check box values, you need to change the report query.

To edit the report query:

1. Under Regions, select **Products**.
2. Scroll down to Source.
3. In Region Source, locate the WHERE clause.
4. Change the WHERE clause to read:

```
WHERE p.image_id = i.image_id (+)
AND instr(':'||:P3_SHOW||:',',category) > 0
```







5. Scroll back to the top of the page and click **Apply Changes**.
6. Run the page by clicking the **Run Page** icon.

As shown in [Figure 5–6](#), notice you can filter the report by selecting a category check box and clicking **Go** at the top of the page.

**Figure 5–6 Product Page with Category Check Boxes and Go Button**

**Products**

Audio
  Computer
  Phones
  Video

	Name	Description	Category	Available	Price	Image
	3.2 GHz Desktop PC	All the options, this machine is loaded!	Computer	Y	\$1,200.00	
	512 MB DIMM	Expand your PCs memory and gain more performance	Computer	Y	\$200.00	
	Ultra Slim Laptop	The power of a desktop in a portable design	Computer	Y	\$1,999.00	

1 - 3

[Export to Spreadsheet](#)

## Adjust Default Check Box Behavior

Although the category check boxes correctly filter the content on page 3, 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), 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. Navigate to the Page Definition for page 3.
2. Under Page Processing, Computations, click the **Create** icon.
3. For Item Location, select **Item on This Page** and click **Next**.
4. For Item:
  - a. For Compute Item, select **P3\_SHOW**
  - b. For Computation Point, select **After Submit**.
  - c. For Computation Type, select **Static Assignment**.
  - d. Click **Next**.
5. For Computation:
  - a. In Computation, enter the following:
 

```
none(bogus_value)
```
  - b. Click **Next**.

---



---

**Note:** A static assignment of an item needs to be something other than Y or NULL.

---



---

Next, create a condition that controls when the computation executes.

6. On Condition:

- a. For Condition Type, select **Value of Item in Expression 1 is NULL**.
- b. In Expression 1, enter:

P3\_SHOW

As a result of these settings, this computation will only execute when the value of the check box item, P3\_SHOW is NULL.

7. Click **Create**.

Run the page again and deselect all the category check boxes and click **Go**. Notice that this time the report contains the expected result (no returned records).

## Adding Check Boxes to Each Row in the Report

In the next exercise, you add a delete check box to each row in the Products report. To accomplish this you edit the report query and make a call to HTMLDB\_ITEM package.

HTMLDB\_ITEM is a supplied package for generating certain form elements dynamically. In this instance, you use HTMLDB\_ITEM.CHECKBOX to generate check boxes in the Products 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 HTMLDB\_APPLICATION.G\_F01 to HTMLDB\_APPLICATION.G\_F50 based on the p\_idx parameter value that was passed in.

Topics in this section include:

- [Call HTMLDB\\_ITEM.CHECKBOX](#)
- [Add a Button to Submit Check Box Array Values](#)
- [Add a Process](#)

**See Also:** "HTMLDB\_ITEM" in *Oracle Database Express Edition Application Express User's Guide*.

### Call HTMLDB\_ITEM.CHECKBOX

To edit the query to call HTMLDB\_ITEM.CHECKBOX:

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






```
SELECT p.product_id edit_product, p.product_id view_product_id,
htmldb_item.checkbox(1,p.product_id) del,
p.product_name, p.product_description, p.category, p.product_avail, p.list_
price,
''
img
```

```
FROM demo_product_info p, demo_images i
WHERE p.image_id = i.image_id (+)
AND instr(':'||:P3_SHOW||':',category) > 0
```

HTMLDB\_ITEM is an Oracle Application Express supplied package you can use to generate certain form elements dynamically. Note the value passed in for p\_idx in the above example is 1. You reference the check box values using the global variable HTMLDB\_APPLICATION.G\_F01 later on.

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 until the DEL column is directly beneath VIEW\_PRODUCT\_ID. (See [Figure 5-7](#)).

**Figure 5-7 Report Column Attributes Page**

	Alias	Link	Edit		Heading	Column Alignment	Heading Alignment
	EDIT_PRODUCT	✓		▼ ▲	&nbsp;	center	center
	VIEW_PRODUCT_ID			▼ ▲	ID	right	center
	DEL			▼ ▲	Del	left	center
	PRODUCT_NAME			▼ ▲	Name	left	center
	PRODUCT_DESCRIPTION			▼ ▲	Description	left	center

8. Click **Apply Changes**.

### Add a Button to Submit Check Box Array Values

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

1. Navigate to the Page Definition for page 3.
2. Under Buttons, click the **Create** icon.
3. For Button Region, select **Products 1 (10)** and click **Next**.
4. For Position, select **Create a button in a region position** and click **Next**.
5. For Button Attributes:
  - a. For Button Name, enter DELETE\_PRODUCTS.
  - b. For Label, enter Delete Products.
  - c. Accept the remaining defaults and click **Next**.
6. For Button Template, accept the default selection and click **Next**.
7. For Display Properties:
  - a. For Position, select **Top of Region**.
  - b. Accept the remaining defaults and click **Next**.
8. For Position, select **Top of Region** and click **Next**.
9. For Branching, select **3 Products** 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. For Name, enter `Delete Products`.
  - b. For Point, select **On Submit - After Computations and Validations**.
  - c. Click **Next**.
4. Enter the following PL/SQL process and click **Next**:

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

HTMLDB\_ITEM is an Oracle Application Express supplied package you can use to generate certain form elements 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 HTMLDB\_APPLICATION.G\_F01 to HTMLDB\_APPLICATION.G\_F50. In this exercise, the value passed in for product\_id in EMPNO column is 1, so you reference the EMPNO column values using the global variable HTMLDB\_APPLICATION.G\_F01.

5. On Messages:
  - a. In Success Message, enter:  
`Product(s) deleted.`
  - b. In Failure Message, enter:  
`Unable to delete product(s).`
  - c. Click **Next**.
6. Click **Create Process**.
7. Run the page by clicking the **Run Page** icon in the upper right corner.

Notice the Delete Products button appears above the report as shown in [Figure 5–8](#) on page 5-13. To remove a product from the report, select the **Del** check box and click **Delete Products**.









**Figure 5–8 Products Report with Delete Products Check Box**

**Products**

[Delete Products](#)

Audio  Computer  Phones  Video [Go](#)

	Del	Name	Description	Category	Available	Price	Image
	<input type="checkbox"/>	3.2 GHz Desktop PC	All the options, this machine is loaded!	Computer	Y	\$1,200.00	
	<input type="checkbox"/>	512 MB DIMM	Expand your PC's memory and gain more performance	Computer	Y	\$200.00	
	<input type="checkbox"/>	Ultra Slim Laptop	The power of a desktop in a portable design	Computer	Y	\$1,999.00	

1 - 3



---

---

## 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 (the 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.

This tutorial illustrates how to call a Web service from within an Oracle Application Express application.

Topics in this section include:

- [About Creating Web Service References](#)
- [Creating a New Application](#)
- [Specifying an Application Proxy Server Address](#)
- [Searching a UDDI Registry for a Business Name](#)
- [Searching a UDDI Registry for a Service Name](#)

---

---

**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 Database Express Edition Application Express 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. Each Web service reference is based on a Web Services Description Language (WSDL) document that describes the target Web service. When

you create a Web service reference, the wizard analyzes the WSDL and collects all the necessary information to create a valid SOAP message.

When you create a Web service reference you need to decide how to locate the WSDL. You can locate a WSDL two ways:

- By searching a Universal Description, Discovery, and Integration (UDDI) registry for either a service name or business name.

A UDDI registry is a directory where businesses register their Web services.

- By entering the URL to the WSDL document.

In this tutorial, you create Web service references by searching a UDDI registry.

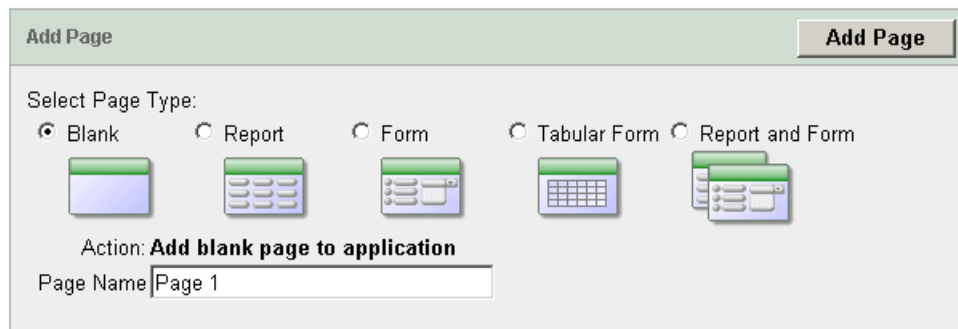
## Creating a New Application

First, create a new application.

To create an application:

1. On the Database Home Page, click the **Application Builder** icon.
2. When the Application Builder home page appears, click **Create**.
3. For Method, select **Create Application** and click **Next**.
4. For Name:
  - a. For Name, enter `Web Services`.
  - b. For Create Application, select **From Scratch**.
  - c. Accept the remaining defaults and click **Next**.
5. Add a blank page:
  - a. Under Select Page Type, select **Blank** as shown in [Figure 6-1](#).

**Figure 6-1 Add Page**



- b. In Page Name, enter `Web Services` and 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 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 Builder 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 for an application:

1. On the Application Builder home page, click **Edit Attributes**.
2. Click **Edit Standard Attributes**.
3. Under Name, enter the proxy server in Proxy Server.
4. Click **Apply Changes**.

## Searching a UDDI Registry for a Business Name

In this exercise you create a form which displays stock quotes based on a stock symbol you provide. You will create a Web service reference by searching the UDDI registry for a business name:

---

---

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

---

---

To create a Web service reference by searching for a business name:

1. On Application Builder home page, click **Shared Components**.
2. Under Logic, select **Web Service References**.  
The Web Service References page appears.
3. Click **Create**.
4. When prompted whether to search a UDDI registry to find a WSDL, select **Yes** and click **Next**.
5. For UDDI Location, select **IBM UDDI** and click **Next**.
6. On Search:
  - a. For Search Type, select **Business Name**.
  - b. In Name, enter:  
`%xMethods%`
  - c. Click **Search**.  
Matching Services appear.
  - d. Under Matching Services, select **xMethods Delayed Stock Quotes**.
  - e. Click **Next**.

A summary page appears describing the selected Web service.

7. Review your selection and click **Next** to continue.  
The URL to the WSDL document displays in the WSDL Location field.
8. Click **Finish**.

## Create a Form to Display a Stock Quote

Next, you create a form to display a stock quote based on the stock symbol you provide.

To create a form after creating a Web Service Reference:

1. On the Create Web Service Reference page, click **Create Form on Web Service**.
2. On Web Service Reference and Operation:
  - a. For Web Service Reference, select **XMethods Delayed Stock Quotes**.
  - b. For Operation, select **getQuote**.
  - c. Click **Next**.
3. For Page and Region Attributes:
  - a. For Page and Page Name, accept the defaults.
  - b. In Region Title, enter `Get Stock Quote`.
  - c. For Region Template, accept the default.
  - d. Click **Next**.
4. For Input Items, accept the defaults and click **Next**.
5. For Output Items:
  - a. For Item Label, enter `Stock Price`.
  - b. Accept the other defaults.
  - c. Click **Finish**.
6. Click **Run Page** to view the form.
7. On the Log in page, enter the User Name and Password for your database user account and click **Login**.

A form resembling [Figure 6-2](#) appears.

**Figure 6-2** *Get Stock Quote Form*

The screenshot shows a web form titled "Get Stock Quote" with a "Submit" button in the top right corner. Below the title, there are two input fields. The first is labeled "Symbol" and the second is labeled "Stock Price". Both fields are currently empty.

8. Test the form. In Symbol, enter a stock symbol (such as ORCL) and click **Submit**.  
The associated stock price displays in the Stock Price field.
9. Click **Edit Application** on the Developer toolbar to return to the Application Builder home page.

## Searching a UDDI Registry for a Service Name

Next, you create a form that displays market futures. In this exercise you create a Web service reference by searching a UDDI registry for a service name. Then, you create a form and report.

---

---

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

---

---

To create a new Web service by searching for a service name:

1. Navigate to the Application Builder home page.
2. Click **Shared Components**.  
The Shared Components page appears.
3. Under Logic, select **Web Service References**.  
The Web Service References page appears.
4. Click **Create**.
5. When prompted whether to search a UDDI registry to find a WSDL, select **Yes** and click **Next**.
6. For UDDI Location, select **XMethods UDDI** and click **Next**.
7. For Search:
  - a. For Search Type, select **Service Name**.
  - b. In Name, enter `xignite` and click **Search**.  
This is a search engine for market news.
  - c. Under Matching Services, select `XigniteFutures`.
  - d. Click **Next**.  
A summary page appears describing the selected Web service.
8. Review your selection and click **Next** to continue.  
The WSDL Location field displays the URL to the WSDL document.
9. Click **Finish**.

The Web service reference, `XigniteFutures`, 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 create a form and report after creating a Web Service Reference:

1. On the Create Web Service Reference page, select **Create Form and Report on Web Service**.
2. For Choose Service and Operation:
  - a. For Web Service Reference, select **XigniteFutures**.

- b. For Operation, select **ListFuturesByExchange**.
  - c. Click **Next**.
3. For Page and Region Attributes:
  - a. Change Form Region Title to `List Futures By Exchange`.
  - b. Accept the other defaults and click **Next**.
4. For Input Items:
  - a. For P2\_USERNAME, P2\_PASSWORD, P2\_TRACER, select **No** in the Create column.
  - b. For P2\_EXCHANGE, accept the default **Yes** in the Create column.
  - c. Click **Next**.
5. For Web Service Results:
  - a. For Temporary Result Set Name (Collection), accept the default.
  - b. Result Tree to Report On, select **Future (tns:Future)**.
  - c. Click **Next**.
6. For Result Parameters to Display, select all the parameters and click **Finish**.
7. Click **Run Page** to view the form.
8. If a Log in page appears, enter the User Name and Password for your database user account and click **Login**.

A form and report resembling [Figure 6–3](#) appears. Notice that the List Futures by Exchange Form on the top of the page contains a data entry field and a submit button, but the Results Report does not contain any data.

**Figure 6–3 List Futures by Exchange Form and Report without Data**

The screenshot shows a web application interface. At the top left, the text 'TWINTERS' is displayed. Below it, the title 'List Futures By Exchange' is shown in blue. To the right of the title is a 'Submit' button. Underneath the title is a label 'Exchange' followed by an empty text input field. Below this is a section titled 'Results' in blue, with the text 'No data found.' underneath. At the bottom of the page, there is a navigation bar with several buttons: 'Edit Application', 'Edit Page 3', 'New', 'Session', 'Debug', and 'Show Edit Links'.

9. Test the form. In Exchange, enter NYMEX and click **Submit**.  
 The report at the bottom of the page should resemble [Figure 6–4](#) on page 6-7. The report lists the symbol, name, month, and year of futures from the New York Mercantile Exchange (NYMEX).



**Figure 6-4 List Futures by Exchange Form and Report with Data**

TWINTERS

---

**List Futures By Exchange** Submit

Exchange

**Results**

Symbol	Name	Month	Year	Exchange
CL	Crude Oil	2	2005	NYMEX
F0	Heating Oil/Crude	2	2005	NYMEX
F5	Unleaded/Crude	2	2005	NYMEX
HO	Heating Oil	2	2005	NYMEX
HU	Gasoline Unleaded	2	2005	NYMEX
JM	PJM Monthly	2	2005	NYMEX
NG	Natural Gas	2	2005	NYMEX
PL	Platinum	2	2005	NYMEX
PN	Propane	2	2005	NYMEX
QG	Natural Gas emiNY	2	2005	NYMEX
QL	Coal Futures	2	2005	NYMEX
QM	Crude Oil emiNY	2	2005	NYMEX
SC	Brent Crude Oil	2	2005	NYMEX

1 - 13



---

---

## 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 communicate 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.

In Application Builder a stacked bar chart is only available as an SVG chart. To create a stacked bar chart, you can create the chart as a stacked bar chart, or you can create a regular (non-HTML) bar chart and then add queries to it.

This tutorial describes how to create a stacked bar chart using existing data within the demonstration application, *Sample Application*.

This section contains the following topics:

- [About Sample Application](#)
- [Creating a Stacked Bar Chart](#)
- [Adding Additional Series](#)
- [Changing the Chart Format](#)
- [Viewing the Chart](#)

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

### About Sample Application

Application Builder installs with a number of demonstration applications. In this exercise you create a stacked bar chart within the demonstration application, *Sample Application*.

To see if *Sample Application* is installed:

1. Log in to the Database Home Page using your database username and password.
2. Click the down arrow on the right side of the **Application Builder** icon.
3. From the menu, select **Demonstrations**.

The Demonstration Applications page appears, displaying links to demonstration applications.

4. Locate *Sample Application* and check the Status column:
  - a. If the Status column displays **Installed**, return to the Database Home Page.

- b. If the Status column displays **Not Installed**, select **Install** in the Action column. Follow the on-screen instructions.

## Creating a Stacked Bar Chart

To create the initial report, 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 Sample Application.

The chart will display the sum for sales by product category from within Sample Application. It will contain sales for the twelve months prior to the current month. In the following exercise, you create four queries (called series) for each of the product categories (phones, computers, audio, and video).

To create a new page:

1. On the Database Home Page, click the **Application Builder** icon.
2. Select **Sample Application**.
3. Click **Create Page**.
4. Select **Chart** and click **Next**.
5. Select **Stacked Bar, Vertical** and click **Next**.
6. For Page Attributes:
  - a. For Page, enter 750.
  - b. For Page Name, enter Revenue by Category.
  - c. For Page Template, accept the default.
  - d. For Region Name, enter Revenue by Category.
  - e. For Chart Color Theme, accept the default.
  - f. Click **Next**.
7. For Tab Options, accept the default **Do not use Tabs** and click **Next**.
8. For Query:
  - a. For Series Name, enter Phones.
  - b. In SQL, enter:

```

SELECT NULL l,
       sales_month,
       revenue
FROM (
SELECT TO_CHAR(o.order_timestamp, 'Mon YYYY') sales_month,
       SUM(oi.quantity * oi.unit_price) revenue,
       TO_DATE(to_char(o.order_timestamp, 'Mon YYYY'), 'Mon YYYY') sales_
month_order
FROM DEMO_PRODUCT_INFO p,
     DEMO_ORDER_ITEMS oi,
     DEMO_ORDERS o
WHERE o.order_timestamp <= (trunc(sysdate, 'MON')-1)
     AND o.order_timestamp > (trunc(sysdate-365, 'MON'))
     AND o.order_id = oi.order_id
     AND oi.product_id = p.product_id
     AND p.category = 'Phones'
GROUP BY TO_CHAR(o.order_timestamp, 'Mon YYYY')
ORDER BY sales_month_order
    
```

)

The syntax for the select of a chart is:

```
SELECT link, label, value
```

You must have all three and only three items in your select. Because there is no appropriate page to link to in this example, the link is defined as null.

Note that you cannot include an `ORDER BY` in the `SELECT` statement for a series in a stacked chart because the information displays in alphabetical order. Displaying dates in alphabetical order does not make sense, for example, October would display before September. A more appropriate approach would be to display the data in chronological order. To display the data in chronological order, you need to order the data inside a nested `SELECT`.

- c. For When No Data Found Message, enter:

No orders found in the past 12 months.

- d. Click **Next**.

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

## Adding Additional Series

Once you have created the new page with a region defining the query, you need to add the series. In the following exercise, you add a series for each of the categories of product in the application (that is, computers, audio, and video). Note you have already created the phones category.

To add a series for computers category:

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

The Page Definition for page 750 appears.

2. Under Regions, select **Chart** to the left of Revenue by Category as shown in [Figure 7-1](#).

**Figure 7-1 Chart Link on the Page Definition**



The Chart Attributes page appears. Scroll down to Chart Series. Notice that only one series appears.

3. Add a chart series for Computer.
- Scroll down to Chart Series and click **Add Series**.
  - For Series Name, enter `Computer`.
  - Scroll down the Series Query.
  - In SQL, enter:

```
SELECT NULL 1,
```

```

        sales_month,
        revenue
    FROM (
SELECT TO_CHAR(o.order_timestamp, 'Mon YYYY') sales_month,
       SUM(oi.quantity * oi.unit_price) revenue,
       TO_DATE(to_char(o.order_timestamp, 'Mon YYYY'), 'Mon YYYY') sales_
month_order
    FROM DEMO_PRODUCT_INFO p,
         DEMO_ORDER_ITEMS oi,
         DEMO_ORDERS o
    WHERE o.order_timestamp <= (trunc(sysdate, 'MON')-1)
        AND o.order_timestamp > (trunc(sysdate-365, 'MON'))
        AND o.order_id = oi.order_id
        AND oi.product_id = p.product_id
        AND p.category = 'Computer'
    GROUP BY TO_CHAR(o.order_timestamp, 'Mon YYYY')
    ORDER BY sales_month_order
    )

```

Note that this SQL matches the previous series. The only difference is that the category in the WHERE clause.

- 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**.

4. Add a chart series for Audio:

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

- b. For Series Name, enter Audio.

- c. Scroll down the Series Query.

- d. In SQL, enter:

```

SELECT NULL 1,
       sales_month,
       revenue
    FROM (
SELECT TO_CHAR(o.order_timestamp, 'Mon YYYY') sales_month,
       SUM(oi.quantity * oi.unit_price) revenue,
       TO_DATE(to_char(o.order_timestamp, 'Mon YYYY'), 'Mon YYYY') sales_
month_order
    FROM DEMO_PRODUCT_INFO p,
         DEMO_ORDER_ITEMS oi,
         DEMO_ORDERS o
    WHERE o.order_timestamp <= (trunc(sysdate, 'MON')-1)
        AND o.order_timestamp > (trunc(sysdate-365, 'MON'))
        AND o.order_id = oi.order_id
        AND oi.product_id = p.product_id
        AND p.category = 'Audio'
    GROUP BY TO_CHAR(o.order_timestamp, 'Mon YYYY')
    ORDER BY sales_month_order
    )

```

- 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**.

5. Add a chart series for Video:
  - a. Scroll down to Chart Series and click **Add Series**.
  - b. For Series Name, enter Video.
  - c. Scroll down the Series Query.
  - d. In SQL, enter:
 

```
SELECT NULL 1,
        sales_month,
        revenue
      FROM (
SELECT TO_CHAR(o.order_timestamp, 'Mon YYYY') sales_month,
        SUM(oi.quantity * oi.unit_price) revenue,
        TO_DATE(to_char(o.order_timestamp, 'Mon YYYY'), 'Mon YYYY') sales_
month_order
      FROM DEMO_PRODUCT_INFO p,
        DEMO_ORDER_ITEMS oi,
        DEMO_ORDERS o
      WHERE o.order_timestamp <= (trunc(sysdate, 'MON')-1)
        AND o.order_timestamp > (trunc(sysdate-365, 'MON'))
        AND o.order_id = oi.order_id
        AND oi.product_id = p.product_id
        AND p.category = 'Video'
      GROUP BY TO_CHAR(o.order_timestamp, 'Mon YYYY')
      ORDER BY sales_month_order
      )
```
  - 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**.

## Changing the Chart Format

Next, you enhance the appearance of the chart with axis titles by adding a region footer.

To format the y-axis:

1. Scroll down to Axes Settings.
2. In Y Axis Format, enter:
 

```
FML999G999G999G999G990
```

This formats the `sales_month` column as money, but without displaying the cents
3. Scroll up and select the **Region Definition** tab.
4. Scroll down to Header and Footer.
5. In Region Footer, enter:
 

Note: This reflects sales for the 12 months prior to the current month.
6. Scroll up to the top of the page and click **Apply Changes**.

## Viewing the Chart

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

To run the chart:

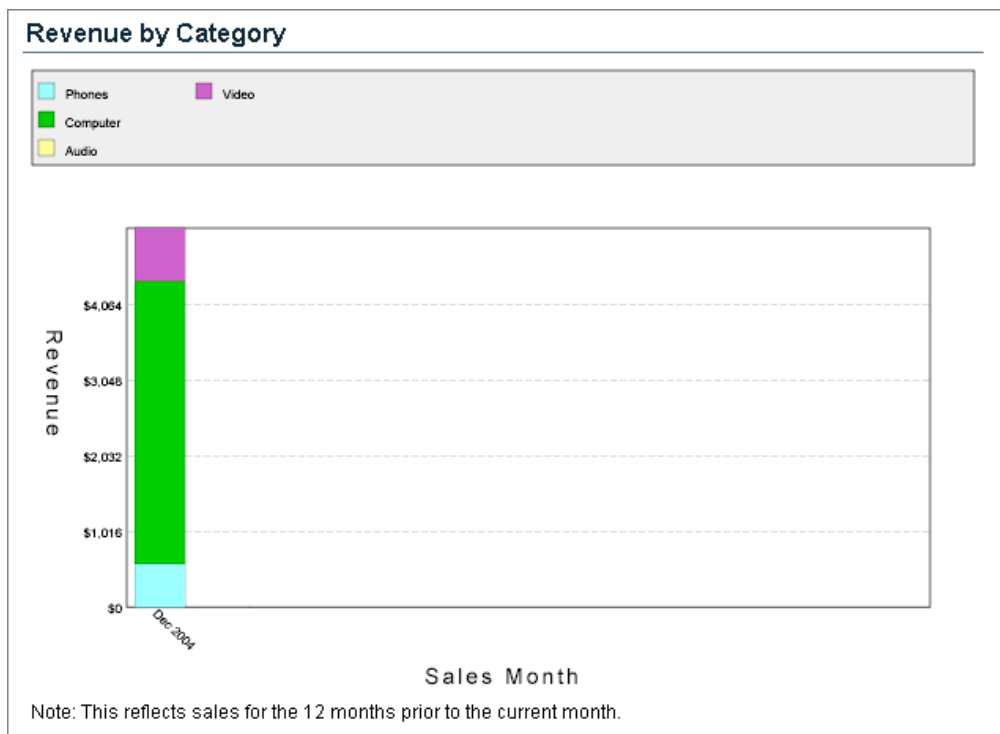
1. Click the **Run Page** icon in the upper right corner of the page. (See [Figure 7-2](#).)

**Figure 7-2 Run Page Icon**



2. If you have already run Sample Application in this session, page 750 displays. If prompted to enter a username and password:
  - a. For User Name, enter either `demo` or `admin`.
  - b. For Password, enter your database user name in lowercase letters.
  - c. Click **Login**.
  - d. Navigate to page 6. Click **Edit Page 1** on the Developer toolbar at the bottom of the page. The Page Definition appears.
    - e. In the Page field, enter `750` and click **Go**.
    - f. Click the **Run Page** icon in the upper right corner. As shown in [Figure 7-3](#), the Revenue by Category chart appears.

**Figure 7-3 Revenue by Category Bar Chart**





**Tip:** One way to navigate to a new page within a running application is to change the second parameter (the page identifier) to 750. For example, you would change:

`http://apex.oraclecorp.com/pls/apex/f?p=2046:1: ...`

to

`http://apex.oraclecorp.com/pls/apex/f?p=2046:750: ...`



---

---

# How to Upload and Download Files in an Application

Oracle Application Express applications may include 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](#)
- [Store the Document in a Custom Table](#)
- [Downloading Documents from the Custom Table](#)

## Creating an Application

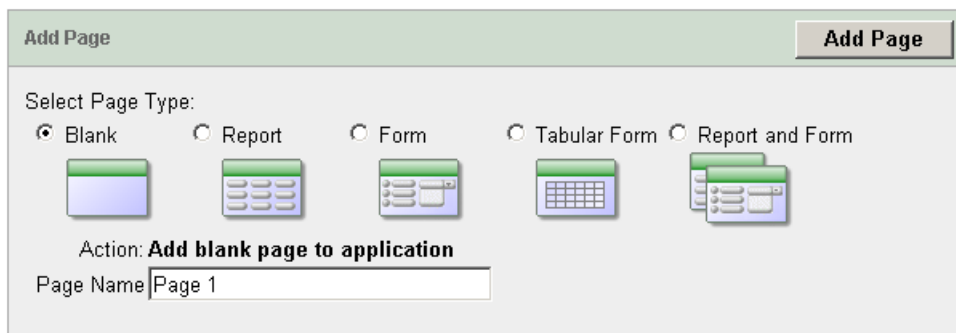
First, create a new application using the Create Application Wizard with the assumption you will include an upload form on page 2.

To create an application using the Create Application Wizard:

1. On the Database Home Page, click the **Application Builder** icon.  
The Application Builder home page appears.
2. Click **Create**.
3. Select **Create Application** and click **Next**.
4. Specify the page name.
  - a. For Name, enter Download App.
  - b. For Application, accept the default.
  - c. For Create Application, select **From scratch**.
  - d. For Schema, accept the default.
  - e. Click **Next**.
5. Add a blank page:

- a. Under Select Page Type, select **Blank** and click **Add Page** as shown in [Figure 8-1](#).

**Figure 8-1 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 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. Navigate to the Page Definition for page 1 by clicking **Page 1**.
2. Under Regions, click the **Create** icon as shown in [Figure 8-2](#) on page 8-3.

**Figure 8–2 Create Icon**

3. For Region, select **HTML** and click **Next**.
4. Select a type of HTML region container. Select **HTML** again and click **Next**.
5. For Display Attributes:
  - a. In Title, enter `Submit File`.
  - b. Accept the remaining defaults and click **Next**.
6. 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.

To create a file upload item:

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

5. For Button Template, accept the default and click **Next**.
6. On Display Properties, accept the defaults and click **Next**.
7. On Branching:
  - a. In Branch to Page, select **Page 1**.
  - b. Click **Create Button**.
8. Run the page by clicking the **Run Page** icon as shown in [Figure 8-3](#).

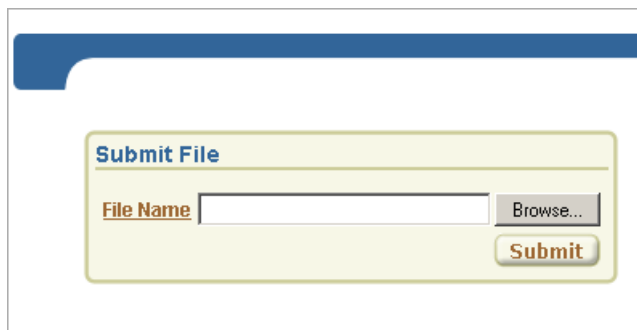
**Figure 8-3 Run Page Icon**



9. When prompted for a user name and password:
  - a. For User Name, enter the name of your database user account.
  - b. For Password, enter the password for your database user account.
  - c. Click **Login**.

When you run the page, it should look similar [Figure 8-4](#).

**Figure 8-4 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 download documents. When you use the file upload item type, the files you upload are stored in a table called `wwv_flow_file_objects$`. Every database user account has access to this table through a view called `HTMLDB_APPLICATION_FILES`.

Topics in this section include:

- [Create a Report on HTMLDB\\_APPLICATION\\_FILES](#)
- [Add Link to Download Documents](#)

### Create a Report on HTMLDB\_APPLICATION\_FILES

To create a report on `HTMLDB_APPLICATION_FILES`:

1. Navigate to the Page Definition for page 1. Click **Edit Page 1** on the Developer toolbar.
2. Under Regions, click the **Create** icon.

3. For Region, select **Report** and click **Next**.
4. Specify the type of report. For Report Implementation, select **SQL Report** and click **Next**.
5. For Display Attributes:
  - a. In Title, enter `Uploaded Files`.
  - b. Accept the remaining defaults and click **Next**.
6. For Source, enter the following SQL query:
 

```
SELECT id,name FROM HTMLDB_APPLICATION_FILES
```
7. Click **Create Region**.
8. Run the page by clicking the **Run Page** icon.

As shown in [Figure 8–5](#), the report you just created shows all documents that have been uploaded.

**Figure 8–5** *Uploaded Files Report*

ID	NAME
11315724051556232	11315724051556232/HT_EMP
12133629820404211	12133629820404211/my_test1
12137232254480699	12137232254480699/my_plsql_block
12155516604646440	12155516604646440/my_sql_script
13052616061157279	13052616061157279/my_trigger_script

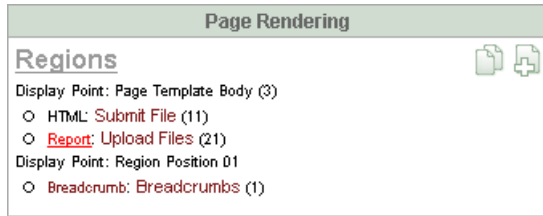
## Add Link to Download Documents

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

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

1. Navigate to the Page Definition for page 1. Click **Edit Page 1** on the Developer toolbar.
2. Under Regions, click **Report** adjacent to Uploaded Files as shown in [Figure 8–6](#) on page 8-6.

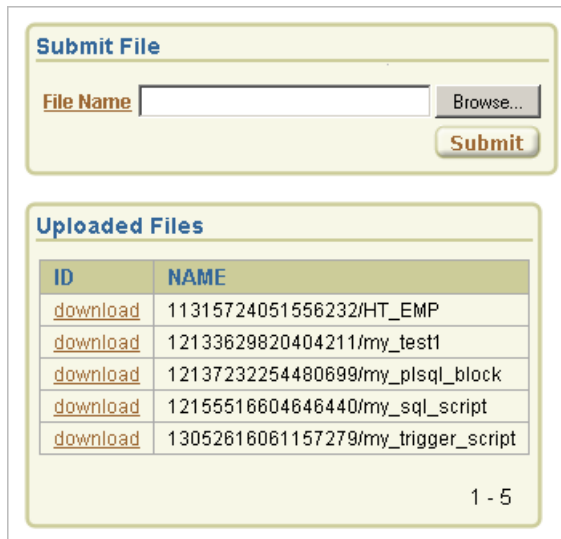
**Figure 8–6 Report Link**



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 next to the ID column.
4. Scroll down to Column Link.
5. Under Column Link:
  - a. In the Link Text field, enter:  
download
  - b. From Target, select **URL**.
  - c. In the URL field, enter the following:  
p?n=#ID#  
  
#ID# passes the value contained in the column where ID is the column alias.
6. Scroll back to the top of the page and click **Apply Changes**.  
The Page Definition appears.
7. Run the page by clicking the **Run Page** icon.  
When you run the page, it should look similar to [Figure 8–7](#).

**Figure 8–7 Uploaded Files Report with Download Links**



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



9. Return to the Database Home Page. Click the **Home** breadcrumb link at the top of the page as shown in [Figure 8-8](#).

**Figure 8-8 Breadcrumb Menu**

Home > Application Builder > Application 208 > 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](#)
- [Showing 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 Database Express Edition Application Express User's Guide*.

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

1. On the Database Home Page, click the **SQL** icon.
2. Click **SQL Commands**.
3. In the SQL Commands, enter:

```
CREATE TABLE file_subjects(name VARCHAR2(4000), subject VARCHAR2(4000) );
```

4. Click **Run**.

The message `Table created` appears.

5. Click the **Home** breadcrumb link.

The Database Home Page appears.

### Create an Item to Capture the Document Subject

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

1. Navigate to the Page Definition for page 1:
  - a. On the Database Home Page, click the **Application Builder** icon.

The Application Builder home page appears.

- b. Select **Download App**.  
The Application home page appears.
    - c. Select **Page 1**.  
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. For Sequence, accept the default.
    - c. From Region, select **Uploaded Files**.
    - d. Click **Next**.
  6. For Item Attributes:
    - a. In the Label field, enter Subject.
    - b. Accept the remaining defaults.
    - c. Click **Next**.
  7. Click **Create Item**.

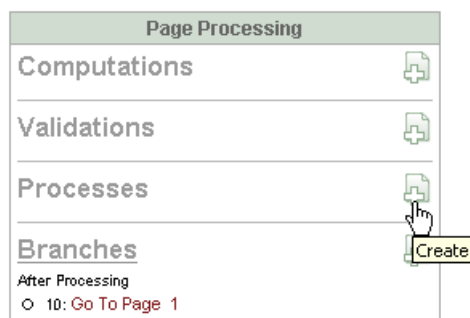
## Create a Process to Insert Information

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 as shown in [Figure 8–9](#).

**Figure 8–9 Create Icon**



2. For Process Type, select **PL/SQL** and click **Next**.
3. For Process Attributes:
  - a. For Name, enter Insert.
  - b. For 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 file_subjects(name, subject) VALUES(:P1_FILE_NAME, :P1_SUBJECT);
```

5. Click **Next**.
6. For Messages:
  - a. In Success Message, enter:  
Subject inserted
  - b. In Failure Message enter:  
Error inserting subject
  - c. Click **Next**.
7. For Process Conditions:
  - a. From When Button Pressed, select **SUBMIT**.
  - b. Accept the remaining defaults and click **Create Process**.

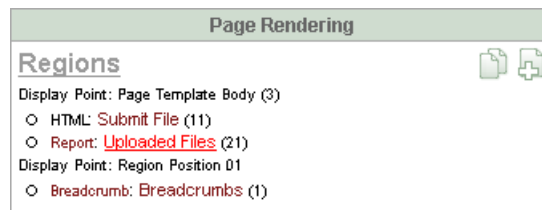
## Showing Additional Attributes in the Report Region

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** as shown in [Figure 8–10](#).

**Figure 8–10** *Uploaded Files Link*



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 HTMLDB_APPLICATION_FILES w,file_subjects s
WHERE w.name = s.name
```
4. Scroll up to the top of the page and click **Apply Changes**.
5. Run the page by clicking the **Run Page** icon.

As shown in [Figure 8–11](#) on page 8-10, the Uploaded Files report now contains a Subject column.

**Figure 8–11** Uploaded Files Report with Subject Column

**Submit File**

File Name

**Uploaded Files**

Subject

ID	NAME	Subject
<a href="#">download</a>	F340967991/Time_Estimator.xls	-

1 - 1

If your Uploaded Files report does not initially contain all three columns, try uploading a file and clicking the **Submit** button.

6. Return to the Page Definition by clicking **Edit Page 1** on the Developer toolbar.
7. Return to the Database Home Page. Click the **Home** breadcrumb link at the top of the page.

## Store 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 `file_subjects` table:

1. On to the Database Home Page, click the **SQL** icon.
2. Click **SQL Commands**.

SQL Commands appears.

3. Enter the following SQL statement:

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

4. Click **Run**.

The message Table Altered appears.

5. Return to the Database Home Page. Click the **Home** breadcrumb link at the top of the page.

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

1. On the Database Home Page, click the **Application Builder** icon.
2. Select **Download App**.
3. Select **Page 1**.

4. Under Processes, select **Insert**.
5. Scroll down to Source.
6. Under Source, replace the process with the following:
 

```
IF ( :P1_FILE_NAME is not null ) THEN
  INSERT INTO file_subjects(id,NAME, SUBJECT, BLOB_CONTENT, MIME_TYPE)
  SELECT ID, :P1_FILE_NAME, :P1_SUBJECT, blob_content, mime_type
  FROM HTMLDB_APPLICATION_FILES WHERE name = :P1_FILE_NAME;
  WHERE name = :P1_FILE_NAME;
DELETE from HTMLDB_APPLICATION_FILES WHERE name = :P1_FILE_NAME;
END IF;
```
7. Scroll up to the top of the page and click **Apply Changes**.
8. Return to the Database Home Page. Click the **Home** breadcrumb link at the top of the page.

## 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 HTMLDB\_PUBLIC\_USER.

To accomplish this you need to change:

- The SQL report region to no longer join to the HTMLDB\_APPLICATION\_FILES view
- The URL supplied for the ID column in the SQL report to execute the new procedure instead of executing the p procedure

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

1. On the Database Home Page, click the **SQL** icon.
2. Click **SQL Commands**.
3. 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 file_subjects
  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/octet
  -- application/octet 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
```

```

        http.p('Content-length: ' || v_length);
        -- the filename will be used by the browser if the users does a
save as
        http.p('Content-Disposition: attachment; filename="' || substr(v_
file_name,instr(v_file_name, '/')+1) || '.');
        -- close the headers
        owa_util.http_header_close;
        -- download the BLOB
        wpg_docload.download_file( Lob_loc );
end download_my_file;
/

```

**4. Click Run.**

The message Procedure Created appears. Run another SQL statement.

**5. Click SQL breadcrumb link and then SQL Commands.**

SQL Commands appears.

**6. Enter the following SQL statement:**

```

GRANT EXECUTE ON download_my_file TO PUBLIC
/

```

**7. Click Run.**

The message Statement processed appears.

**8. Return to the Database Home Page. Click the Home breadcrumb link at the top of the page.**

To change the SQL report region to no longer join with the HTMLDB\_APPLICATION\_FILES view:

**1. Navigate to the Page Definition of page 1:**

**a. On the Database Home Page, click the Application Builder icon.**

The Application Builder home page appears.

**b. Select Download App.**

The Application home page appears.

**c. Select Page 1.**

**2. Under Regions, select Uploaded Files.**

**3. Scroll down to Source.**

**4. Replace the Region Source with the following:**

```

SELECT s.id,s.name,s.subject FROM file_subjects s

```

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

The Page Definition appears.

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

**1. Under Regions, click Report adjacent to Uploaded Files.**

**2. Next to the ID column, click the Edit icon.**

**3. Scroll down to the Column Link.**

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





---

---

# How to Incorporate JavaScript into an Application

Adding JavaScript to a Web applications 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 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](#)
- [About Referencing Items Using JavaScript](#)
- [Calling JavaScript from a Button](#)
- [Creating a Client Side JavaScript Validation](#)
- [Enabling and Disabling Form Elements](#)
- [Changing the Value of Form Elements](#)

## Understanding How to Incorporate JavaScript Functions

There are two primary places to include JavaScript functions:

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

Topics in this section include:

- [Incorporating JavaScript in the HTML Header Attribute](#)
- [Including JavaScript in a .js File Referenced by the Page Template](#)

## Incorporating JavaScript in the HTML Header Attribute

One way to include JavaScript into your application is 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 the .js file.

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

To add JavaScript code in the HTML Header attribute:

1. On the Database Home Page, click the **Application Builder** icon.
2. Select an application.
3. Select a page.
4. Click **Edit Attributes**.
5. Scroll down to HTML Header.
6. Enter code into HTML Header and click **Apply Changes**.

For example, adding the following 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 since a .js file loads on the first page view of your application and is then cached by the browser.

The following 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 Database Express Edition Application Express User's Guide*

## About Referencing Items Using JavaScript

When you reference an item, the best approach is to reference by ID. If you view the HTML source of an Oracle Application Express page in a Web browser, you would notice that all items have an id attribute. This ID corresponds to the name of the item,

not the item label. For example, if you create an item with the name `P1_FIRST_NAME` and a label of `First Name`, the id would be `P1_FIRST_NAME`.

Knowing the item ID 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 would need to select **Action Redirect to URL without submitting page**. Then, you would specify a URL target such as the following:

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

## 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. For example, you can create a function in the HTML Header attribute of a page and then call that function from an item.

Creating this type of JavaScript validation involves the following steps:

- Create a new application on the `EMPLOYEES` table.
- Create an item on page 1 called `P2_LAST_NAME` that has a label of Last Name.
- Add a function to the HTML Header attribute on page 2
- Edit the `P2_LAST_NAME` item on page 2 to call the function

Topics in this section include:

- [About Accessing the HR Account](#)
- [Create an Application on the EMPLOYEES Table](#)
- [Add a Function to the HTML Header Attribute](#)
- [Edit an Item to Call the Function](#)

### About Accessing the HR Account

To complete this exercise, you need to log in to the sample database user account HR. Oracle Database Express Edition (Oracle Database XE) comes with a sample database user account called HR. This user owns a number of database tables in a sample schema that can be used to create applications for a fictional Human Resources department.

For security reasons, the HR user account is locked after installation. To learn how to unlock it, see "[About the HR Account](#)" on page 1-1.

### Create an Application on the EMPLOYEES Table

To create a new application on the `EMPLOYEES` table:

1. On the Database Home Page, click the **Application Builder** icon.
2. Click **Create**.
3. For Method, select **Create Application**.
4. For Name:
  - a. In Name, enter a name that describes the application.
  - b. For Application, accept the default.
  - c. For Create Application, select **From scratch**.
  - d. For Schema, accept the default.
  - e. Click **Next**.
5. Add a blank page, containing a report:
  - a. Under Select Page Type, select **Report and Form**.
  - b. From Table Name, select **EMPLOYEES**.
  - c. Click **Add Page**.

The new pages display 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 **EMPLOYEES** for page 2 as shown in [Figure 9-1](#).

**Figure 9-1 Newly Created Pages**

Create Application					Cancel	< Previous	Next >
Page	Page Name	Page Type	Source Type	Source			
1	EMPLOYEES	Report	Table	EMPLOYEES			✘
2	EMPLOYEES	Form	Table	EMPLOYEES			✘

The Page Definition appears.

- b. In Page Name, enter `Update Form` and click **Apply Changes**.
  - c. Click **Next**.
7. For Tabs, select **One Level of Tabs** and 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 - EMPLOYEES
- 2 - Update Form
- 101 - Login

To view the application:

1. Click the **Run Application** icon as shown in [Figure 9-2](#).

**Figure 9-2 Run Page Icon**



2. When prompted for a user name and password, enter `hr` for both and click **Login**.  
A standard report appears. To view the update form click either the **Create** button or Edit icon.
3. Click **Edit Application** on the Developer toolbar.

## 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, select **2 - Update Form**.  
The Page Definition appears.
2. Click **Edit Attributes**.  
The Edit Page attributes page appears.
3. Scroll down to HTML Header. Note that HTML Header already contains a script.
4. In HTML Header, scroll down and enter the following script after the `</script>` tag:

```
<script type="text/javascript">
  function notNull(object){
    if(object.value=="")
      alert('This field must contain a value.');
```
5. Scroll back to the top of the page and click **Apply Changes**.

## Edit an Item to Call the Function

Next, you need to edit the `P2_LAST_NAME` item to call the function. In the running form, the `P2_LAST_NAME` item displays as the Last Name field.

To edit the `P2_LAST_NAME` item to call the function:

1. Navigate to the Page Definition for page 2 - Update Form.
2. Under Items, select **P2\_LAST\_NAME**.
3. Scroll down to Element.
4. In HTML Form Element Attributes, enter the following:

```
onblur="notNull(this);"
```
5. Scroll back to the top of the page and click **Apply Changes**.  
The Page Definition appears.
6. Navigate to page 1. Enter 1 in the Page field and click **Go**.
7. Click the **Run Page** icon in the upper right corner.
8. When the application appears, click **Create**.  
The Update Form appears.
9. Position your cursor in the Last Name field and click **Create**. The following message appears:  
This field must contain a value.

## 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 the HTML Header Attribute](#)
- [Edit an Item to Call the Function](#)
- [Change P2\\_DEPARTMENT\\_ID to a Select List](#)
- [Create a Call to the disFormItems Function](#)

### Add a Function the HTML Header Attribute

To add a function to the HTML Header attribute on page 2:

1. Navigate to the Page Definition for page 2.
2. On the Page Definition, click **Edit Attributes**.

The Page Attributes appear.

3. Scroll down to HTML Header.
4. In HTML Header, scroll down and enter the following script after the last `</script>` tag:

```
<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>
```

5. 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. Navigate to the Page Definition for page 2.
2. Under Items, select **P2\_DEPARTMENT\_ID**.
3. Scroll down to Element.
4. In HTML Form Element Attributes, enter the following:  

```
onchange="disFormItems()"
```
5. Click **Apply Changes**.

## Change P2\_DEPARTMENT\_ID to a Select List

To change the `P2_DEPARTMENT_ID` to display as a select list:

1. Navigate to the Page Definition for page 2.
2. Under Items, select `P2_DEPARTMENT_ID`.
3. Under Display As, select **Select List**.
4. Scroll down to List of Values.
5. Under List of Values:
  - a. From Display Null, select **No**.
  - b. In List of Values definition, enter:  

```
SELECT department_name, department_id FROM departments
```
6. Click **Apply Changes**.

## 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. Navigate to the Page Definition for page 2.
2. Click **Edit Attributes**.
3. Scroll down to On Load.
4. In the Page HTML Body Attribute, enter the following:  

```
onload="disFormItems()"
```
5. Click **Apply Changes**.
6. Run the page by clicking the **Run Page** icon.

**Figure 9-3** on page 9-9 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 9–3 Revised Update Form

## 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 call the function from the first three items:

1. Navigate to the appropriate Page Definition.
2. On the Page Definition, click **Edit Attributes**.

The Page Attributes appear.

3. In HTML Header, 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. Navigate to the appropriate Page Definition.
2. For each item:
  - a. Select the item name.
  - b. Scroll down to Element.
  - c. In HTML Form Element Attributes, enter:

```
onchange="sumItems();"

```

- d. Click **Apply Changes**.

---

---

# How to Build and Deploy an Issue Tracking Application

Storing information in an Oracle database organizes it into tables that group similar information together and removes redundancies. Using the Oracle Application Express development environment, you can quickly build an application that enables a user to view and update information stored in an Oracle Database.

This tutorial describes how to create and deploy an application that tracks the assignment, status, and progress of issues related to a project.

---

---

**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](#)
- [Designing the Database Objects](#)
- [Implementing Database Objects](#)
- [Loading Demonstration Data](#)
- [Building a Basic User Interface](#)
- [Adding Advanced Features](#)
- [Deploying Your Application](#)

## 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 with spreadsheets.

By creating a hosted application in Oracle Application Express, project leads can easily record and track issues in one central location. This approach allows each project lead

to 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:

- [Gathering the Necessary Data](#)
- [Defining Security Requirements](#)
- [Selecting Data Management Functions](#)
- [Selecting Data Presentation Functions](#)
- [Defining Special Function Requirements](#)

### Gathering 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 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

### Defining Security Requirements

Because the project leads were 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

### Selecting 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

### Selecting 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

### Defining 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 in 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 10-2, 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 10–1](#) describes the columns to be included in the `Projects` table.

**Table 10–1** *Project Table Details*

Column Name	Type	Size	Not Null?	Constraints	Description
<code>project_id</code>	integer	n/a	Yes	Primary key	A unique numeric identification number for each project.  Populated by a sequence using a trigger.
<code>project_name</code>	<code>varchar2</code>	100	Yes	Unique key	A unique alphanumeric name for the project.
<code>start_date</code>	date	n/a	Yes	None	The project start date.
<code>target_end_date</code>	date	n/a	Yes	None	The targeted project end date.
<code>actual_end_date</code>	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 10–2](#) on page 10-5 describes the columns that will be included in the `People` table.

**Table 10–2** *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.
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 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 last modified the issue and on what date that modification was made.

Table 10–3 describes the columns to be included in the Issues table.

**Table 10–3** *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.

**Table 10-3 (Cont.) Issue Table Details**

Column Name	Type	Size	Not Null?	Constraints	Description
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.
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 needed to be dynamic, you would need 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:

- [Build the Database Objects](#)
- [View the Created Database Objects](#)

### Build the 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 Object Browser and click **Create**. See "Managing Database Objects with Object Browser" in *Oracle Database Express Edition Application Express 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** icon on Database Home Page and then click **SQL Commands**. See "Using SQL Commands" in *Oracle Database Express Edition Application Express 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** on the Database Home Page, click **SQL Scripts**, and then click **Upload**. See "Uploading a SQL Script" in *Oracle Database Express Edition Application Express 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** icon on the Database Home Page, select **SQL Scripts**, and then click **Create**. See "Creating a SQL Script in the Script Editor" in *Oracle Database Express Edition Application Express User's Guide*.

To build database objects by creating a script:

1. Log in to Oracle Database Express Edition.
2. On the Database Home Page, click the **SQL** icon.
3. Click **SQL Scripts**.
4. Click **Create**.
5. In the Script Editor:
  - a. For Script Name, enter `DDL for Issue Management Application`.
  - b. Copy the DDL (data definition language) in "[Create Application Database Objects DDL](#)" on page A-1 and paste it into the script.
  - c. Click **Save**.

To run the DDL for Issue Management Application script:

1. On the SQL Scripts page, select the **DDL for Issue Management Application** icon.  
The Script Editor appears.
2. Click **Run**.  
A summary page appears.
3. 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 Database Home Page by clicking the **Home** breadcrumb link.
2. On the Database Home Page, click the **Object Browser** icon.
3. From the Object list on 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 Database Express Edition Application Express 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, you use SQL Scripts to load demonstration data.

Look at the DDL you copied from "[Create 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. Because 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. Return to the Database Home Page by clicking the **Home** breadcrumb link.
2. On the Database Home Page, click the **SQL** icon.
3. Click the **SQL Scripts** icon.
4. Click **Create**.
5. In the Script Editor:
  - a. In Script Name, enter `Load Project Data`.
  - b. In 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**.
6. On the SQL Scripts page, select the **Load Project Data** icon.
7. Click **Run**.

A summary page appears.

8. 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. In 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, select the **Update Project Dates** icon.
5. Click **Run**.
 

A summary page appears.
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.

To load data into the People table:

1. Click the **SQL Scripts** breadcrumb link.
2. Click **Create**.
3. In the Script Editor:
  - a. In Script Name, enter Load People Data.
  - b. In 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.williams@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, select the **Load People Data** icon.**

**5. Click Run.**

A summary page appears.

**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. In Script Name, enter `Load Issue Data`.
  - b. In Script, copy and paste the script in "[Create Issues Script](#)" on page A-1.
  - c. Click **Save**.
4. On the SQL Scripts page, select the **Load Issue Data** icon.
5. Click **Run**.

A summary page appears.

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 10-2.

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 selected tables within a schema. Alternatively, you can create an application first and then add pages to it. Since 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 Database Home Page, click the **Application Builder** icon.

3. Click **Create**.
4. For Method, select **Create Application**.
5. For Name:
  - a. In Name, enter a `Issue Tracker`.
  - b. For Create Application, select **From scratch**.
  - c. Click **Next**.
6. Add a blank page:
  - a. Under Select Page Type, select **Blank**.
  - b. Click **Add Page**.
  - c. Click **Next**.
7. For Tabs, select **No Tabs** and click **Next**.
8. For Shared Components, accept the default 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 database account username and password and click **Login**.

This authentication is part of the default security of any newly created application. As shown in [Figure 10-1](#), the home page appears.

**Figure 10-1** Issue Tracking Application Home Page



Although the page has no content, notice that the Create Application Wizard 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 at the bottom of 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. From left to right, the Developer toolbar contains the following links:
    - **Edit Application** - Edit the application by linking to the Application Builder home page.
    - **Edit Page 1** - Edit the current running page. This link takes you to Page Definition for the current page.
    - **Create** - Add a new component to the current page.
    - **Session** - Open a new page containing session details for the current page.
    - **Debug** - Display the current page in debug mode.
    - **Show Edit Links** - Displays edit links next to each object on the page that can be edited. Each edit link resembles two colons (: :) and appears to the right of navigation bar items, tabs, region titles, buttons, and items. Clicking an edit link displays another window where you can edit the object.
3. Click **Edit Application** on the Developer toolbar to return to Application Builder 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:

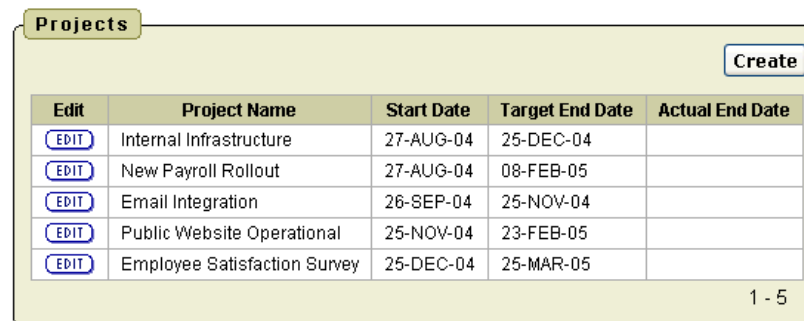
1. On the Application Builder home page, click **Create Page**.
2. Select **Form** and click **Next**.
3. Select **Form on a Table with Report** and click **Next**.
4. For Table/View Owner, select the appropriate schema and click **Next**.
5. For Table/View Name, select HT\_PROJECTS and click **Next**.
6. For Define Reports Page:
  - a. For Page, enter 2.
  - b. For Page Name and Region Title, enter `Projects`.



- c. Click **Next**.
7. For Tab Options, accept the default selection **Do not use tabs** and click **Next**.
8. For Select Column(s), select every column except `PROJECT_ID` and 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 third option (the word Edit in blue with a white background) and click **Next**.
10. For Define Form Page:
  - a. For Page, enter 3.
  - b. For 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 10–2](#), the newly created report displays the demo data.

**Figure 10–2** *Projects Page*



Edit	Project Name	Start Date	Target End Date	Actual End Date
<a href="#">EDIT</a>	Internal Infrastructure	27-AUG-04	25-DEC-04	
<a href="#">EDIT</a>	New Payroll Rollout	27-AUG-04	08-FEB-05	
<a href="#">EDIT</a>	Email Integration	26-SEP-04	25-NOV-04	
<a href="#">EDIT</a>	Public Website Operational	25-NOV-04	23-FEB-05	
<a href="#">EDIT</a>	Employee Satisfaction Survey	25-DEC-04	25-MAR-05	

Click the **Edit** icon to view an existing row or click the **Create** button to create a new record. If you click the **Edit** icon to the left of Employee Satisfaction Survey, a form resembling [Figure 10–3](#) on page 10-16 appears.

Figure 10–3 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. Navigate to the Page Definition for page 2, Projects:
  - a. Click **Edit Application** on the Developer toolbar.
  - b. On the Application home page, select **2 - Projects**.
2. Under Regions, select **Report** adjacent to Projects.
3. Edit the format for START\_DATE:
  - a. Click the **Edit** icon the left of START\_DATE.  
The Column Attributes page appears.
  - b. 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 navigate to the next Report Item.  
The Column Attributes page appears.
  - b. 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 navigate to the next Report Item.  
The Column Attributes page appears.
  - b. 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.
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 overridden by any user selections.

11. Scroll down to Sorting. For Ascending and Descending Image, select the light gray arrow.
12. Under Messages, enter the following in When No Data Found Message:  
No Projects found.
13. Scroll up to the top of the page and click **Apply Changes**.

To view your changes, click the **Run Page** icon in the upper right of the page.

As shown in [Figure 10-4](#), 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 10-4 Projects Page with Sort Control**

	Project Name ▲	Start Date	Target End Date	Actual End Date
<a href="#">EDIT</a>	Email Integration	26-SEP-2004	25-NOV-2004	
<a href="#">EDIT</a>	Employee Satisfaction Survey	25-DEC-2004	25-MAR-2005	
<a href="#">EDIT</a>	Internal Infrastructure	27-AUG-2004	25-DEC-2004	
<a href="#">EDIT</a>	New Payroll Rollout	27-AUG-2004	08-FEB-2005	
<a href="#">EDIT</a>	Public Website Operational	25-NOV-2004	23-FEB-2005	

### 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.

To make the Project Name field larger and the date fields smaller:

1. Navigate to the Page Definition for Page 3, Create/Edit Project.
  - a. From the Developer toolbar, click **Edit Application**.
  - b. Select **3 - Create/Edit Project**.
2. Under the Page Rendering section, select the **Items** heading.
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. Click the **Edit Page** icon in the upper right corner of the page to return to the Page Definition.

To change the date picker type and add a format mask for dates:

1. Edit the item P3\_START\_DATE.
  - a. Under Items, select **P3\_START\_DATE**.

- b. Under Name, for Display As select **Date Picker (DD-MON-YYYY)**.
- c. Click **Apply Changes**.

Edit the item P3\_TARGET\_END\_DATE.

- a. Under Items, select **P3\_TARGET\_END\_DATE**.
- b. Under Name, for Display As select **Date Picker (DD-MON-YYYY)**.
- c. Click **Apply Changes**.

Edit the item P3\_ACTUAL\_END\_DATE.

- a. Under Items, select **P3\_ACTUAL\_END\_DATE**.
- b. Under Name, for Display As select **Date Picker (DD-MON-YYYY)**.
- c. Click **Apply Changes**.

To add validations to check if the target and actual end dates are after the start date:

1. Under the Validations section, 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 and Error Message:
  - a. For Validation, enter:

```
to_date(:P3_ACTUAL_END_DATE, 'DD-MON-YYYY') >= to_date(:P3_START_DATE, 'DD-MON-YYYY')
```
  - b. For Error Message, enter:

```
Actual End Date must be same or after Start Date.
```
  - c. Click **Next**.
8. For Conditions:
  - a. For **Condition Type**, select **Value of Item in Expression 1 Is NOT NULL**, or click the shortcut link **[item not null]**.
  - b. For Expression 1, enter:

```
P3_ACTUAL_END_DATE.
```

This selection ensures that this validation only executes 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 10-5](#) on page 10-19.)

Figure 10–5 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. Return to the Application home page. If you are viewing the Create/Edit Project form, click **Edit 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 Attributes:
  - a. For Page, enter 4.
  - b. For 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 third option (the word Edit in blue with a white background) and click **Next**.
11. For Define Form Page:
  - a. For Page, enter 5.
  - b. For 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 click **Finish**.

To preview your page, click **Run Page**. As shown in [Figure 10–6](#), notice the newly created report displays the demo data.

**Figure 10–6 People Page**

Edit	Person Name	Person Email	Person Role	Assigned Project
<a href="#">EDIT</a>	Joe Cerno	joe.cerno@mrvl-bademail.com	CEO	
<a href="#">EDIT</a>	Kim Roberts	kim.roberts@mrvl-bademail.com	Manager	
<a href="#">EDIT</a>	Tom Suess	tom.suess@mrvl-bademail.com	Manager	
<a href="#">EDIT</a>	Al Bines	al.bines@mrvl-bademail.com	Lead	1
<a href="#">EDIT</a>	Carla Downing	carla.downing@mrvl-bademail.com	Lead	2
<a href="#">EDIT</a>	Evan Fanner	evan.fanner@mrvl-bademail.com	Lead	3
<a href="#">EDIT</a>	George Hurst	george.hurst@mrvl-bademail.com	Lead	4
<a href="#">EDIT</a>	Irene Jones	irene.jones@mrvl-bademail.com	Lead	5
<a href="#">EDIT</a>	Karen London	karen.london@mrvl-bademail.com	Member	1
<a href="#">EDIT</a>	Mark Nile	mark.nile@mrvl-bademail.com	Member	1
<a href="#">EDIT</a>	Jane Kerry	jane.kerry@mrvl-bademail.com	Member	5
<a href="#">EDIT</a>	Olive Pope	olive.pope@mrvl-bademail.com	Member	2
<a href="#">EDIT</a>	Russ Sanders	russ.sanders@mrvl-bademail.com	Member	3
<a href="#">EDIT</a>	Tucker Uberton	tucker.uberton@mrvl-bademail.com	Member	3
<a href="#">EDIT</a>	Vicky Williams	vicky.willaims@mrvl-bademail.com	Member	4

row(s) 1 - 15 of 17 [Next](#)

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.

To change the query to include a join to the Projects table:

1. Navigate to the Page Definition for page 4 - People:
  - a. If you are viewing a running form, click **Edit Application** on the Developer toolbar.
  - b. When the Application home page appears select **4 - People**.
2. Under Regions, select **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.

5. Select the Report Attributes tab at the top of the page.
  - a. For PERSON\_ID, remove the Heading **Edit**.
  - b. For PERSON\_NAME, change Heading to Name.
  - c. For PERSON\_EMAIL, change Heading to Email.
  - d. For PERSON\_ROLE, change Heading to Role.
  - e. For PROJECT\_NAME, change Heading to Assigned Project and select **left** for Heading Align.
6. Enable column heading sorting by selecting **Sort** for all columns except PERSON\_ID.
7. 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.
8. Scroll down to Sorting. For Ascending and Descending Image, select the light gray arrow.
9. Under Messages, enter the following in When No Data Found Message:  
No people found.

#### 10. Click **Apply Changes**.

To view your changes, click the **Run Page** icon in the upper right of the page. As shown in [Figure 10-7](#), note the addition of a sort control on the Name column.

**Figure 10-7 Revised People Page**

	Name ▲	Email	Role	Assigned Project
<a href="#">EDIT</a>	Al Bines	al.bines@mrvl-bademail.com	Lead	Internal Infrastructure
<a href="#">EDIT</a>	Carla Downing	carla.downing@mrvl-bademail.com	Lead	New Payroll Rollout
<a href="#">EDIT</a>	Evan Fanner	evan.fanner@mrvl-bademail.com	Lead	Email Integration
<a href="#">EDIT</a>	George Hurst	george.hurst@mrvl-bademail.com	Lead	Public Website Operational
<a href="#">EDIT</a>	Irene Jones	irene.jones@mrvl-bademail.com	Lead	Employee Satisfaction Survey
<a href="#">EDIT</a>	Jane Kerry	jane.kerry@mrvl-bademail.com	Member	Employee Satisfaction Survey
<a href="#">EDIT</a>	Joe Cerno	joe.cerno@mrvl-bademail.com	CEO	
<a href="#">EDIT</a>	Karen London	karen.london@mrvl-bademail.com	Member	Internal Infrastructure
<a href="#">EDIT</a>	Kim Roberts	kim.roberts@mrvl-bademail.com	Manager	
<a href="#">EDIT</a>	Mark Nile	mark.nile@mrvl-bademail.com	Member	Internal Infrastructure
<a href="#">EDIT</a>	Olive Pope	olive.pope@mrvl-bademail.com	Member	New Payroll Rollout
<a href="#">EDIT</a>	Russ Sanders	russ.sanders@mrvl-bademail.com	Member	Email Integration
<a href="#">EDIT</a>	Scott Tiger	scott.tiger@mrvl-bademail.com	Member	Public Website Operational
<a href="#">EDIT</a>	Tom Suess	tom.suess@mrvl-bademail.com	Manager	
<a href="#">EDIT</a>	Tucker Uberton	tucker.uberton@mrvl-bademail.com	Member	Email Integration

row(s) 1 - 15 of 17 [Next](#)

## 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 Lists of Values** To add a list of values for Projects:

1. Navigate to the Page Definition for page 5, Create/Edit Person:
  - a. If you are viewing a running form, click **Edit Application** on the Developer toolbar.
  - b. When the Application home page appears select **5 - Create/Edit Person**.
2. Under Shared Components, locate the Lists of Values section and click the **Create** icon.
3. For Source, accept the default **From Scratch** and click **Next**.
4. For Name and Type:
  - a. For Name, enter PROJECTS.
  - b. For 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**.

To add a list of values for Roles:

1. Under Shared Components, locate the Lists of Values section and click the **Create** icon.
2. For Source, accept the default **From Scratch** and click **Next**.
3. For Name and Type:
  - a. For Name, enter ROLES.
  - b. For Type:, select **Static**
  - c. Click **Next**.
4. Enter the display value and return value pairs shown in [Table 10–4](#):

**Table 10–4** *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. Return to the Page Definition for Page 5. Click the **Edit Page** icon in the upper right corner.



**Edit Display Attributes** To edit display attributes for P5\_PERSON\_ROLE:

1. Under Items, select **P5\_PERSON\_ROLE**.
2. Under Name, select **Radiogroup** from the Display As list.
3. Scroll down to Label.
4. Change Label to Role.
5. Under Element, enter the following in Form Element Option Attribute:

```
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, select **P5\_ASSIGNED\_PROJECT**.
2. Under Name, select **Select List** from the Display As 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 the Page Rendering section, select the **Items** heading.
2. For P5\_PERSON\_NAME:
  - a. For Prompt, enter Name.
  - b. For Width, enter 60.
3. For P5\_PERSON\_EMAIL:
  - a. For 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 the Validations section, click the **Create** icon.
2. On Level, accept the default **Item level validation** and click **Next**.
3. On Item, select **Create/Edit Person Information: 50. P5\_ASSIGNED\_PROJECT (Assigned Project)** and click **Next**.
4. On Validation Method:
  - a. Select **PL/SQL** and click **Next**.
  - b. Accept the default, **PL/SQL Expression**, and click **Next**.
5. For Validation Name, enter `PROJECT_MAND_FOR_LEADER_AND_MEMBER` and click **Next**.
6. For Validation and Error Message:

- a. For 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 so to keep it in the validation, you need to break the string apart so that it is not recognized and replaced.

- b. For 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**.

To view your changes, click the **Run Page** icon in the upper right of the page. (See [Figure 10–8](#).)

**Figure 10–8 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. Return to the Application home page. If you are viewing a running form, click Edit 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. For Page, enter 6.
  - b. For 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), select the following and click **Next**:
  - ISSUE\_SUMMARY
  - IDENTIFIED\_BY
  - RELATED\_PROJECT
  - ASSIGNED\_TO
  - STATUS
  - PRIORITY
  - TARGET\_RESOLUTION\_DATE
  - ACTUAL\_RESOLUTION\_DATE
10. For **Edit Link Image**, select the third option (the word Edit in blue with a white background) and click **Next**.
11. On Define Form Page:
  - a. For Page, 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. Define the source for the primary key columns. Accept the default, **Existing Trigger**, and click **Next**.
- 15. For Select Column(s), press **SHIFT** and 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**.

### Refine the Create/Edit Issues Page

When you refine the Create/Edit Page you:

- 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. Navigate to the Page Definition for page 7.
2. Under the Lists of Values section, click the **Create** icon.
3. For Create List of Values, accept the default **From Scratch** and click **Next**.
4. On Create List of Values (LOV):
  - a. For Name, enter `STATUS`.
  - b. For Type, select **Static**.
  - c. Click **Next**.
5. Enter the Display Value and Return Value pairs shown in [Table 10-5](#):

**Table 10-5** *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. Return to the Page Definition for Page 7. Click the **Edit Page** icon in the upper right corner.
2. Under the Lists of Values section and click the **Create** icon.
3. For Create List of Values, accept the default **From Scratch** and click **Next**.

4. On Create List of Values (LOV):
  - a. For Name, enter PRIORITIES.
  - b. For Type, select **Static**.
  - c. Click **Next**.
5. Enter the Display Value and Return Value pairs shown in [Table 10–6](#).

**Table 10–6 Display Value and Return Value Pairs**

Display Value	Return Value
High	High
Medium	Medium
Low	Low

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

To add a list of values for People:

1. Return to the Page Definition for Page 7. Click the **Edit Page** icon in the upper right corner.
2. Under the Lists of Values section and click the **Create** icon.
3. For Create List of Values, accept the default **From Scratch** and click **Next**.
4. On Create List of Values (LOV):
  - a. For Name, enter PEOPLE.
  - b. For Type, select **Dynamic**.
  - c. Click **Next**.
5. In Query, replace the existing statements with the following:

```
SELECT person_name d, person_id v
FROM ht_people
ORDER BY 1
```

6. Click **Created List of Values**.

**Edit Specific Items** Next, you edit individual items.

To edit P7\_IDENTIFIED\_BY:

1. Return to the Page Definition for Page 7 by clicking the **Edit Page** icon in the upper right corner.
2. Under Items, select **P7\_IDENTIFIED\_BY**.
3. Under Name, select **Select List** from the Display As list.
4. Scroll down to List of Values:
  - a. For Named LOV, select **PEOPLE**.
  - b. For 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. For Null display value, enter:

- Select Person -

5. Click the Next button (>) at the top of the page to navigate to the next item.

To edit P7\_IDENTIFIED\_DATE:

1. Navigate to P7\_IDENTIFIED\_DATE.
2. Under Name, select **Date Picker (DD-MON-YYYY)** from the Display As list.
3. Scroll down to Default:
  - a. For Default value, enter:
 

```
to_char(sysdate, 'DD-MON-YYYY')
```
  - b. For Default Value Type, select **PL/SQL Expression**.
4. Click the Next button (>) at the top of the page to navigate to the next item.

To edit P7\_RELATED\_PROJECT:

1. Navigate to P7\_RELATED\_PROJECT.
2. Under Name, select **Select List** from the Display As list.
3. Scroll down to List of Values:
  - a. For Named LOV, select **PROJECTS**.
  - b. For Display Null, select **Yes**.
  - c. For Null display value, enter:
 

```
- Select Project -
```

4. Click the Next button (>) at the top of the page to navigate to P7\_STATUS.

To edit P7\_STATUS:

1. Navigate to P7\_STATUS.
2. Under Name, select **Radiogroup** from the Display As list.
3. In Label, enter:

```
Status:
```

4. Scroll down to Element. Enter the following in the Form Element Option Attributes:

```
class="instructiontext"
```

5. Scroll down to Default. In Default Value, enter `Open`.
6. Scroll down to List of Values:
  - a. For Named LOV, select **STATUS**.
  - b. For Columns, enter 3.

This selection enables the three valid values to display side by side.

7. Click the Next button (>) at the top of the page to navigate to P7\_PRIORITY.

To edit P7\_PRIORITY:

1. Navigate to P7\_PRIORITY.
2. Under Name, select **Radiogroup** from the Display As list.
3. In Label, enter:

Priority:

4. Scroll down to Element. Enter the following in the Form Element Option Attributes:

```
class="instructiontext"
```

5. Scroll down to Default. In Default value, enter `Open`.
6. Scroll down to List of Values:
  - a. For Named LOV, select **PRIORITIES**.
  - b. For Display Null, select **Yes**.
  - c. For Columns, enter 4.

This selection reflects that fact there are three valid values plus the null value.

- d. For Null display value, enter `-None-`.
7. Click the Next button (>) at the top of the page to navigate to the next item.

To edit P7\_TARGET\_RESOLUTION\_DATE:

1. Navigate to P7\_TARGET\_RESOLUTION\_DATE.
2. Under Name, select **Date Picker (DD-MON-YYYY)** from the Display As list.
3. Click the Next button (>) at the top of the page to navigate to P7\_ACTUAL\_RESOLUTION\_DATE.

To edit P7\_ACTUAL\_RESOLUTION\_DATE:

1. Navigate to P7\_ACTUAL\_RESOLUTION\_DATE.
2. Under Name, select **Date Picker (DD-MON-YYYY)** from the Display As list.
3. 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. Next, you create four new regions named Buttons, Identification, Progress, Resolution, and Auditing.

To create new regions to group items:

1. Under Regions, click the **Create** icon.
2. Select **Multiple HTML**.
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, select **Create/Edit Issue**.
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 the Page Rendering section, select the **Items** heading.
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. For P7\_ISSUE\_SUMMARY, enter 60 for Width.
6. For P7\_IDENTIFIED\_DATE, enter 12 for Width.
7. For P7\_TARGET\_RESOLUTION\_DATE, enter 12 for Width.
8. For P7\_ACTUAL\_RESOLUTION\_DATE, enter 12 for Width.
9. Click **Apply Changes**.
10. Click the **Edit Page** icon in the upper right to return the Page Definition of Page 7.

To move buttons to the Button region:

1. Return to the Page Definition for Page 7.
2. Under the Page Rendering section, select the **Buttons** heading.
3. Under Region for all buttons, select **Buttons**.
4. Click **Apply Changes**.
5. 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. On the Page Definition for Page 7, under Regions, select **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 `P7_ISSUE_ID`.
5. Click **Apply Changes**.

Next, change the audit columns to display only.

To edit `P7_CREATED_DATE`:

1. Under Items, select **P7\_CREATED\_DATE**.
2. Under Name, select **Display as Text (saves state)** from the Display As list.
3. Scroll down to 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. Scroll down to Source and enter the following in Format Mask:

`DD-MON-YYYY`

5. Click the Next button (>) at the top of the page to navigate to the next item.

To edit `P7_CREATED_BY`:

1. Under Name, select **Display as Text (saves state)** from the Display As list.
2. Scroll down to 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 navigate to the next item.

To edit `P7_LAST_MODIFIED_DATE`:

1. Under Name, select **Display as Text (saves state)** from the Display As list.
2. Scroll down to 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. Scroll down to Source and enter the following in Format Mask:

DD-MON-YYYY

4. Click the Next button (>) at the top of the page to navigate to the next item.

To edit P7\_LAST\_MODIFIED\_BY:

1. Under Name, select **Display as Text (saves state)** from the Display As list.

2. Scroll down to 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. Under Validations, select **P7\_CREATED\_DATE not null**.

2. Click **Delete**.

A dialog box displays prompting you to confirm the deletion.

3. Click **OK** to confirm your selection.

4. Under Validations, select **P7\_CREATED\_BY not null**.

5. Click **Delete**.

A dialog box displays prompting you to confirm the deletion.

6. 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. For Item Name, enter:

P7\_PREV\_PAGE

- b. For 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, select **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.
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 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. Under Name, select **CREATE**.
3. For Target Page, accept the default 7 and click **Next**.
4. For Button Name, enter `CREATE_AGAIN`.
5. For Label, enter `Create` and `Create Another`.
6. 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 Branches, click the **Create** icon.
2. For Point and Type, accept the defaults and click **Next**.
3. For Target:
  - a. For Page, enter 7.
  - b. For Clear Cache, enter 7.

- c. For Set these items, enter the following:  
P7\_PREV\_PAGE
  - d. For With these values, enter (be sure to include the period):  
&P7\_PREV\_PAGE.
  - e. Click **Next**.
4. For Branch Conditions:
    - a. For Sequence, enter 0.
    - b. For 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.
  7. Under Action, select **include process success message**.
  8. Click **Apply Changes**.  
To see the changes, click the **Run Page** icon. (See [Figure 10-9](#).)

**Figure 10-9 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 of the Create button:

1. Navigate to the Page Definition for page 6, Issues.
2. Under Regions, click the **Create** icon.
3. Select **HTML** and click **Next**.
4. Specify the type of HTML region container you want to create. Select **HTML** and click **Next**.
5. For Display Attributes:
  - a. For Title, enter `Buttons`.
  - b. For Region Template, select **Button Region without Title**.
  - c. For 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, select the **CREATE** button.
2. In Text Label, enter:  
`Add a New Issue`
3. From Display in Region, select **Buttons**.
4. Scroll down to Optional URL Redirect:
  - a. For Set These Items, enter:  
`P7_PREV_PAGE`
  - b. For With These Values, enter `6`.
5. Click **Apply Changes**.

**Alter the Query and Display** Next, alter 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 the Regions section, select **Report** adjacent to Issues.  
The Report Attribute page appears.
2. Click the **Edit Icon** to the left of `ISSUE_ID`.

3. Scroll down to Column Link.
  - a. For Item 2, for Name enter:

`P7_PREV_PAGE`

- b. For Item 2, for 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 the Display As list, select **Display as Text (based on LOV, does not save state)**.
3. Scroll down to List of Values. For 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 the Display As list, select **Display as Text (based on LOV, does not save state)**.
6. Scroll down to List of Values:
  - a. For Named LOV, select **PROJECTS**.
  - b. For Display Null, select **Yes**.
  - c. In 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 the Display As list, select **Display as Text (based on LOV, does not save state)**.
9. Scroll down to List of Values:
  - a. For Named LOV, select **PEOPLE**.
  - b. For Display Null, select **Yes**.
  - c. In 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, select **Custom**.
2. For ISSUE\_ID, delete the current entry under Heading.
3. For ISSUE\_SUMMARY, change 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 Align., select **center**.

- c. For Heading Align, select **center**.
5. For all columns except `ISSUE_ID`, check **Sort**.
6. For `ISSUE_SUMMARY`, select **1** for Sort Sequence.
7. Scroll down to Layout and Pagination:
  - a. For Show Null Values as, enter a hyphen (-).
  - b. For Number of Rows, enter 5.
8. Under Sorting, select the light gray arrow for Ascending and Descending Image.
9. Under Messages, enter the following in When No Data Found Message:
 

No issues found.
10. Click **Apply Changes**.

**Add Support of 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 which will display above the report.

To create a new region:

1. Under Regions, click the **Create** icon.
2. Select **HTML** and click **Next**.
3. Select the type of HTML region container you want to create. Select **HTML** and click **Next**.
4. For Display Attributes:
  - a. For Title, enter `Issue Report Parameters`.
  - b. For Region Template, select accept the default of **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. On Display Position and Name:
  - a. For Item Name, enter `P6_IDENTIFIED_BY`.
  - b. For Region, select **Issue Report Parameters**.
  - c. Click **Next**.
5. On Identify List of Values:
  - a. For Named LOV, select **PEOPLE**.
  - b. For Null Text, enter:

- All -





- 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. On Source, for 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. On Identify Item Name and Display Position:
  - a. For Item Name, enter P6\_PRIORITY.
  - b. For Region, select **Issue Report Parameters**.
  - c. Click **Next**.
5. On 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. On 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. On Identify Item Name and Display Position:
  - a. For Item Name, enter P6\_RELATED\_PROJECT.
  - b. For Region, select **Issue Report Parameters**.



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_PRIORITY = '-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, select **Issues**.
2. 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;

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.

The report is now complete. Click the **Run Page** icon. (See [Figure 10-10](#) on page 10-43).

Figure 10–10 Issues Report

**Issue Report Parameters**

**Identified By**   
**Assigned To** 
**Status** 
**Priority** 
**Related Project**

**Issues**

	Summary ▲	Identified By	Related Project	Assigned To	Status	Priority	Target Resolution Date	Actual Resolution Date
<input type="button" value="EDIT"/>	Access through proxy servers blocks some usage tracking tools	George Hurst	Public Website Operational	Vicky Williams	Closed	High	19-JAN-05	23-JAN-05
<input type="button" value="EDIT"/>	Action plan review dates conflict with effectivity of organizational consolidations for Great Lakes region	Joe Cerno	Employee Satisfaction Survey	Jane Kerry	Open	Medium	10-MAR-05	-
<input type="button" value="EDIT"/>	Auditors' signoff requires full CSB compliance report	Carla Downing	New Payroll Rollout	Carla Downing	Open	High	17-JAN-05	-
<input type="button" value="EDIT"/>	Client software licenses expire for Bangalore call center before cutover	Joe Cerno	Email Integration	Evan Fanner	Closed	High	25-NOV-04	19-NOV-04
<input type="button" value="EDIT"/>	Cooling and Power requirements exceed 90% headroom limit -- variance from Corporate requested	Al Bines	Internal Infrastructure	Karen London	Closed	High	25-DEC-04	20-DEC-04

row(s) 1 - 5 of 28

To change the report parameters, make new selections under Issue Report Parameters and 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. Navigate 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, select the appropriate schema.
 

Since the purpose of this form is to enable users to assign issues, it is assumed users will only update existing records, not create or delete issues.
6. To enforce this assumption, select **Update Only** from the Allowed Operations list and click **Next**.
7. For Table/View Name, select **HT\_ISSUES** and click **Next**.
8. For Displayed Columns, press **CTRL**, select the following columns, and click **Next**:
  - ISSUE\_SUMMARY
  - IDENTIFIED\_BY

- IDENTIFIED\_DATE
  - RELATED\_PROJECT
  - ASSIGNED\_TO
  - STATUS
  - PRIORITY
9. For Primary Key, accept the default **ISSUE\_ID** and click **Next**.
  10. For Primary Key Source, accept the default **Existing trigger** and click **Next**.
  11. For Updatable Columns, press **CTRL**, select the following, and click **Next**:
    - RELATED\_PROJECT
    - ASSIGNED\_TO
    - STATUS
    - PRIORITY
  12. For Page and Region Attributes:
    - a. For Page, enter 8.
    - b. For Page Name, enter `Assign Issues`.
    - c. For Region Title, enter `Assign Issues`.
    - d. Click **Next**.
  13. For Tab Options, accept the default **Do not use tabs** and click **Next**.
  14. On Button Labels:
    - a. For Cancel Button Label, accept the default.
    - b. For Submit Button Label, enter `Apply Changes`.
    - c. Click **Next**.
  15. For Branching, accept the defaults and click **Next**.
  16. Review your selections and 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 only display unassigned issues.

To add lists of values:

1. From the Success page, select **Edit Page**.  
The Page Definition for page 8, Assign Issues, appears.
2. Under Regions, select **Assign Issues**.
3. In **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.
2. Under Report Column Attributes:
  - a. For ISSUE\_SUMMARY, edit the existing Heading to read:  
Summary
  - b. For all columns except ISSUE\_ID, select **Sort**.
  - c. For IDENTIFIED\_DATE, for **Sort Sequence**, select **1**.
3. Click the **Edit** icon to the left of IDENTIFIED\_BY and edit the following attributes:
  - a. Scroll down to Tabular Form Element. From the Display As list, select **Display as Text (based on LOV, does not save state)**.
  - b. Scroll down to List of Values. From Named LOV list, select **PEOPLE**.
  - c. Click the Next button (>) at the top of the page to navigate to IDENTIFIED\_DATE.
4. Edit the following attributes for IDENTIFIED\_DATE:
  - a. Under Column Formatting, enter DD-MON-YYYY in Number/Date Format.
  - b. Click the Next button (>) at the top of the page to navigate to the next column.
5. Edit the following attributes for RELATED\_PROJECT:
  - a. Scroll down to Tabular Form Element. From Display As, select **Select List (Named LOV)**.
  - b. Scroll down to List of Values. From the Named LOV list, select **PROJECTS**.
  - c. Click the Next button (>) at the top of the page to navigate to the next column.
6. Edit the following attributes for ASSIGNED\_TO:
  - a. Scroll down to Tabular Form Element. From Display As, select **Select List (Named LOV)**.
  - b. Scroll down to List of Values:
    - From the Named LOV list, select **PEOPLE**.
    - For Display Null, select **Yes**.
    - For Null Text, enter a hyphen (-).
  - c. Click the Next button (>) at the top of the page to navigate to the next column.
7. Edit the following attributes for STATUS:
  - a. Scroll down to Tabular Form Element. From Display As, select **Select List (Named LOV)**.
  - b. Scroll down to List of Values. From the Named LOV lists, select **STATUS**.
  - c. Click the Next button (>) at the top of the page to navigate to the next column.
8. Edit the following attributes for PRIORITY:
  - a. Scroll down to Tabular Form Element. From Display As, select **Select List (Named LOV)**.

- b. Scroll down to 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.

9. Scroll down to Sorting. Under Ascending and Descending Image, select the light gray arrow.

10. Under Messages, enter the following in When No Data Found Message:

No Unassigned Issues.

11. Click **Apply Changes**.

The wizard created an unnecessary Cancel button.

To delete the Cancel button:

1. On the Page Definition for page 8, select the **CANCEL** button.
2. Click **Delete**.  
A dialog box displays prompting you to confirm the deletion.
3. Click **OK** to confirm your selection.

The tabular form is now complete. To view the new form, click the **Run Page** icon. (See [Figure 10-11](#) on page 10-46.)

**Figure 10-11 Assign Issues**

Assign Issues <span style="float: right;">Apply Changes</span>						
Summary	Identified By	Identified Date ▲	Related Project	Assigned To	Status	Priority
Emergency Response plan failed county inspector's review at buildings 2 and 5	Al Bines	20-DEC-2004	Internal Infrastructure ▼	▼	Open ▼	High ▼
Review rollout schedule with HR VPs/Directors	Irene Jones	25-DEC-2004	Employee Satisfaction Survey ▼	▼	Closed ▼	Medium ▼
Need better definition of terms like work group, department, and organization for categories F, H, and M-W	Joe Cerno	16-JAN-2005	Employee Satisfaction Survey ▼	▼	Open ▼	Low ▼
Multi-region batch trial run schedule and staffing plan due to directors by end of phase review	Carla Downing	24-JAN-2005	New Payroll Rollout ▼	▼	Open ▼	High ▼

1 - 4

To assign an issue, make a selection from the Assigned To list and 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 Target Resolution Dates Report](#)
- [Add an Average Days to Resolve Report](#)

### 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 a singular result and the second statement gathers information having multiple results.

To add an Issue Summary by Project report:

1. Navigate to the Application home page.
2. Click **Create Page**.
3. Select **Report** and click **Next**.
4. Select **SQL Report** and click **Next**.
5. On Page Attributes:
  - a. For Page, enter 9.
  - b. For 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, enter the following **SQL SELECT** statement and click **Next**:

```
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))
FROM ht_issues
WHERE related_project = :P9_PROJECT
```

8. For Report Attributes:
  - a. For Report Template, select **default: vertical report, look 1 (include null columns)**.
  - b. For Region Name, enter `Issue Summary by Project`.
  - c. 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 the type of HTML region container you want to create. Select **HTML** and click **Next**.
5. For Display Attributes:
  - a. For Title, enter `Issue Summary Report Parameters`.
  - b. For Display Point, select **Page Template Body (2. items below region content)**.
  - c. For Sequence, enter 5.
  - d. 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. On Display Position and Name:
  - a. For Item Name, enter `P9_PROJECT`.
  - b. For Region, select **Issue Summary Report Parameters**.
  - c. Click **Next**.
5. On List of Values:
  - a. For Named LOV, select **PROJECTS**.
  - b. For Null Text, enter:

- Select -

- c. For **Null Value**, enter -1.
- d. Click **Next**.
6. On **Item Attributes**, accept the defaults and click **Next**.
7. On **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. On **Button Attributes**:
  - a. For **Button Name**, enter P9\_GO.
  - b. For **Button Style**, select **Template Based Button**.
  - c. For **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**, select **Report** adjacent 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:
  - 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 click **Next**.
3. For Report Implementation, select **SQL Report** and click **Next**.
4. On Display Attributes:
  - a. For Title, enter `Assignments by Status`.
  - b. For Column, select **2**.
  - c. Click **Next**.
5. For Source:
  - a. In **Enter SQL Query**, enter:
 

```
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. For Rows Per Page, enter 20.
  - c. For Break Columns, select **Column 1**.
  - d. Click **Next**.

6. For Conditional Display:
  - a. From 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, select **Report** adjacent 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 and enter the following in When No Data Found Message:  
No issues found.
6. Click **Apply Changes.**

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. Your report should resemble [Figure 10-12](#).

**Figure 10-12 Issue Summary by Project Report**

The screenshot shows a web interface for an issue tracking application. At the top, there is a section titled "Issue Summary Report Parameters" with a dropdown menu for "Project" set to "Employee Satisfaction Survey" and a "Go" button. Below this, the main report area is divided into two sections. The left section, titled "Issue Summary by Project", displays a list of statistics: First Issue Identified: 25-DEC-04, Last Issue Closed: 23-JAN-05, Total Issues: 7, Open Issues: 4, On-Hold Issues: 0, Closed Issues: 3, Open Issues with No Priority: 0, Open Issues with High Priority: 0, Open Issues with Medium Priority: 3, and Open Issues Low Priority: 1. The right section, titled "Assignments by Status", contains a table with three columns: "Assigned To", "Status", and "Issues". The table has three rows of data: Jane Kerry (Open, 2), an empty row (Closed, 2), and Irene Jones (Open, 1). Below the table, the text "1 - 3" indicates pagination.

Assigned To	Status	Issues
Jane Kerry	Open	2
	Closed	2
Irene Jones	Open	1

1 - 3

### 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. Navigate to the Application home page.
2. Click **Create Page**.
3. Select **Chart** and click **Next**.
4. Select **Line** and click **Next**.
5. For Page Attributes:
  - a. For Page, enter 10.
  - b. For Page Name and Region Name, enter Resolved by Month Identified.
  - c. Click **Next**.
6. For Tab (Optional), accept the default **Do not use tabs** and click **Next**.
7. For Query:
  - a. For Series Name, enter Resolved.
  - b. In **SQL**, enter:

```
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.
8. Click **Next**.
9. Review your selections and click **Finish**.

Next, add a correct axis label and turn off the Chart Title and legend.

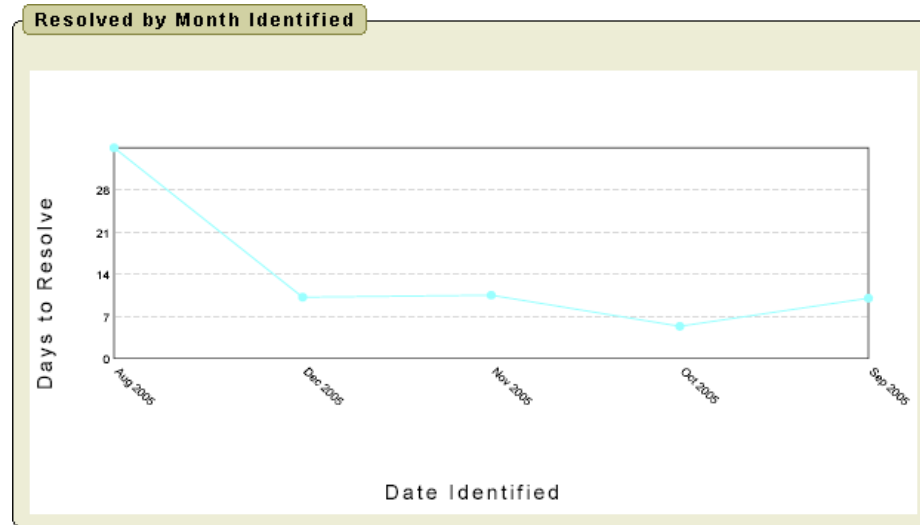
To edit the chart:

1. From the Success page, select **Edit Page**.  
The Page Definition for page 10, Resolved by Month Identified, appears.
2. Under Regions, select **Chart**, adjacent to Resolved by Month Identified.
3. Under Chart Settings:
  - a. For Chart Height, enter 300.
  - b. Disable **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**.

To view your newly created line chart, click the **Run Page** icon. Your line chart should resemble [Figure 10-13](#).

**Figure 10-13 Resolved by Month Identified Line Chart**



### Add a Target Resolution Dates Report

The Target Resolution Dates report is a calendar which displays issues that have not yet closed and 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. Navigate 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. For Page, enter 11.
  - b. For 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. 1.For Table/View Owner:
  - a. For **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.** For 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** again and click **Next**.**

**5. For Display Position and Name, enter `P11_PROJECT` for Item Name, accept the remaining defaults, and click **Next**.**

**6. For List of Values:**

**a.** For Named LOV, select **PROJECTS**.

**b.** For 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, enter the following for **Default**:**

-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. For Button Name, enter `P11_GO`.
  - b. For Button Style, select **Template Based Button**.
  - c. For 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, select **CAL** to the left of Target Resolution Dates.
2. Scroll down to Column Link:
  - a. For Page, enter 7.
  - b. For Clear Cache, enter 7.
  - c. For Set these items, enter:
 

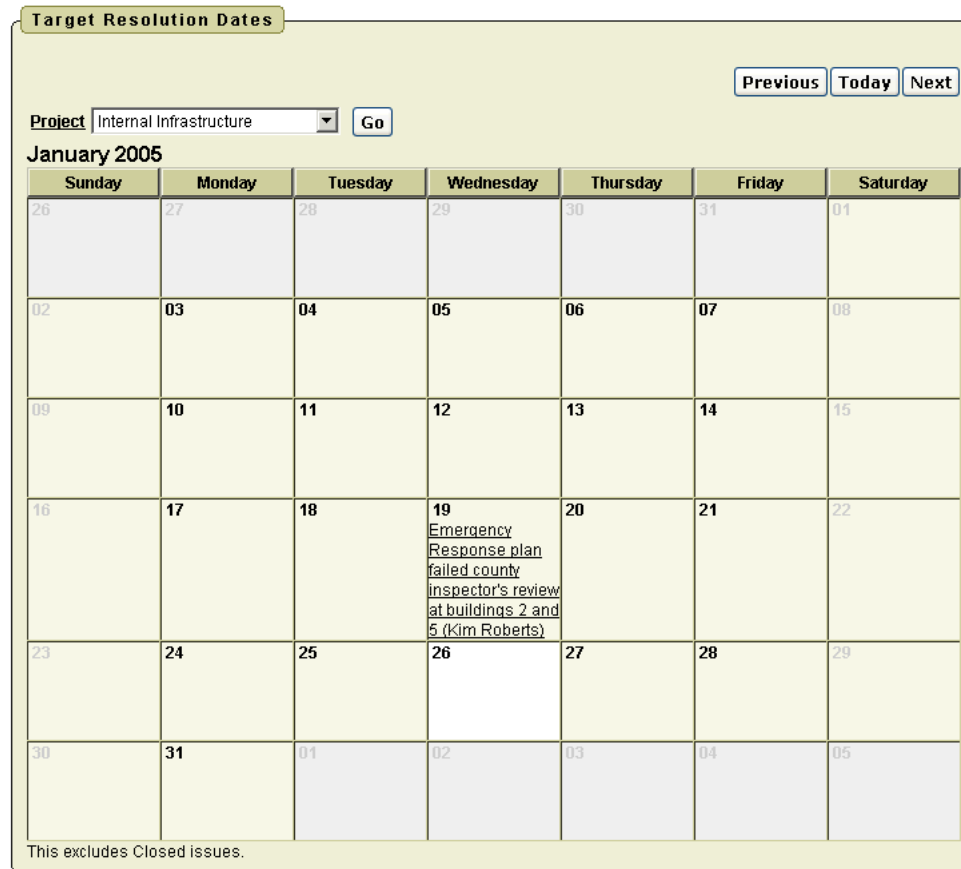
```
P7_ISSUE_ID, P7_PREV_PAGE
```
  - d. For With these values, enter:
 

```
#ISSUE_ID#, 11
```
3. Select the **Region Definition** tab at the top of the page.
4. Scroll down to Header and Footer Text.
5. Enter the following in Region Footer:
 

```
This excludes Closed issues.
```
6. Click **Apply Changes**.

To see your newly created calendar, click the **Run Page** icon. Your report should resemble [Figure 10-14](#) on page 10-56. Note that you can click the text displayed for an issue to display the Edit Issue page. To return to the calendar, click **Cancel**.

**Figure 10–14 Target Resolution Dates Report**



**Add an Average Days to Resolve Report**

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 that by assigned person.

To add the Average Days to Resolve report:

1. Navigate to the Application home page.
2. Click **Create Page**.
3. Select **Chart** and click **Next**.
4. Select **Bar (HTML)** and click **Next**.
5. On Page Attributes:
  - a. For Page, enter 12.
  - b. For Page Name, enter Average Days to Resolve.
  - c. For Region Name, enter Average Days to Resolve.
  - d. Click **Next**.
6. For Tab (Optional), accept the default **Do not use tabs** and click **Next**.
7. On Chart Definition:
  - a. In **Chart SQL**, replace the existing statements with the following:

```

SELECT NULL 1,
       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

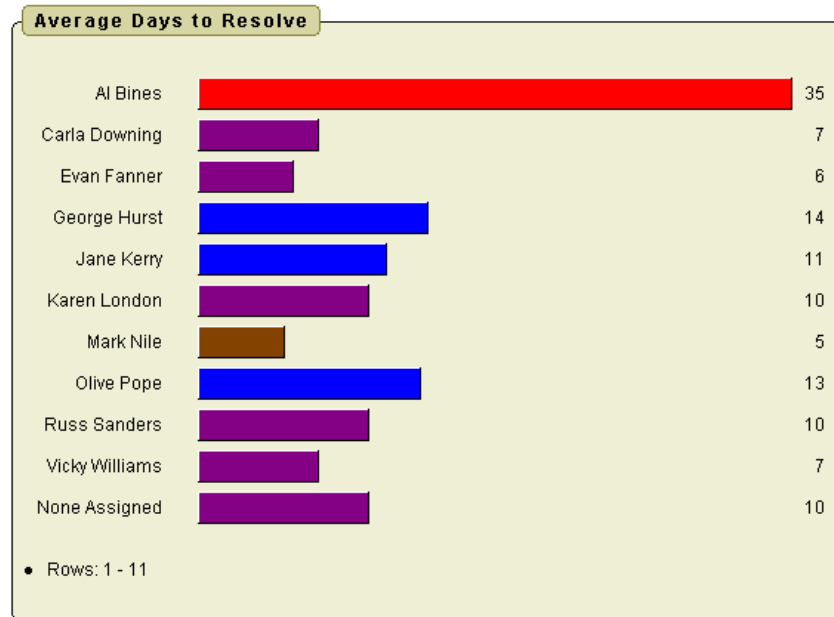
```

In the above SELECT statement:

- The first item selected is the link. Because this report does not link to any other page, 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. Click **Next**.
8. Review your selections and click **Finish**.

To view your newly created bar chart, select **Run Page**. Your report should resemble [Figure 10-15](#).

**Figure 10-15 Average Days to Resolve Report**



## Add Content to the Home Page

Now that you have completed all the detail pages, next you 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 Reports Menu](#)
- [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 Reports Menu**

First, you add a menu implemented as a list.

To add a menu:

1. Navigate to the Application home page.
2. Click the **Shared Components** icon.
3. Under Navigation, select **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 **Create List Entry**.
2. For List Entry Label, enter `Assign Issues`.
3. Under Target:
  - a. For Page, select **8**.
  - b. Select **reset pagination for this page**.
4. Click **Create**.

Now you need to create four more list items, one for each of the reports in your application.

To add list items for each of the other reports in your application:

1. Click **Create List Entry**.

2. To define list item attributes for Issues:
  - a. For Sequence, enter 20.
  - b. For List Entry Label, enter `Issues`.
  - c. Under the Target section:
    - For Page, select 6.
    - Select **reset pagination for this page**.
    - For Clear Cache, enter 6.

This clears any selections for page 6 from the session state.
3. Click **Create and Create Another**.
4. To define list item attributes for Issue Summary:
  - a. For Sequence, enter 30.
  - b. For List Entry Label, enter `Issue Summary by Project`.
  - c. Under the Target section:
    - For Page, select 9.
    - Select **reset pagination for this page**.
    - For **Clear Cache**, enter 9.
5. Click **Create and Create Another**.
6. To define list item attributes for Resolved by Month Identified:
  - a. For Sequence, enter 40.
  - b. For List Entry Label, enter `Resolved by Month Identified (chart)`.
  - c. Under the Target section, select **10** for Page.
7. Click **Create and Create Another**.
8. To define list item attributes for Target Resolution Dates:
  - a. For List, select **Main Menu**.
  - b. For Sequence, enter 50.
  - c. For List Entry Label, enter `Target Resolution Dates (calendar)`.
  - d. Under the Target section:
    - For Page, select **11**.
    - Select **reset pagination for this page**.
9. Click **Create and Create Another**.
10. To define list item attributes 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.  
The Page Definition for page 12, Average Days to Resolve, appears.
2. In the Page field, enter 1 and click **Go**.
3. Click **Edit Attributes**.
4. Locate Display Attributes.
5. From the Page Template list, select **No Tabs with Sidebar**.
6. Click **Apply Changes**.

Next, create a region to contain your menu.

To create a a new region:

1. Under Regions, click the **Create** icon.
2. Select **List** and click **Next**.
3. For Display Attributes:
  - a. For Title, enter Menu.
  - b. For Region Template, select **No Template**.
  - c. For 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. Navigate to the Application home page.
2. Click the **Shared Components** icon.
3. Under Navigation, select **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 list item attributes for the first list item:

- a. For List Entry Label, enter:  
     &nbsp;
- b. Under Target, select **1** for Page.
3. Click **Create and Create Another**.
4. To define list item attributes 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 list item attributes 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.
2. Under Regions, click the **Create** icon.
3. Select **List** and click **Next**.
4. For Display Attributes:
  - a. For Title, enter **Maintenance**.
  - b. For Region Template, select **No Template**.
  - c. For 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 the type of HTML region container you want to create. Select **HTML** and click **Next**.
4. For Display Attributes:
  - a. For Title, enter `Buttons`.
  - b. For Region Template, select **No Template**.
  - c. For 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 click **Next**.
4. For Button Attributes:
  - a. For Button Name, enter `ADD`.
  - b. For Label, enter:  
`Add a New Issue`
  - c. For 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** 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 returns the user to the home page.

7. On Branching:
  - a. For Page, select `7`.
  - b. For Clear Cache, enter `7`.
  - c. For Set these items, enter:  
`P7_PREV_PAGE`
  - d. For 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.

To add a report to display overdue issues:

1. Under Regions, click the **Create** icon.



2. Select **Report** and click **Next**.
3. For Report Implementation, select **SQL Report** and click **Next**.
4. For Display Attributes, enter `Overdue Issues` for Title 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**.

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, select **Report** to the left of Overdue Issues.
2. For Headings Type, select **Custom**.
3. For `ISSUE_ID`, remove the Heading.
4. For `ISSUE_SUMMARY`, edit the Heading to read:

`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 Align, select **center**.
- c. For Heading Align, select **center**.

7. For `ISSUE_ID`, deselect **Show**.

This enables the query to pass in the link, but not display it.

8. 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. Scroll down to 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 the Messages section, 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 the Unassigned Issues report by copying an existing report:

1. Under Regions, click the **Copy** icon.
2. Under Name, select **Overdue Issues**.
3. For To Page, accept the default 1 and click **Next**.
4. For New Region, enter Unassigned Issues.
5. Click **Copy Region**.

To modify the query and edit the report region:

1. Under the Regions section, select **Unassigned Issues** in Region Name.
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. 21.Scroll down to Messages and 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** 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 the **Report** to the left of Recently Opened Issues.
2. For all columns:
  - a. Disable sorting by deselecting **Sort**.
  - b. Set Sequence to `-`.
3. Select the **Region Definition** tab.
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 gray up arrow to the right of the Edit 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 the Messages section, 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 **Pie** and click **Next**.
4. For Display Attributes, enter Open Issues by Project in Title and click **Next**.
5. 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.

#### 6. Click **Create Region**.

To edit the chart.

1. Under Regions, select **Chart** to the left of 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**.

To view the revised page, click the **Run Page** icon. Your home page should resemble [Figure 10-16](#).

**Figure 10-16** Revised Home Page

The screenshot shows the TWINTERS application home page. The top right corner has the text 'TWINTERS | [Print](#) | [Logout](#)'. Below this is a button labeled 'Add an Issue'. The left sidebar contains several navigation links: [Assign Issues](#), [Issues](#), [Issue Summary by Project](#), [Resolved by Month Identified \(chart\)](#), [Target Resolution Dates \(calendar\)](#), [Average Days to Resolve \(chart\)](#), [Projects](#), and [People](#). The main content area is divided into two sections: 'Overdue Issues' and 'Unassigned Issues'.

**Overdue Issues**

Priority	Summary	Assigned To	Target Resolution Date	Project Name
Low	<a href="#">EDIT</a>	Tucker Uberton	05-NOV-04	Email Integration
Medium	<a href="#">EDIT</a>	Olive Pope	25-DEC-04	New Payroll Rollout
High	<a href="#">EDIT</a>	Carla Downing	17-JAN-05	New Payroll Rollout
High	<a href="#">EDIT</a>	Kim Roberts	19-JAN-05	Internal Infrastructure
Medium	<a href="#">EDIT</a>	Scott Tiger	19-JAN-05	Public Website Operational

1 2

**Unassigned Issues**

Priority	Summary	Target Resolution Date	Project Name	Identified By
High	<a href="#">EDIT</a>	08-FEB-05	New Payroll Rollout	Carla Downing
Low	<a href="#">EDIT</a>	08-FEB-05	Employee Satisfaction Survey	Joe Cerno

1

## 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. When you created your application, the wizard automatically created a breadcrumb.

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. Navigate to the Application home page.
2. Click **Shared Components**.
3. Under Navigation, select **Breadcrumbs**.
4. Select the breadcrumb, **Breadcrumb**.

### **Add Breadcrumb Entries**

Next, add breadcrumb entries.

To edit the existing breadcrumb entry for page 1:

1. Under Breadcrumb Entries, click **Page 1**.
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**.

To create a breadcrumb entry for page 2:

1. Click **Create Breadcrumb Entry**.
2. Under Breadcrumb, enter 2 in Page.
3. Under Entry:
  - a. For Parent Entry, select **Home**.
  - b. For Short Name, enter `Projects`.
4. Under Target, enter 2 in Page.
5. Click **Create**.

To create a breadcrumb entry for page 3:

1. Click **Create Breadcrumb Entry**.
2. Under Breadcrumb, enter 3 in Page.
3. Under Entry:
  - a. For Parent Entry, select **Projects**.
  - b. For Short Name, enter `Create/Edit Projects`.
4. Under Target, enter 3 in Page.

**5. Click Create.**

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**.

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**.

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**.

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**.

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**.

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**.

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**.

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**.

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. Navigate to the Application home page.
2. Click **Create Page**.
3. Select **Blank Page** and click **Next**.
4. For Page Attributes, enter 0 for Page and click **Next**.
5. For Page Name, enter **Breadcrumbs** for Name and click **Next**.
6. On Tabs, accept the default, **No**, and click **Next**.
7. Review your selections and click **Finish**.

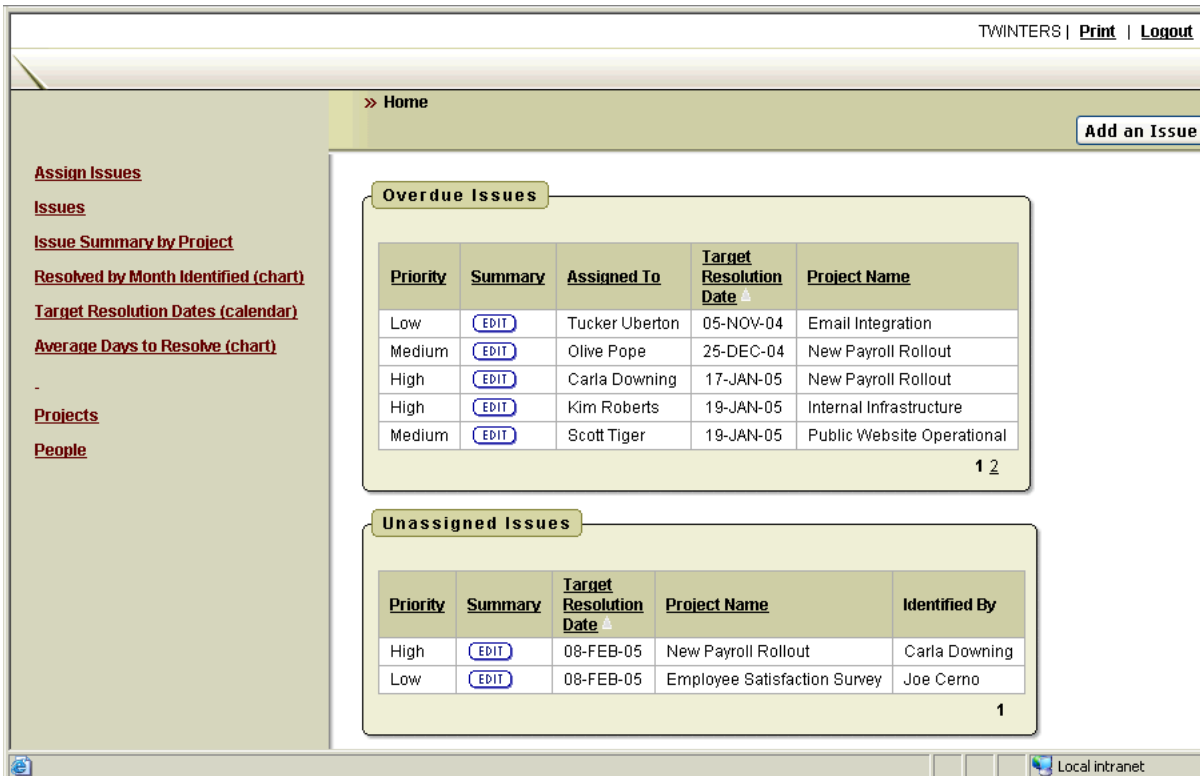
## Create a Region to Contain the Breadcrumb

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 Select a common region type, select **Breadcrumb** and click **Next**.
4. For Breadcrumb Container Region:
  - a. For Region Title, enter **Breadcrumbs**.
  - b. For Region Template, select **No Template**.
  - c. For 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. To see your completed home page, click the **Run Page** icon. Your home page should resemble [Figure 10-17](#) on page 10-72.

Figure 10–17 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 menu changes and watch the breadcrumb change.

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 a 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 `HTMLDB_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 Database Express Edition Application Express User's Guide* to learn about configuring Email Environment settings.

The following is a description of the `SEND` procedure of the `HTMLDB_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 he or she 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. Because Assign Issues is a tabular form, 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.

---

**Important:** The trigger you are about to create sends emails. If you plan on using this application, change the `p_to` and `p_from` within the Trigger Body to your own valid email address. Otherwise, you create emails with invalid addresses each time you assign or reassign an issue.

---

To create a before insert and update trigger on `HT_ISSUES`:

1. On the Database Home Page, click the **Object Browser** icon.
2. Click **Create**.
3. For Select the type of database object you want to create, select **Trigger** and click **Next**.
4. For Name:
  - a. For Schema, select the appropriate schema and click **Next**.

- b. For Table Name, select **HT\_ISSUES** and click **Next**.
5. For Action, keep the default of **Create Trigger** and click **Next**.
6. 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 following, replacing the `p_to` and `p_from` with your own valid email address:

```

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

        HTMLDB_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. Click **Next**.
7. To review the code, expand the **SQL** icon.
8. Click **Finish**.

## Add a 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. Because there is no human interaction to determine if an issue is overdue, 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
- Use SQL Command Processor to run the create commands

To create the package:

1. On the Database Home Page, click the **SQL** icon.
2. Click **SQL Commands**.

SQL Commands appears.

3. Enter the following in the field provided:

```
CREATE OR REPLACE package ht_check_overdue_issues
AS
    PROCEDURE email_overdue;
END;
/
```

4. Click **Run**.

To create the package body:

1. On the Database Home Page, click the **SQL** icon.
2. Click **SQL Commands**.

SQL Commands appears.

3. Enter the following in the field provided, changing p\_to and p\_from to your own email address:

```
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
      HTMLDB_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;
/

```

#### 4. Click Run.

To update demonstration data to include valid email addresses:

1. On the Database Home Page, click the **Object Browser** icon.
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 Database Home Page by clicking the **Home** breadcrumb link.

To create the DBMS\_JOB:

---



---

**Important:** This job generates email. Do not execute the following script if you running on a hosted Oracle Application Express instance. If you are running Oracle Application Express locally and want to test this script, you must update the demonstration data to include valid email addresses as described in the previous procedure.

---



---

1. On the Database Home Page, click the **SQL** icon.
  2. Click **SQL Commands**.
- SQL Commands appears.
3. Enter the following in the field provided:

```

DECLARE
    jobno number;
BEGIN
    DBMS_JOB.SUBMIT(
        job => jobno,
        what => 'BEGIN
                ht_check_overdue_issues.email_overdue;
                END;',
        next_date => SYSDATE,
        interval => 'TRUNC(SYSDATE) + (25/24) '
        );
    COMMIT;
END;
/

```

4. Click **Run**.

This dbms\_job executes just after midnight each day.

## 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 database user account 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 Database Home Page, click **Application Builder**.
2. Select the **Issue Tracker** application.
3. Click **Shared Components**.
4. Under Security, select **Authorization Schemes**.
5. Click **Create**.
6. For Create Authorization Scheme, accept the default **From Scratch** and click **Next**.
7. Under Authorization Scheme Identification, enter the following in Name:

```
USER_CEO_OR_MANAGER
```

8. Under Authorization Scheme:
  - a. For Scheme Type, select **Exists SQL Query**.
  - b. In 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. For 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 since 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.
2. In Page, enter 2 and click **Go**.  
The Page Definition for page 2, Projects, appears.
3. Under Regions, select **Report** to the left of 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. Navigate to the Page Definition for page 2.
2. Under Buttons, select the **Create** button.
3. Under Authorization, select the Authorization Scheme **USER\_CEO\_OR\_MANAGER**.
4. Click **Apply Changes**.

**Associate Objects with the Create/Edit Report** To associate the authorization scheme with the Create/Edit Project page:

1. Navigate to page 3 by clicking the Next Page (>) button.  
The Page Definition for page 3, Create/Edit Project, appears.
2. Click **Edit Attributes**.
3. Scroll down to Security.
4. Select the Authorization Scheme **USER\_CEO\_OR\_MANAGER**.
5. Click **Apply Changes**.

**Associate Objects with the People Report** To associate the authorization scheme with the People report.

1. Navigate to page 4 by clicking the Next Page (>) button.  
The Page Definition for page 4, People, appears.
2. Under Regions, select **Report** to the left of 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. Navigate to the Page Definition for page 5.
2. Under Buttons, select **Create**.
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. Navigate to page 5 by clicking the Next Page (>) button.  
The Page Definition for page 5, Create/Edit Person Information, appears.
2. Click **Edit Attributes**.
3. Scroll down to Security.
4. Select the Authorization Scheme **USER\_CEO\_OR\_MANAGER**.
5. Click **Apply Changes**.

You can test this creating a user with the username 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 10-86

### **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 display are those that are unassigned. Since 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

which 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. Navigate to the Application home page.
2. Click **Shared Components**.
3. Under Security, select **Authorization Schemes**.
4. Click **Create**.
5. For Creation Method, accept the default **From Scratch** and click **Next**.
6. Under Authorization Scheme Identification, 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. Scroll down to 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. Navigate to Application home page.
2. Select page 7 - **Create/Edit Issues**.
3. Under Buttons, select **DELETE**.
  - a. Under Authorization, select the Authorization Scheme **P7\_ASSIGNED\_OR\_PROJECT\_LEAD**.
  - b. Click **Apply Changes**.
4. Under Buttons, select **SAVE**.
  - a. Under Authorization, select the Authorization Scheme **P7\_ASSIGNED\_OR\_PROJECT\_LEAD**.
  - b. Click **Apply Changes**.
5. Under 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. Next specify the type of region container you wish to create. Select **HTML** 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 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 10–18 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 10–18** New Region Displaying Authorization Failure

The screenshot shows a web application interface for 'Create/Edit Issue'. At the top right, there are links for 'TWINTERS | Print | Logout'. Below the header, there is a breadcrumb trail: '>> Home | Create/Edit Issue'. A prominent message box titled 'Not Authorized' contains the text: 'You are not authorized to modify the data for this issue because you are not the Project Lead nor is the issue assigned to you.' Below this message is a large, light-colored rectangular area with a 'Cancel' button on the right side. Underneath, there is a section titled 'Issue Identification' which contains several form fields: 'Issue Summary' (a text input field), 'Issue Description' (a large text area), 'Identified By' (a dropdown menu showing '- Select Person -'), 'Identified Date' (a date field showing '27-JAN-2005' with a calendar icon), and 'Related Project' (a dropdown menu showing '- Select Person -').

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 Database Express Edition 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 Database Express Edition 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 Database Home Page, click the **Application Builder** icon.
2. Select the application you want to export.
3. Click the **Export/Import** icon and then **Export**.
4. For Application, select the application created in this exercise.
5. Click **Export Application**.
6. When prompted, click to **Save** the file.
7. Specify a location on your local hard drive and 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 "[Build the Database Objects](#)" on page 10-6.

---

---

**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

Login to the production instance of Database Home Page:

1. On the Database Home Page, click the **Application Builder** icon.
2. Select the application you want to export.
3. Click the **Export/Import** icon and then **Import**.
4. On Import File:
  - a. For Import File, click the **Browse** button and locate your exported file.
  - b. For File Type, select **Application/Page Export**.
  - c. For 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. For Parse As Schema, select the schema on your production server that contains your application objects.
  - b. For Build Status, you 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. For 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 database user account, 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 database user account, 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 application running
- Select **Edit Application Attributes** to view the application definition within Application Builder

**Load the Data**

The next step in deploying your application would be 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 10-8 and ["Importing, Exporting, Loading, and Unloading Data"](#) in *Oracle Database Express Edition 2 Day DBA Guide*

**Alternate Authentication Mechanisms to Consider**

When the application login page calls the login API with a username 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 username and password against the LDAP directory that you specify.
- Use the built-in Oracle Application Express authentication method, which checks the username and password against the Oracle Application Express user account repository.

Your application is currently using the built-in Oracle Application Express authentication method.

**See Also:** ["Establishing User Identity Through Authentication"](#) in *Oracle Database Express Edition Application Express User's Guide*

**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 is to access the Manage Users page.

To create a new user:

1. Navigate to the Database Home Page and click the **Application Builder** icon.



2. From the Administration list, select **Manage Application Express Users**.
3. Select **Create User**.
4. Under User Identification, enter the required information.
5. Click **Create** 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 displays 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 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 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 each user logs in, he or she receives an 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 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.





```

/
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(wv_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(wv_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.'
/
```

## Create Issues Script

The following script populates Issues table for the issue tracking application.

```
--
-- 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.', '', '')
    /
```