A HIDDEN GEM: THE PL/SQL WEB TOOLKIT

John P. Jeunnette

Prairie Systems Group, Limited

Introduction

The title of this presentation refers to components included with the Oracle database that produce dynamically-generated HTML to implement web-based applications written in PL/SQL. The PL/SQL Web Toolkit is only hidden in the sense that it is frequently overlooked in favor of Open Source technologies like J2EE. I have found it to be an exceptionally productive way to build web-based applications that run in any client browser using the programming knowledge that has served me so well for a long time. In fact, the PL/SQL Web Toolkit forms the underlying basis for the Webview, Webdb, and HTML DB products. There is even an Oracle Designer "Web Server Generator" option if you prefer the CASE (Computer Aided Software Engineering) route. Even more important to a database-centric developer like me is that the code lives in the database with all the advantages that provides rather than on a web server somewhere.

This paper will review a simple data maintenance application (insert, update, and delete operations) to review the concepts and practical realities of this development paradigm. The appendices include extensive code demonstrating the concepts described.

So how does the PL/SQL Web Toolkit fit with other HTML/HTTP/web development technologies such as PL/SQL Server Pages (PSP), Java Server Pages (JSP), Active Server Pages (ASP), .NET, and Oracle Developer Webforms? The major advantage, in my opinion (your mileage may vary), is that the PL/SQL Web Toolkit "pushes" HTML with embedded data and application logic rather than a middleware application that processes commands embedded in HTML pages. Oracle Developer Webforms operate as "thick client" applications and require installation of the JInitiator Java runtime environment (JRE) (or use a JRE included in a web browser) where Web PL/SQL Toolkit applications produce only HTML that will display in any browser ("thin client" because nothing needs to be installed in the browser). Javascript, Cascading Style Sheets and technologies like Ajax can easily be incorporated into generated applications. Thin Client applications like PL/SQL Web Toolkit applications (and PSP, JSP, ASP, and .NET for that matter) tend to operate in "block mode". (Remember IBM 3270 terminals? A screen of data is sent to the computer for processing and the computer sends back a new screen.)

To minimize some of the confusion we will refer to dynamic HTML generated by a PL/SQL package displayed in a browser as "Web Server Forms".

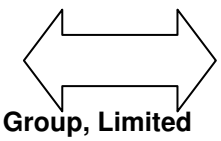
Example Application

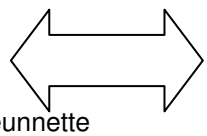
This example application tracks bugs or other issues reported by system users, testers, and management. The master-detail relationships implemented are Applications have Issues and Issues have Actions. Application users select an application, add and edit issues, and add actions to selected issues.

Application Structure

So, how does an Oracle PL/SQL Web Toolkit application actually work? How does an application interact with the user and the database to do something useful?

The first concept to understand is how the client browser communicates with the web server and how the web server communicates with the database. The client browser sends an HTTP (Hypertext Transport Protocol) request to the Webserver (just like you are requesting a page from your favorite web site) but in this case an Apache "mod" (called mod_plsql) reroutes the request to a PL/SQL package in the Oracle database. The connection to the database is defined in a "Data Access Descriptor" (or DAD). Any GET or POST parameters are visible to the PL/SQL package procedure as "normal" PL/SQL parameters. The PL/SQL procedure can read and/or write data from or to the database and then creates HTML code to send back to the client browser using Web Toolkit packages (HTP and HTF). The return HTML is passed back to the client browser by the mod_plsql and the cycle is complete.





HTTP/HTTPS **Oracle HTTP Server + mod_plsql** **Oracle PL/SQL**

The following example is a one table maintenance form to allow the users to enter and track application issues and enhancement requests.

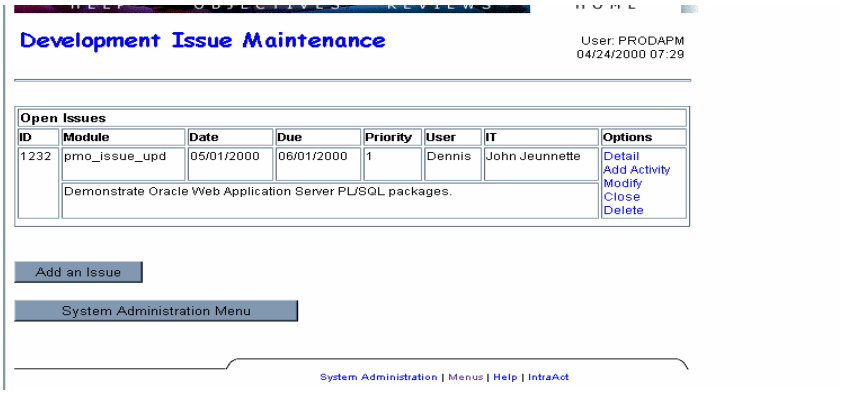
Package Example 1 is an annotated copy of the PITS_ISSUES_UPD table maintenance package showing details on how a typical table maintenance package is structured. (I know it is a bit long, but the reality is that you need a lot of code to get anything done.)

Due to the block mode nature of the Webserver most application modules follow the following general processing scheme:

1. A parameter or selection page is presented with one or more text entry fields, select lists, or other data entry objects (radio buttons or check boxes) for the user to specify what to process. Typically, an exit or "return to menu" button is also provided to exit the module (see the pits_issue_upd package main procedure below for actual code).

2. In response to a submit button on the parameter screen a collection of selected records is displayed. Each record will have "Modify" and "Delete" links (with or without buttons). Either at the beginning or the end (or both) of the collection there will be an "Add a new record" button. A "Select another record" or "Exit" button will return to the parameter screen in step 1. (Screen 1 and the display_issues procedure below)

Screen 1:



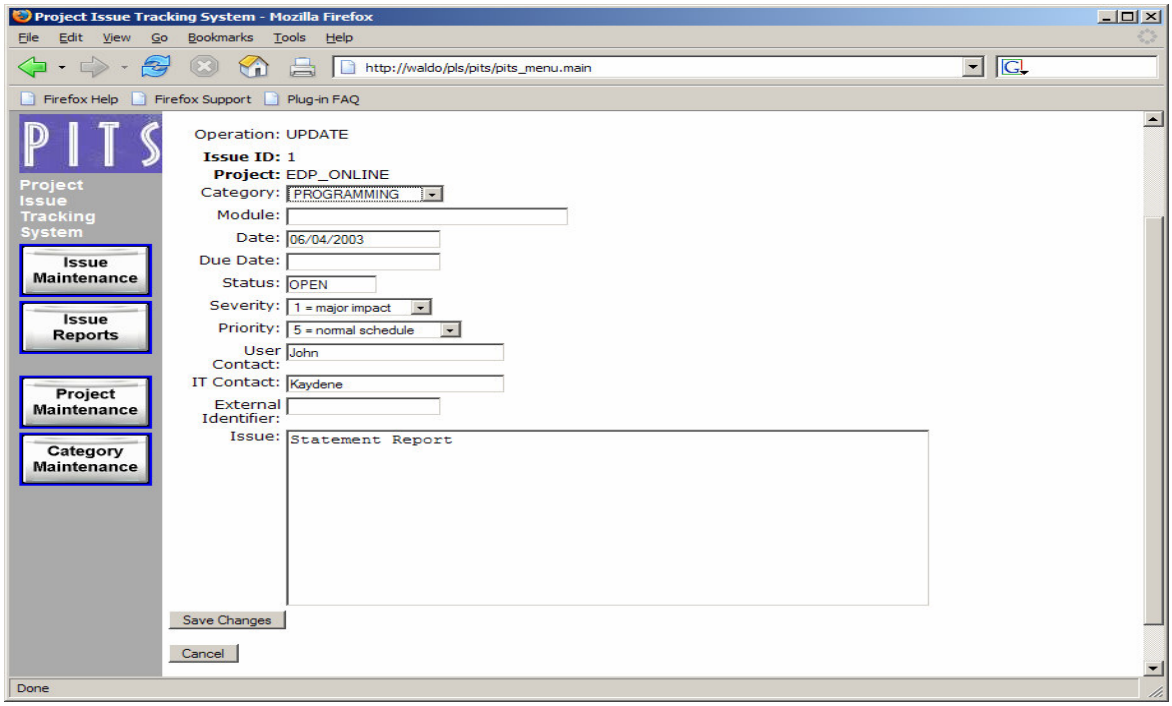
3. The "Add a new record" button will execute the add routine (see the add_issue, process_issue, and validate_issue procedures below).

4. The "Modify" link will execute the modify routine for the current record (see the modify_issue, process_issue, and validate_issue procedures below).

5. The "Delete" link will execute the delete routine for the current record (see the delete_verification and delete_issue procedures below).

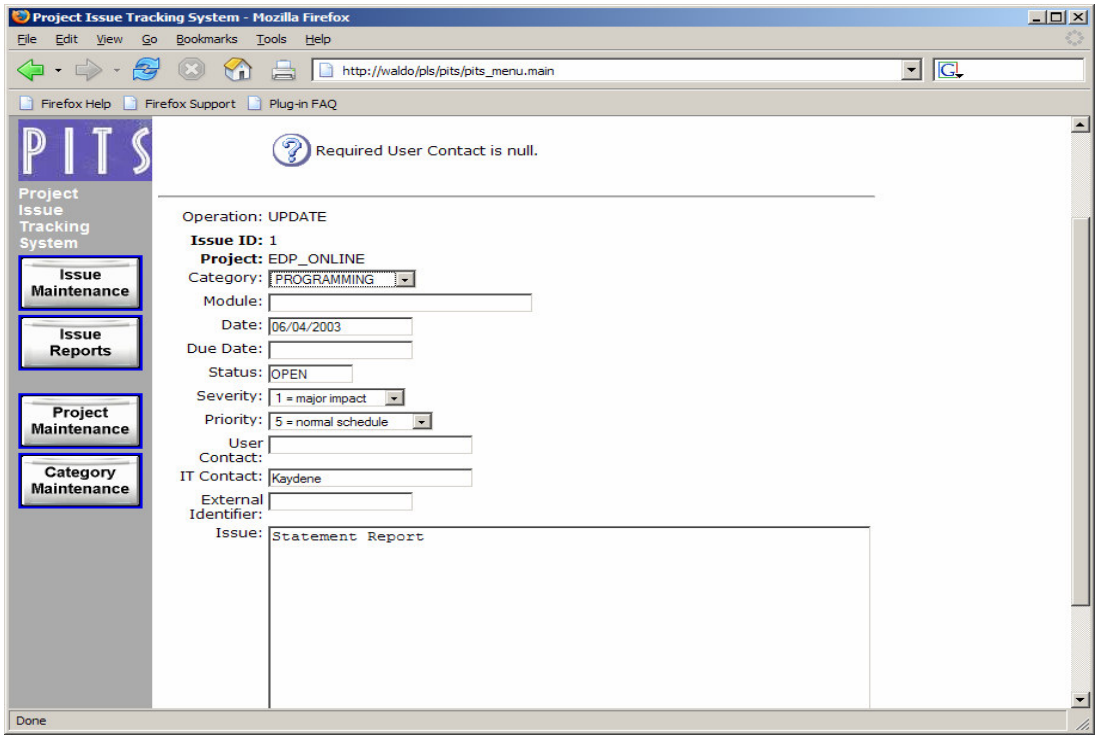
6. The add and modify routines will call a common routine to display a data entry form for the user to either enter new data or modify existing data for the selected record. If an existing record is being modified the checksum will be calculated and saved in a hidden form variable. A "Save Changes" button on the data entry form will navigate to a validation routine. (Screen 2)

Screen 2:



7. The validation routine will check for missing data, non-numeric characters in numeric fields, valid dates, and recalculate the checksum (for updates). If the checksums do not match, the record is reread from the database and sent back to the common data entry routine for re-entry. (Screen 3)

Screen 3:



8. If the data passes the validation tests it is inserted or updated, as appropriate and control is returned to the collection display routine (step 2 above).

Observations

We have "discovered" a number of things over the years:

- Web Server Forms (dynamic HTML) behave like block mode terminals (anybody remember those? Or am I dating myself.) Basically, you display something on the screen and set up data entry fields for the user to enter data and wait for the user to put data in the boxes and press the Submit button. When that happens your procedure receives one or more parameters containing data to be processed. After doing something useful with the data (validation, database activity, or whatever) another screen is displayed with some more widgets for the user to enter data or press buttons.
- The Web PL/SQL Toolkit does implicit commits at the end of each procedure execution.
- In Structured Programming 101 we all learned that each module should have one entry point and one exit point or at least that when you call a module you should expect that your program will return at some point to the statement following the call statement. A Web Server Form may not follow that convention. You (or the user) can direct the program flow almost anywhere so each procedure must be self-contained with any data required by subsequent procedures.
- You can put a lot of effort into making a HTML generated page look a certain way but the first time someone uses a different browser you can almost guarantee that the page will not look anything like you wanted it to. So concentrate on the operational aspects of the application and let the browser display the pages the way it wants to.
- Embedded spaces or non-alphanumeric characters in generated URLs can be a problem. We ended up making sure all URLs had spaces replaced with '%20' and ampersands replaced with '&'. Likewise, passing data like rowids on the URL line (GET parameters) is next to impossible because the rowid can contain characters that confuse HTTP.
- A simple one table maintenance package may contain ten pages of PL/SQL code, while more complicated packages require 70 to 80 pages of code.
- All data is passed from the client browser to the PL/SQL procedure as string data. If you define a number parameter that may receive data entered by a user and the user enters a non-numeric character the procedure call will fail (actually, the

PL/SQL processor looks for an overloaded procedure and does not find one). It is a better practice to define the parameter as a varchar2 and then verify that the user has entered a numeric value. If the value is not numeric the routine can report the error and ask the user to reenter.

- Every routine called by an URL should have an exception block to trap errors. If there is no exception block the webserver basically only tells you that something failed with no context or information for debugging.

Conclusions/Issues

The Oracle PL/SQL Web Toolkit is an excellent way to develop easy-to-use wide-distribution applications. It allows you to leverage your PL/SQL skills to develop web-based applications with minimum operating overhead.

I find the development and application execution environment far easier to understand and deal with than "embedded" technologies like Active Server Pages (ASP) or Java Server Pages (JSP).

Author

John Jeunnette is a codeslave for Prairie Systems Group, Limited in Denver, Colorado developing applications with the Oracle Designer and Developer Toolsets and the PL/SQL Web Toolkit. He is actively involved with the Oracle Development Tools User Group and the Rocky Mountain Oracle Users Group because both groups are a great way to share information. He may be contacted at (303) 703 3940 or JohnJeunnette@PrairieSystemsGroup.com. Our web site (www.prairiesystemsgroup.com) is entirely dynamic HTML generated from the database.

Appendicies: Package Examples

Example 1 is the selection and add, edit, and delete processing procedures for the Issue maintenance routine. Issue activity maintenance is left as an exercise for the reader. Example 2 is a dynamic SQL reporting procedure.

Package Example 1

```
/*
 *   pits_issue_upd.sql
 *
 *   pits_issue_upd package definition script
 *
 */
set scan off
--
CREATE OR REPLACE PACKAGE pits_issue_upd AS
/*
 *   Package:  pits_issue_upd
 *
 *   Purpose:  Project Issue Tracking System Issue Data maintenance
 *
 *   1. 30 Aug 1999  Jeunnette  Initial implementation.
 *   2.  2 Mar 2004  Jeunnette  Addition of external_identifier.
 *   3. 26 Sep 2004  Jeunnette  Fix textarea processing.
 *   4.  2 Nov 2004  Jeunnette  Add issue severity (1-10, default 5).
 *
 */
--
--  Package global/persistent data.
--
  version constant varchar2(10) := '4';
--
--  Global functions and procedures
--
  PROCEDURE main;
```

```
--
  PROCEDURE dispatch( display_option in varchar2 default null,
                      add_option in varchar2 default null,
                      project_identifier in varchar2 default null,
                      issue_category in varchar2 default null,
                      issue_status in varchar2 default null );
--
  PROCEDURE display_project( project_identifier in varchar2 default null );
--
  PROCEDURE display_issues( project_identifier in varchar2 default null,
                            issue_category in varchar2 default null,
                            issue_status in varchar2 default null );
--
  PROCEDURE issue_detail( issue_rowid in varchar2 default null );
--
  PROCEDURE modify_issue( issue_rowid in varchar2 default null );
--
  PROCEDURE add_issue( project_identifier in varchar2 default null,
                       issue_category in varchar2 default null,
                       issue_status in varchar2 default null );
--
  PROCEDURE process_issue( operation in varchar2 default null,
                           pits_issues_rowid varchar2 default null,
                           pits_issues_checksum number default null,
                           issue_id in number default null,
                           project_identifier in varchar2 default null,
                           issue_category in varchar2 default null,
                           module_name in varchar2 default null,
                           external_identifier in varchar2 default null,
                           issue_date in varchar2 default null,
                           due_date in varchar2 default null,
                           severity in number default null,
                           priority in number default null,
                           issue_status in varchar2 default null,
                           user_contact in varchar2 default null,
                           it_contact in varchar2 default null,
                           issue_description in varchar2 default null,
                           validation_message in varchar2 default null );
--
  PROCEDURE validate_issue( operation in varchar2 default 'UPDATE',
                            pits_issues_rowid in varchar2 default null,
                            pits_issues_checksum in number default null,
                            issue_id in number default null,
                            project_identifier in varchar2 default null,
                            issue_category in varchar2 default null,
                            module_name in varchar2 default null,
                            external_identifier varchar2 default null,
                            issue_date in varchar2 default null,
                            due_date in varchar2 default null,
                            severity in number default null,
                            priority in number default null,
                            issue_status in varchar2 default null,
                            user_contact in varchar2 default null,
                            it_contact in varchar2 default null,
                            issue_description in varchar2 default null );
--
  PROCEDURE update_issue( issue_id in number default null,
                          issue_category in varchar2 default null,
                          module_name in varchar2 default null,
                          external_identifier in varchar2 default null,
                          issue_date in varchar2 default null,
                          due_date in varchar2 default null,
                          severity in number default null,
                          priority in number default null,
                          issue_status in varchar2 default null,
                          user_contact in varchar2 default null,
                          it_contact in varchar2 default null,
                          issue_description in varchar2 default null );
--
  PROCEDURE insert_issue( project_identifier in varchar2 default null,
                          issue_category in varchar2 default null,
                          module_name in varchar2 default null,
                          external_identifier in varchar2 default null,
```

```
                          issue_date in varchar2 default null,
                          due_date in varchar2 default null,
                          severity in number default null,
                          priority in number default null,
                          issue_status in varchar2 default null,
                          user_contact in varchar2 default null,
                          it_contact in varchar2 default null,
                          issue_description in varchar2 default null );
--
  PROCEDURE close_issue( issue_rowid in varchar2 default null );
--
  PROCEDURE delete_verification( issue_rowid in varchar2 default null );
--
  PROCEDURE delete_issue( issue_rowid in varchar2 default null );
--
  PROCEDURE modify_issue_activity( issue_rowid in varchar2 default null,
                                   issue_activity_rowid in varchar2 default null );
--
  PROCEDURE add_issue_activity( issue_rowid in varchar2 default null );
--
  PROCEDURE process_issue_activity( operation in varchar2 default null,
                                    pits_issue_activity_rowid varchar2 default null,
                                    pits_issue_activity_checksum number default null,
                                    pits_issues_rowid in varchar2 default null,
                                    issue_id in number default null,
                                    activity_date in varchar2 default null,
                                    activity_person in varchar2 default null,
                                    activity_description in varchar2 default null,
                                    validation_message in varchar2 default null );
--
  PROCEDURE validate_issue_activity( operation in varchar2 default 'UPDATE',
                                     pits_issue_activity_rowid in varchar2 default null,
                                     pits_issue_activity_checksum in number default null,
                                     pits_issues_rowid in varchar2 default null,
                                     issue_id in number default null,
                                     activity_date in varchar2 default null,
                                     activity_person in varchar2 default null,
                                     activity_description in varchar2 default null );
--
  PROCEDURE update_issue_activity( issue_activity_rowid in varchar2 default null,
                                   issue_id in number default null,
                                   activity_date in varchar2 default null,
                                   activity_person in varchar2 default null,
                                   activity_description in varchar2 default null );
--
  PROCEDURE insert_issue_activity( issue_id in number default null,
                                   activity_date in varchar2 default null,
                                   activity_person in varchar2 default null,
                                   activity_description in varchar2 default null );
--
  PROCEDURE delete_issue_activity( issue_activity_rowid in varchar2 default null );
--
END pits_issue_upd;
/
show errors
--
CREATE OR REPLACE PACKAGE BODY pits_issue_upd AS
/*
 *  Package:  pits_issue_upd
 *
 *  Purpose:  Automated Performance Management issue data maintenance.
 *
 *   History:
 *
 *   28 Oct 2000  Jeunnette  Initial implementation.
 *   30 Oct 2000  Jeunnette  Add selection by category.
 */
--
/*****************************************************************************/
--
PROCEDURE main
/*
 *   Establish the processing environment and get things going.
```

```
 *
 *    Input:  (none)
 *
 *   Output:  (none)
 *
 */
IS
--
BEGIN
--
  htp.bodyOpen( cattributes=>'onLoad="document.issue_upd.project_identifier.focus()"' );
  htp.bodyClose;
--
  pits_utility.page_header( 'PITS_ISSUE_MAIN', 'Issue Maintenance' );
--
  htp.nl;
--
  htp.tableOpen( cattributes=>'WIDTH=600 BORDER=0' );
--
  htp.formOpen( owa_util.get_owa_service_path || 'pits_issue_upd.dispatch',
                cattributes=>'NAME="issue_upd"' );
--
  htp.tableRowOpen;
  htp.tableData( ' ', cattributes=>'WIDTH=50' );
  htp.tableData( htf.bold('Select a Project:'), 'RIGHT',
                 cattributes=>'WIDTH=150' );
  htp.print( '<td WIDTH=300>' );
  pits_utility.project_select_list( null, null );
  htp.print( '</td>' );
  htp.tableRowClose;
--
  htp.tableRowOpen;
  htp.tableData( '', cattributes=>'WIDTH=50' );
  htp.tableData( htf.bold('Select an Issue Category:'), 'RIGHT',
                 cattributes=>'WIDTH=150' );
  htp.print( '<td WIDTH=300>' );
  pits_utility.issue_category_select_list( null, null );
  htp.print( '</td>' );
  htp.tableRowClose;
--
  htp.tableRowOpen;
  htp.tableData( '', cattributes=>'WIDTH=50' );
  htp.tableData( htf.bold('Select an Issue Status:'), 'RIGHT',
                 cattributes=>'WIDTH=150' );
  htp.print( '<td WIDTH=300>' );
  pits_utility.issue_status_select_list( null, 'OPEN' );
  htp.print( '</td>' );
  htp.tableRowClose;
--
  htp.tableRowOpen;
  htp.tableData( htf.formSubmit( 'display_option', 'Display Issues' ), ccolspan=>2 );
  htp.tableData( htf.formSubmit( 'add_option', 'Add an Issue' ) );
  htp.tableRowClose;
--
  htp.formClose;
--
  htp.tableClose;
--
  pits_utility.page_footer( 'PITS_ISSUE_MAIN', pits_issue_upd.version );
--
EXCEPTION
  when others then
    pits_utility.report_error( 'pits_issue_upd.main' );
--
END main;
--
/*****************************************************************************/
--
PROCEDURE dispatch( display_option in varchar2 default null,
                    add_option in varchar2 default null,
                    project_identifier in varchar2 default null,
                    issue_category in varchar2 default null,
                    issue_status in varchar2 default null )
```

```
/*
 *   Establish the processing environment and get things going.
 *
 *    Input:  display_option
 *            add_option
 *            project_identifier
 *            issue_category
 *            issue_status
 *
 *   Output:  (none)
 *
 */
IS
--
BEGIN
--
  if dispatch.display_option is not null then
    pits_issue_upd.display_issues( dispatch.project_identifier,
                                   dispatch.issue_category,
                                   dispatch.issue_status );
  end if;
--
  if dispatch.add_option is not null then
    pits_issue_upd.add_issue( dispatch.project_identifier,
                              dispatch.issue_category,
                              dispatch.issue_status );
  end if;
--
EXCEPTION
  when others then
    pits_utility.report_error( 'pits_issue_upd.dispatch' );
--
END dispatch;
--
/*****************************************************************************/
--
PROCEDURE display_project( project_identifier in varchar2 default null )
/*
 *   Display the issues for the select project.
 *
 *    Input:  project_identifier
 *
 *   Output:  (none)
 *
 */
IS
--
BEGIN
--
  pits_utility.page_header( 'PITS_ISSUE_MAIN', 'Issue Maintenance' );
--
  htp.formOpen( owa_util.get_owa_service_path || 'pits_issue_upd.display_issues' );
--
  htp.tableOpen( cattributes=>'WIDTH=600' );
--
  htp.tableRowOpen;
  htp.tableData( '', cattributes=>'WIDTH=50' );
  htp.tableData( htf.bold('Project:'), 'right', cattributes=>'WIDTH=150' );
  htp.tableData( display_project.project_identifier );
  htp.tableRowClose;
--
  htp.tableRowOpen;
  htp.tableData( '', cattributes=>'WIDTH=50' );
  htp.tableData( htf.bold('Select an Issue Category:'), 'RIGHT',
                 cattributes=>'WIDTH=150' );
  htp.print( '<td WIDTH=300>' );
  pits_utility.issue_category_select_list( null, null );
  htp.print( '</td>' );
  htp.tableRowClose;
--
  htp.tableRowOpen;
  htp.tableData( '', cattributes=>'WIDTH=50' );
  htp.tableData( htf.bold('Select an Issue Status:'), 'RIGHT',
```

```
                cattributes=>'WIDTH=150' );
  htp.print( '<td WIDTH=300>' );
  pits_utility.issue_status_select_list( null, 'OPEN' );
  htp.print( '</td>' );
  htp.tableRowClose;
--
  htp.tableClose;
--
  htp.nl;
  htp.formSubmit( null, 'Display Issues' );
--
  htp.formClose;
--
  pits_utility.page_footer( 'pits_ISSUE_MAIN', pits_issue_upd.version );
--
EXCEPTION
  when others then
    pits_utility.report_error( 'pits_issue_upd.main' );
--
END display_project;
--
/*****************************************************************************/
--
PROCEDURE display_issues( project_identifier in varchar2 default null,
                          issue_category in varchar2 default null,
                          issue_status in varchar2 default null )
/*
 *   Establish the processing environment and get things going.
 *
 *    Input:  project_identifier
 *            issue_category
 *            issue_status
 *
 *   Output:  (none)
 *
 */
IS
--
  cursor issue_cursor is
         select rowid, issue_id, issue_category, external_identifier,
                to_char(issue_date,'mm/dd/yyyy') issue_date,
                to_char(due_date,'mm/dd/yyyy') due_date,
                severity, priority, issue_status, user_contact, it_contact,
                issue_description, module_name
           from pits_issues
           where project_identifier = display_issues.project_identifier
             and issue_status = nvl(display_issues.issue_status,pits_issues.issue_status)
             and issue_category = nvl(display_issues.issue_category,pits_issues.issue_category)
           order by issue_category, priority, module_name, issue_date;
  issue_count number := 0;
  modify_anchor varchar2(200);
  close_anchor varchar2(200);
  delete_anchor varchar2(200);
  activity_anchor varchar2(200);
  detail_anchor varchar2(200);
  line_color    varchar2(20);
--
  cursor issue_activity_cursor( cursor_issue_id number ) is
         select rowid, to_char(activity_date,'mm/dd/yyyy') activity_date,
                activity_person, activity_description
           from pits_issue_activity
           where issue_id = cursor_issue_id
           order by activity_date desc;
  issue_activity_count number := 0;
  modify_activity_anchor varchar2(200);
  delete_activity_anchor varchar2(200);
BEGIN
--
--  set up the HTML page
--
  pits_utility.page_header( 'ISSUE_MAINTENANCE_MAIN', 'Issue Maintenance' );
--
  htp.nl;
```

```
--
  htp.formOpen( owa_util.get_owa_service_path || 'pits_issue_upd.add_issue' );
--
  htp.formHidden( 'project_identifier', display_issues.project_identifier );
--
  htp.tableOpen( cattributes=>'WIDTH = 600 BORDER CELLSPACING=0' );
  htp.tableHeader( display_issues.project_identifier || ' ' ||
                   display_issues.issue_status || ' ' ||
                   display_issues.issue_category ||
                   ' Issues', 'Left', ccolspan=>'10' );
--
  htp.tableRowOpen;
  htp.tableData( htf.bold('ID') );
  htp.tableData( htf.bold('Priority<br>(1=high)'), 'center' );
  htp.tableData( htf.bold('Module') );
  htp.tableData( htf.bold('Date') );
  htp.tableData( htf.bold('Severity<br>(1=high)'), 'center' );
  htp.tableData( htf.bold('User Contact') );
  htp.tableData( htf.bold('IT Contact') );
  htp.tableData( htf.bold('Due Date') );
  htp.tableData( htf.bold('External Identifier') );
  htp.tableData( htf.bold('Options') );
  htp.tableRowClose;
--
  for issue_record in issue_cursor
  loop
    issue_count := issue_count + 1;
--
    if issue_count = trunc(issue_count/2)*2 then
      line_color := pits_utility.table_even_line_bgcolor;
    else
      line_color := pits_utility.table_odd_line_bgcolor;
    end if;
--
    detail_anchor := owa_util.get_owa_service_path ||
                     'pits_issue_upd.issue_detail' ||
                     '?issue_rowid=' || issue_record.rowid;
    modify_anchor := owa_util.get_owa_service_path ||
                     'pits_issue_upd.modify_issue' ||
                     '?issue_rowid=' || issue_record.rowid;
    delete_anchor := owa_util.get_owa_service_path ||
                     'pits_issue_upd.delete_verification' ||
                     '?issue_rowid=' || issue_record.rowid;
    close_anchor := owa_util.get_owa_service_path ||
                    'pits_issue_upd.close_issue' ||
                    '?issue_rowid=' || issue_record.rowid;
    activity_anchor := owa_util.get_owa_service_path ||
                    'pits_issue_upd.add_issue_activity' ||
                    '?issue_rowid=' || issue_record.rowid;
--
    htp.tableRowOpen( 'left', 'top', cattributes=>'BGCOLOR="' || line_color || '"' );
    htp.tableData( issue_record.issue_id, crowspan=>'2' );
    htp.tableData( to_char(issue_record.priority), 'center' );
    htp.tableData( issue_record.module_name );
    htp.tableData( issue_record.issue_date );
    htp.tableData( to_char(issue_record.severity), 'center' );
    htp.tableData( nvl(issue_record.user_contact,' ') );
    htp.tableData( nvl(issue_record.it_contact,' ') );
    htp.tableData( nvl(issue_record.due_date,' ') );
    htp.tableData( nvl(issue_record.external_identifier,' ') );
    if issue_record.issue_status = 'OPEN' then
      htp.tableData( htf.anchor( detail_anchor, htf.img('/pits_images/pits_details_button.jpg',
cattributes=>'ALT="Details"') ) || htf.nl ||
                     htf.anchor( activity_anchor, htf.img('/pits_images/pits_add_activity_button.jpg',
cattributes=>'ALT="Add Activity"') ) || htf.nl ||
                     htf.anchor( modify_anchor, htf.img('/pits_images/pits_modify_issue_button.jpg',
cattributes=>'ALT="Modify Issue"') ) || htf.nl ||
                     htf.anchor( close_anchor, htf.img('/pits_images/pits_close_issue_button.jpg',
cattributes=>'ALT="Close Issue"') ) || htf.nl ||
                     htf.anchor( delete_anchor, htf.img('/pits_images/pits_delete_issue_button.jpg',
cattributes=>'ALT="Delete Issue"') ), crowspan=>'2' );
    else
```

```
      htp.tableData( htf.anchor( detail_anchor, htf.img('/pits_images/pits_details_button.jpg',
cattributes=>'ALT="Details"') ), crowspan=>'2' );
    end if;
--
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top', cattributes=>'BGCOLOR="' || line_color || '"' );
    htp.tableData( replace(issue_record.issue_description,chr(13),'<br>'), ccolspan=>'8' );
    htp.tableRowClose;
--
    issue_activity_count := 0;
    for issue_activity_record in issue_activity_cursor( issue_record.issue_id )
    loop
      issue_activity_count := issue_activity_count + 1;
      if issue_activity_count = 1 then
        htp.tableRowOpen( 'left', 'top', cattributes=>'BGCOLOR="' || line_color || '"' );
        htp.tableData( '' );
        htp.tableData( htf.bold('Date') );
        htp.tableData( htf.bold('Who') );
        htp.tableData( htf.bold('Activity'), ccolspan=>'6' );
        htp.tableData( htf.bold('Options') );
        htp.tableRowClose;
      end if;
      modify_activity_anchor := owa_util.get_owa_service_path ||
                      'pits_issue_upd.modify_issue_activity' ||
                      '?issue_rowid=' || issue_record.rowid ||
                      '&issue_activity_rowid=' || issue_activity_record.rowid;
      delete_activity_anchor := owa_util.get_owa_service_path ||
                      'pits_issue_upd.delete_issue_activity' ||
                      '?issue_activity_rowid=' || issue_activity_record.rowid;
--
      htp.tableRowOpen( 'left', 'top', cattributes=>'BGCOLOR="' || line_color || '"' );
      htp.tableData( '' );
      htp.tableData( issue_activity_record.activity_date );
      htp.tableData( issue_activity_record.activity_person );
      htp.tableData( replace(issue_activity_record.activity_description,chr(13),'<br>'), ccolspan=>'6' );
      if issue_record.issue_status = 'OPEN' then
        htp.tableData( htf.anchor( modify_activity_anchor,
htf.img('/pits_images/pits_modify_activity_button.jpg', cattributes=>'ALT="Modify Activity"') ) || htf.nl
||
                       htf.anchor( delete_activity_anchor,
htf.img('/pits_images/pits_delete_activity_button.jpg', cattributes=>'ALT="Delete Activity"') ) );
      else
        htp.tableData( ' ' );
      end if;
      htp.tableRowClose;
    end loop;
--
    htp.tableRowOpen( 'left', 'top', cattributes=>'BGCOLOR="' || line_color || '"' );
    htp.tableData( '', ccolspan=>'11' );
    htp.tableRowClose;
--
  end loop;
  htp.TableClose;
--
  htp.nl;
  htp.nl;
--
  if display_issues.issue_status != 'CLOSED' then
    htp.formSubmit( null, 'Add an Issue' );
  end if;
--
  htp.formClose;
--
  htp.formOpen( owa_util.get_owa_service_path || 'pits_issue_upd.main' );
--
  htp.formSubmit( null, 'Return to Project Selection' );
--
  htp.formClose;
--
  pits_utility.page_footer( 'ISSUE_MAINTENANCE_MAIN', pits_issue_upd.version );
--
EXCEPTION
```

```
  when others then
    pits_utility.report_error( 'pits_issue_upd.display_issues' );
--
END display_issues;
--
/*****************************************************************************/
--
PROCEDURE issue_detail( issue_rowid in varchar2 default null )
/*
 *   Display an issue and associated activity.
 *
 *    Input:  issue_rowid
 *
 *   Output:  database data
 *
 */
IS
  cursor issue_cursor is
         select rowid, project_identifier, issue_id, issue_category,
                external_identifier, module_name,
                to_char(issue_date,'mm/dd/yyyy') issue_date,
                to_char(due_date,'mm/dd/yyyy') due_date,
                severity, priority, issue_status, user_contact, it_contact,
                issue_description
           from pits_issues
           where rowid = issue_detail.issue_rowid;
--
  cursor issue_activity_cursor( cursor_issue_id number ) is
         select rowid, to_char(activity_date,'mm/dd/yyyy') activity_date,
                activity_person, activity_description
           from pits_issue_activity
           where issue_id = cursor_issue_id
           order by activity_date desc;
--
  issue_activity_count number := 0;
BEGIN
--
--  set up the HTML page
--
  pits_utility.page_header( 'ISSUE_DETAIL', 'Issue Detail' );
--
  htp.nl;
--
  htp.tableOpen( cattributes=>'WIDTH = 600' );
--
  for issue_record in issue_cursor
  loop
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Project:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.project_identifier, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Issue ID:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( to_char(issue_record.issue_id), ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Category:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.issue_category, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Module:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.module_name, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Date:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.issue_date, ccolspan=>'3' );
    htp.tableRowClose;
--
```

```
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Due Date:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.due_date, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Severity:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.severity, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Priority:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.priority, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('User Contact:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.user_contact, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('IT Contact:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.it_contact, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('External Identifier:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.external_identifier, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.hr( cattributes=>'WIDTH=600' ), ccolspan=>'4' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Issue:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( replace(issue_record.issue_description,chr(13),'<br>'), ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.hr( cattributes=>'WIDTH=600' ), ccolspan=>'4' );
    htp.tableRowClose;
--
    for issue_activity_record in issue_activity_cursor( issue_record.issue_id )
    loop
      issue_activity_count := issue_activity_count + 1;
      if issue_activity_count = 1 then
        htp.tableRowOpen( 'left', 'top' );
        htp.tableData( '' );
        htp.tableData( htf.bold('Date') );
        htp.tableData( htf.bold('Who') );
        htp.tableData( htf.bold('Activity') );
        htp.tableRowClose;
      end if;
--
      htp.tableRowOpen( 'left', 'top' );
      htp.tableData( '' );
      htp.tableData( issue_activity_record.activity_date );
      htp.tableData( issue_activity_record.activity_person );
      htp.tableData( replace(issue_activity_record.activity_description,chr(13),'<br>') );
      htp.tableRowClose;
    end loop;
--
  end loop;
--
  htp.tableClose;
--
  pits_utility.page_footer( 'ISSUE_DETAIL', pits_issue_upd.version );
--
END issue_detail;
--
/*****************************************************************************/
--
```

```
PROCEDURE modify_issue( issue_rowid in varchar2 default null )
/*
 *   Select the requested issue record from the database and call the
 *   processing routine.
 *
 *    Input:  issue_rowid
 *
 *   Output:  (none)
 *
 */
IS
  cursor issue_cursor( cursor_issue_rowid varchar2 ) is
         select rowid, project_identifier, issue_id, issue_category,
                external_identifier, module_name,
                to_char(issue_date,'mm/dd/yyyy') issue_date,
                to_char(due_date,'mm/dd/yyyy') due_date,
                severity, priority, issue_status, user_contact, it_contact,
                issue_description
           from pits_issues
           where rowid = cursor_issue_rowid;
--
  pits_issues_checksum number;
BEGIN
--
  for issue_record in issue_cursor( modify_issue.issue_rowid )
  loop
    pits_issues_checksum := owa_opt_lock.checksum( issue_record.project_identifier ||
                                                   to_char(issue_record.issue_id) ||
                                                   upper(issue_record.issue_category) ||
                                                   issue_record.module_name ||
                                                   issue_record.external_identifier ||
                                                   issue_record.issue_date ||
                                                   issue_record.due_date ||
                                                   to_char(issue_record.severity) ||
                                                   to_char(issue_record.priority) ||
                                                   issue_record.issue_status ||
                                                   issue_record.user_contact ||
                                                   issue_record.it_contact ||
                                                   issue_record.issue_description );
--
    process_issue( 'UPDATE',
                   issue_record.rowid,
                   pits_issues_checksum,
                   issue_record.issue_id,
                   issue_record.project_identifier,
                   issue_record.issue_category,
                   issue_record.module_name,
                   issue_record.external_identifier,
                   issue_record.issue_date,
                   issue_record.due_date,
                   issue_record.severity,
                   issue_record.priority,
                   issue_record.issue_status,
                   issue_record.user_contact,
                   issue_record.it_contact,
                   issue_record.issue_description );
  end loop;
--
END modify_issue;

/*****************************************************************************/

PROCEDURE add_issue( project_identifier in varchar2,
                     issue_category in varchar2 default null,
                     issue_status in varchar2 default null )
/*
 *   Create a new issue record.
 *
 *    Input:  project_identifier
 *            issue_category
 *            issue_status
 *
 *   Output:  (none)
```

```
 *
 */
IS
--
BEGIN
--
  process_issue( 'INSERT',
                project_identifier=>add_issue.project_identifier,
                issue_category=>add_issue.issue_category,
                issue_status=>add_issue.issue_status );
--
END add_issue;
--
/*****************************************************************************/
--
PROCEDURE process_issue( operation in varchar2 default null,
                         pits_issues_rowid in varchar2 default null,
                         pits_issues_checksum in number default null,
                         issue_id in number default null,
                         project_identifier in varchar2 default null,
                         issue_category in varchar2 default null,
                         module_name in varchar2 default null,
                         external_identifier in varchar2 default null,
                         issue_date in varchar2 default null,
                         due_date in varchar2 default null,
                         severity in number default null,
                         priority in number default null,
                         issue_status in varchar2 default null,
                         user_contact in varchar2 default null,
                         it_contact in varchar2 default null,
                         issue_description in varchar2 default null,
                         validation_message in varchar2 default null )
/*
 *   Enter, review and/or modify a menu header record.
 *
 *    Input:  pits_issues fields
 *            validation_message
 *
 *   Output:  database data
 *
 */
IS
--
  this_issue_date varchar2(20);
  this_issue_status pits_issues.issue_status%type;
  this_severity varchar2(1);
  this_priority varchar2(1);
--
BEGIN
--
--  set up the HTML page
--
  htp.bodyOpen( cattributes=>'onLoad="document.issue_upd.issue_category.focus()"' );
  htp.bodyClose;
--
  pits_utility.page_header( 'PROCESS_ISSUE', 'Process Issue' );
--
  htp.nl;
--
  if validation_message is not null then
    pits_utility.display_validation_message( validation_message );
  end if;
--
  htp.formOpen( owa_util.get_owa_service_path || 'pits_issue_upd.validate_issue',
                cattributes=>'NAME="issue_upd"' );
--
  htp.formHidden( 'operation', process_issue.operation );
  htp.formHidden( 'pits_issues_rowid',
                  process_issue.pits_issues_rowid );
  htp.formHidden( 'pits_issues_checksum',
                  process_issue.pits_issues_checksum );
  htp.formHidden( 'issue_id', to_char(process_issue.issue_id) );
  htp.formHidden( 'project_identifier', process_issue.project_identifier );
```

```
--
  htp.tableOpen( cattributes=>'WIDTH = 600' );
--
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( 'Operation:', 'right', cattributes=>'WIDTH = 125' );
  htp.tableData( process_issue.operation );
  htp.tableRowClose;
--
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( '' );
  htp.tableData( '' );
  htp.tableRowClose;
--
  if operation = 'UPDATE' then
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Issue ID:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( to_char(process_issue.issue_id) );
    htp.tableRowClose;
  end if;
--
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( htf.bold('Project:'), 'right', cattributes=>'WIDTH = 125' );
  htp.tableData( process_issue.project_identifier );
  htp.tableRowClose;
--
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( 'Category:', 'right', cattributes=>'WIDTH=125' );
  htp.print( '<td>' );
  pits_utility.issue_category_select_list( null, process_issue.issue_category );
  htp.print( '</td>' );
--  htp.tableData( htf.formtext( 'issue_category', 30, 30, process_issue.issue_category,
--                               cattributes=>'onBlur=this.value=this.value.toUpperCase()' ) );
  htp.tableRowClose;
--
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( 'Module:', 'right', cattributes=>'WIDTH = 125' );
  htp.tableData( htf.formText( 'module_name', 40, 50, process_issue.module_name ) );
  htp.tableRowClose;
--
  if operation = 'INSERT' then
    this_issue_date := to_char(sysdate,'mm/dd/yyyy');
  else
    this_issue_date := process_issue.issue_date;
  end if;
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( 'Date:', 'right', cattributes=>'WIDTH = 125' );
  htp.tableData( htf.formText( 'issue_date', 20, 20, this_issue_date ) );
  htp.tableRowClose;
--
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( 'Due Date:', 'right', cattributes=>'WIDTH = 125' );
  htp.tableData( htf.formText( 'due_date', 20, 20,
                               process_issue.due_date ) );
  htp.tableRowClose;
--
  if operation = 'INSERT' then
    this_issue_status := 'OPEN';
  else
    this_issue_status := process_issue.issue_status;
  end if;
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( 'Status:', 'right', cattributes=>'WIDTH = 125' );
  htp.tableData( htf.formText( 'issue_status', 10, 10, this_issue_status,
                               cattributes=>'onBlur=this.value=this.value.toUpperCase()' ) );
  htp.tableRowClose;
--
  if operation = 'INSERT' then
    this_severity := '5';
  else
    this_severity := to_char(process_issue.severity);
  end if;
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( 'Severity:', 'right', cattributes=>'WIDTH = 125' );
```

```
  htp.print( '<td>' );
  htp.formSelectOpen( 'severity', '' );
  if this_severity = '1' then
    htp.formSelectOption( '1 = major impact', 'SELECTED', cattributes=>'VALUE="1"' );
  else
    htp.formSelectOption( '1 = major impact', cattributes=>'VALUE="1"' );
  end if;
  if this_severity = '2' then
    htp.formSelectOption( '2', 'SELECTED', cattributes=>'VALUE="2"' );
  else
    htp.formSelectOption( '2', cattributes=>'VALUE="2"' );
  end if;
  if this_severity = '3' then
    htp.formSelectOption( '3', 'SELECTED', cattributes=>'VALUE="3"' );
  else
    htp.formSelectOption( '3', cattributes=>'VALUE="3"' );
  end if;
  if this_severity = '4' then
    htp.formSelectOption( '4', 'SELECTED', cattributes=>'VALUE="4"' );
  else
    htp.formSelectOption( '4', cattributes=>'VALUE="4"' );
  end if;
  if this_severity = '5' then
    htp.formSelectOption( '5 = module impact', 'SELECTED', cattributes=>'VALUE="5"' );
  else
    htp.formSelectOption( '5 = module impact', cattributes=>'VALUE="5"' );
  end if;
  if this_severity = '6' then
    htp.formSelectOption( '6', 'SELECTED', cattributes=>'VALUE="6"' );
  else
    htp.formSelectOption( '6', cattributes=>'VALUE="6"' );
  end if;
  if this_severity = '7' then
    htp.formSelectOption( '7', 'SELECTED', cattributes=>'VALUE="7"' );
  else
    htp.formSelectOption( '7', cattributes=>'VALUE="7"' );
  end if;
  if this_severity = '8' then
    htp.formSelectOption( '8', 'SELECTED', cattributes=>'VALUE="8"' );
  else
    htp.formSelectOption( '8', cattributes=>'VALUE="8"' );
  end if;
  if this_severity = '9' then
    htp.formSelectOption( '9 = minimal impact', 'SELECTED', cattributes=>'VALUE="9"' );
  else
    htp.formSelectOption( '9 = minimal impact', cattributes=>'VALUE="9"' );
  end if;
  --
  htp.formSelectClose;
  htp.print( '</td>' );
  htp.tableRowClose;
--
  if operation = 'INSERT' then
    this_priority := '5';
  else
    this_priority := to_char(process_issue.priority);
  end if;
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( 'Priority:', 'right', cattributes=>'WIDTH = 125' );
  htp.print( '<td>' );
  htp.formSelectOpen( 'priority', '' );
  if this_priority = '1' then
    htp.formSelectOption( '1 = immediate attention', 'SELECTED', cattributes=>'VALUE="1"' );
  else
    htp.formSelectOption( '1 = immediate attention', cattributes=>'VALUE="1"' );
  end if;
  if this_priority = '2' then
    htp.formSelectOption( '2', 'SELECTED', cattributes=>'VALUE="2"' );
  else
    htp.formSelectOption( '2', cattributes=>'VALUE="2"' );
  end if;
  if this_priority = '3' then
    htp.formSelectOption( '3', 'SELECTED', cattributes=>'VALUE="3"' );
```

```
  else
    htp.formSelectOption( '3', cattributes=>'VALUE="3"' );
  end if;
  if this_priority = '4' then
    htp.formSelectOption( '4', 'SELECTED', cattributes=>'VALUE="4"' );
  else
    htp.formSelectOption( '4', cattributes=>'VALUE="4"' );
  end if;
  if this_priority = '5' then
    htp.formSelectOption( '5 = normal schedule', 'SELECTED', cattributes=>'VALUE="5"' );
  else
    htp.formSelectOption( '5 = normal schedule', cattributes=>'VALUE="5"' );
  end if;
  if this_priority = '6' then
    htp.formSelectOption( '6', 'SELECTED', cattributes=>'VALUE="6"' );
  else
    htp.formSelectOption( '6', cattributes=>'VALUE="6"' );
  end if;
  if this_priority = '7' then
    htp.formSelectOption( '7', 'SELECTED', cattributes=>'VALUE="7"' );
  else
    htp.formSelectOption( '7', cattributes=>'VALUE="7"' );
  end if;
  if this_priority = '8' then
    htp.formSelectOption( '8', 'SELECTED', cattributes=>'VALUE="8"' );
  else
    htp.formSelectOption( '8', cattributes=>'VALUE="8"' );
  end if;
  if this_priority = '9' then
    htp.formSelectOption( '9 = next release', 'SELECTED', cattributes=>'VALUE="9"' );
  else
    htp.formSelectOption( '9 = next release', cattributes=>'VALUE="9"' );
  end if;
  --
  htp.formSelectClose;
  htp.print( '</td>' );
  htp.tableRowClose;
--
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( 'User Contact:', 'right', cattributes=>'WIDTH = 125' );
  htp.tableData( htf.formText( 'user_contact', 30, 30,
                               process_issue.user_contact ) );
  htp.tableRowClose;
--
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( 'IT Contact:', 'right', cattributes=>'WIDTH = 125' );
  htp.tableData( htf.formText( 'it_contact', 30, 30,
                               process_issue.it_contact ) );
  htp.tableRowClose;
--
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( 'External Identifier:', 'right', cattributes=>'WIDTH = 125' );
  htp.tableData( htf.formText( 'external_identifier', 20, 20,
                               process_issue.external_identifier ) );
  htp.tableRowClose;
--
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( 'Issue:', 'right', cattributes=>'WIDTH=125'  );
  htp.print( '<td>' );
  htp.formTextareaOpen2( 'issue_description', 10, 60, 'LEFT', 'WORD' );
  if process_issue.issue_description is not null then
    htp.print( process_issue.issue_description );
  end if;
  htp.formTextareaClose;
  htp.print( '</td>' );
  htp.tableRowClose;
--
  htp.tableClose;
--
  htp.formSubmit( null, 'Save Changes' );
  htp.formClose;
--
  htp.formOpen( owa_util.get_owa_service_path || 'pits_issue_upd.display_issues' );
```

```
  htp.formHidden( 'project_identifier', process_issue.project_identifier );
  htp.formHidden( 'issue_category', process_issue.issue_category );
  htp.formHidden( 'issue_status', process_issue.issue_status );
  htp.formSubmit( null, 'Cancel' );
  htp.formClose;
--
  pits_utility.page_footer( 'PROCESS_ISSUE', pits_issue_upd.version );
--
END process_issue;
--
/*****************************************************************************/
--
PROCEDURE validate_issue( operation in varchar2 default 'UPDATE',
                          pits_issues_rowid in varchar2 default null,
                          pits_issues_checksum in number default null,
                          issue_id in number default null,
                          project_identifier in varchar2 default null,
                          issue_category in varchar2 default null,
                          module_name in varchar2 default null,
                          external_identifier in varchar2 default null,
                          issue_date in varchar2 default null,
                          due_date in varchar2 default null,
                          severity in number default null,
                          priority in number default null,
                          issue_status in varchar2 default null,
                          user_contact in varchar2 default null,
                          it_contact in varchar2 default null,
                          issue_description in varchar2 default null )
/*
 *   Validate required fields in the issue record.
 *
 *    Input:  (see parameter list)
 *
 *   Output:  validate_message
 *
 */
IS
--
  cursor issue_cursor is
         select rowid, issue_id, project_identifier, issue_category,
                external_identifier, module_name,
                to_char(issue_date,'mm/dd/yyyy') issue_date,
                to_char(due_date,'mm/dd/yyyy') due_date,
                severity, priority, issue_status, user_contact, it_contact,
                issue_description
           from pits_issues
           where rowid = validate_issue.pits_issues_rowid;
--
  validate_message varchar2(2000) := null;
  validation_error boolean := false;
--
  local_issue_date varchar2(50);
  local_due_date varchar2(50);
--
  new_pits_issues_checksum number;
BEGIN
--
  local_issue_date := validate_issue.issue_date;
  if validate_issue.issue_date is null or
     not pits_utility.ValidateDate( validate_issue.issue_date,
                                    local_issue_date ) then
    validation_error := TRUE;
    if validate_message is not null then
      validate_message := validate_message || htf.nl;
    end if;
    validate_message := validate_message ||
                        'Required Start Date is blank or unrecognizable.';
  end if;
--
  local_due_date := validate_issue.due_date;
  if not pits_utility.ValidateDate( validate_issue.due_date,
                                    local_due_date ) then
    validation_error := TRUE;
```

```
    if validate_message is not null then
      validate_message := validate_message || htf.nl;
    end if;
    validate_message := validate_message ||
                        'Due Date is unrecognizable...Please enter a valid date.';
  end if;
--
  if validate_issue.user_contact is null then
    validation_error := true;
    if validate_message is not null then
      validate_message := validate_message || htf.nl;
    end if;
    validate_message := validate_message || 'Required User Contact is null.';
  end if;
--
  if operation = 'UPDATE' then
    for issue_record in issue_cursor
    loop
      new_pits_issues_checksum := owa_opt_lock.checksum(
                                    issue_record.project_identifier ||
                                    to_char(issue_record.issue_id) ||
                                    upper(issue_record.issue_category) ||
                                    issue_record.module_name ||
                                    issue_record.external_identifier ||
                                    issue_record.issue_date ||
                                    issue_record.due_date ||
                                    to_char(issue_record.severity) ||
                                    to_char(issue_record.priority) ||
                                    issue_record.issue_status ||
                                    issue_record.user_contact ||
                                    issue_record.it_contact ||
                                    issue_record.issue_description );
      if pits_issues_checksum != new_pits_issues_checksum then
        validation_error := true;
        if validate_message is not null then
          validate_message := validate_message || htf.nl;
        end if;
        validate_message := validate_message || 'Record changed during processing.';
--
        process_issue( validate_issue.operation,
                       validate_issue.pits_issues_rowid,
                       new_pits_issues_checksum,
                       issue_record.issue_id,
                       issue_record.project_identifier,
                       issue_record.issue_category,
                       issue_record.module_name,
                       issue_record.external_identifier,
                       issue_record.issue_date,
                       issue_record.due_date,
                       issue_record.severity,
                       issue_record.priority,
                       issue_record.issue_status,
                       issue_record.user_contact,
                       issue_record.it_contact,
                       issue_record.issue_description,
                       validate_message );
      end if;
    end loop;
  end if;
  if validation_error then
    process_issue( validate_issue.operation,
                   validate_issue.pits_issues_rowid,
                   validate_issue.pits_issues_checksum,
                   validate_issue.issue_id,
                   validate_issue.project_identifier,
                   validate_issue.issue_category,
                   validate_issue.module_name,
                   validate_issue.external_identifier,
                   validate_issue.issue_date,
                   validate_issue.due_date,
                   validate_issue.severity,
                   validate_issue.priority,
                   validate_issue.issue_status,
```

```
                  validate_issue.user_contact,
                  validate_issue.it_contact,
                  validate_issue.issue_description,
                  validate_message );
  else
    if validate_issue.operation = 'UPDATE' then
      update_issue( validate_issue.issue_id,
                    validate_issue.issue_category,
                    validate_issue.module_name,
                    validate_issue.external_identifier,
                    local_issue_date,
                    local_due_date,
                    validate_issue.severity,
                    validate_issue.priority,
                    validate_issue.issue_status,
                    validate_issue.user_contact,
                    validate_issue.it_contact,
                    validate_issue.issue_description );
--
    else
      insert_issue( validate_issue.project_identifier,
                    validate_issue.issue_category,
                    validate_issue.module_name,
                    validate_issue.external_identifier,
                    local_issue_date,
                    local_due_date,
                    validate_issue.severity,
                    validate_issue.priority,
                    'OPEN',
                    validate_issue.user_contact,
                    validate_issue.it_contact,
                    validate_issue.issue_description );
--
    end if;
--
    owa_util.redirect_url( owa_util.get_owa_service_path ||
                           'pits_issue_upd.display_issues' ||
                           '?project_identifier=' || validate_issue.project_identifier ||
                           '&issue_category=' || validate_issue.issue_category ||
                           '&issue_status=' || validate_issue.issue_status, TRUE );
  end if;
--
END validate_issue;
--
/*****************************************************************************/
--
PROCEDURE update_issue( issue_id in number default null,
                        issue_category in varchar2 default null,
                        module_name in varchar2 default null,
                        external_identifier in varchar2 default null,
                        issue_date in varchar2 default null,
                        due_date in varchar2 default null,
                        severity in number default null,
                        priority in number default null,
                        issue_status in varchar2 default null,
                        user_contact in varchar2 default null,
                        it_contact in varchar2 default null,
                        issue_description in varchar2 default null )
/*
 *   Update the issue record.
 *
 *    Input:  issue_id
 *            (see parameters)
 *
 *   Output:  database data
 *
 */
IS
--
BEGIN
--
  begin
    update pits_issues
```

```
      set issue_category = upper(update_issue.issue_category),
          module_name = update_issue.module_name,
          external_identifier = update_issue.external_identifier,
          issue_date = to_date(update_issue.issue_date,'mm/dd/yyyy'),
          due_date = to_date(update_issue.due_date,'mm/dd/yyyy'),
          severity = update_issue.severity,
          priority = update_issue.priority,
          user_contact = update_issue.user_contact,
          it_contact = update_issue.it_contact,
          issue_description = pits_utility.trim_string(update_issue.issue_description)
      where issue_id = update_issue.issue_id;
  end;
--
EXCEPTION
  when others then
    pits_utility.report_error( 'pits_issue_upd.update_issue' );
--
END update_issue;
--
/*****************************************************************************/
--
PROCEDURE insert_issue( project_identifier in varchar2 default null,
                        issue_category in varchar2 default null,
                        module_name in varchar2 default null,
                        external_identifier in varchar2 default null,
                        issue_date in varchar2 default null,
                        due_date in varchar2 default null,
                        severity in number default null,
                        priority in number default null,
                        issue_status in varchar2 default null,
                        user_contact in varchar2 default null,
                        it_contact in varchar2 default null,
                        issue_description in varchar2 default null )
/*
 *   Insert a issue record.
 *
 *    Input:  (see parameters)
 *
 *   Output:  database data
 *
 */
IS
--
  new_issue_id number := 0;
--
BEGIN
--
  begin
    select pits_sequence.nextval
      into new_issue_id
      from dual;
  exception
    when others then
      pits_utility.report_error( 'pits_issue_upd.insert_issue' );
  end;
--
  begin
    insert into pits_issues
                ( issue_id,
                  project_identifier,
                  issue_category,
                  module_name,
                  external_identifier,
                  issue_date,
                  due_date,
                  severity,
                  priority,
                  issue_status,
                  user_contact, it_contact, issue_description )
      values ( new_issue_id,
               project_identifier,
               upper(insert_issue.issue_category),
               insert_issue.module_name,
```

```
               insert_issue.external_identifier,
               to_date(insert_issue.issue_date,'mm/dd/yyyy'),
               to_date(insert_issue.due_date,'mm/dd/yyyy'),
               insert_issue.severity,
               insert_issue.priority,
               insert_issue.issue_status,
               insert_issue.user_contact,
               insert_issue.it_contact,
               pits_utility.trim_string(insert_issue.issue_description) );
  end;
--
EXCEPTION
  when others then
    pits_utility.report_error( 'pits_issue_upd.insert_issue' );
--
END insert_issue;
--
/*****************************************************************************/
--
PROCEDURE close_issue( issue_rowid in varchar2 default null )
/*
 *   Close an issue.
 *
 *    Input:  issue_rowid
 *
 *   Output:  database data
 *
 */
IS
  this_project_identifier pits_issues.project_identifier%type := null;
  this_issue_category     pits_issues.issue_category%type := null;
  this_issue_status       pits_issues.issue_status%type := null;
--
BEGIN
--
  for issue_record in
    ( select project_identifier, issue_category, issue_status
        from pits_issues
        where rowid = close_issue.issue_rowid )
  loop
    this_project_identifier := issue_record.project_identifier;
    this_issue_category := issue_record.issue_category;
    this_issue_status := issue_record.issue_status;
  end loop;
--
  begin
    update pits_issues
      set issue_status = 'CLOSED'
      where rowid = close_issue.issue_rowid;
  exception
    when others then
      pits_utility.report_error( 'pits_issue_upd.close_issue issue' );
  end;
--
  owa_util.redirect_url( owa_util.get_owa_service_path ||
                         'pits_issue_upd.display_issues' ||
                         '?project_identifier=' || this_project_identifier ||
                         '&issue_category=' || this_issue_category ||
                         '&issue_status=' || this_issue_status, TRUE );
--  owa_util.redirect_url( owa_util.get_owa_service_path ||
--                        'pits_issue_upd.main', TRUE );
--
EXCEPTION
  when others then
    pits_utility.report_error( 'pits_issue_upd.close_issue' );
--
END close_issue;
--
/*****************************************************************************/
--
PROCEDURE delete_verification( issue_rowid in varchar2 default null )
/*
 *   Ask the user to verify that they really want to delete the
```

```
 *   selected issue.
 *
 *    Input:  issue_rowid
 *
 *   Output:  database data
 *
 */
IS
  cursor issue_cursor is
         select rowid, project_identifier, issue_id,
                external_identifier, module_name, issue_category,
                to_char(issue_date,'mm/dd/yyyy') issue_date,
                to_char(due_date,'mm/dd/yyyy') due_date,
                severity, priority, issue_status, user_contact, it_contact,
                issue_description
           from pits_issues
           where rowid = delete_verification.issue_rowid;
--
  cursor issue_activity_cursor( cursor_issue_id number ) is
         select rowid, to_char(activity_date,'mm/dd/yyyy') activity_date,
                activity_person, activity_description
           from pits_issue_activity
           where issue_id = cursor_issue_id
           order by activity_date desc;
--
  issue_activity_count number := 0;
BEGIN
--
--  set up the HTML page
--
  pits_utility.page_header( 'ISSUE_DELETE_VERIFICATION', 'Issue Delete Verification' );
--
  htp.nl;
--
  htp.tableOpen( cattributes=>'WIDTH=600 BORDER=0' );
--
  htp.formOpen( owa_util.get_owa_service_path || 'pits_issue_upd.delete_issue' );
  htp.formHidden( 'issue_rowid', delete_verification.issue_rowid );
--
  for issue_record in issue_cursor
  loop
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Project:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.project_identifier, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Issue ID:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( to_char(issue_record.issue_id), ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Category:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.issue_category, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Module:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.module_name, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Date:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.issue_date, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Due Date:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.due_date, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
```

```
    htp.tableData( htf.bold('Severity:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.severity, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Priority:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.priority, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('User Contact:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.user_contact, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('IT Contact:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.it_contact, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('External Identifier:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( issue_record.external_identifier, ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.hr( cattributes=>'WIDTH=600' ), ccolspan=>'4' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.bold('Issue:'), 'right', cattributes=>'WIDTH = 125' );
    htp.tableData( replace(issue_record.issue_description,chr(13),'<br>'), ccolspan=>'3' );
    htp.tableRowClose;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.hr( cattributes=>'WIDTH=600' ), ccolspan=>'4' );
    htp.tableRowClose;
--
    for issue_activity_record in issue_activity_cursor( issue_record.issue_id )
    loop
      issue_activity_count := issue_activity_count + 1;
      if issue_activity_count = 1 then
        htp.tableRowOpen( 'left', 'top' );
        htp.tableData( '' );
        htp.tableData( htf.bold('Date') );
        htp.tableData( htf.bold('Who') );
        htp.tableData( htf.bold('Activity') );
        htp.tableRowClose;
      end if;
--
      htp.tableRowOpen( 'left', 'top' );
      htp.tableData( '' );
      htp.tableData( issue_activity_record.activity_date );
      htp.tableData( issue_activity_record.activity_person );
      htp.tableData( replace(issue_activity_record.activity_description,chr(13),'<br>') );
      htp.tableRowClose;
    end loop;
--
  end loop;
--
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( htf.formSubmit( null, 'Delete' ), ccolspan=>4 );
  htp.tableRowClose;
--
  htp.formClose;
--
  for issue_record in
    ( select project_identifier, issue_category, issue_status
      from pits_issues
      where rowid = delete_verification.issue_rowid )
  loop
    htp.formOpen( owa_util.get_owa_service_path || 'pits_issue_upd.display_issues' );
    htp.formHidden( 'project_identifier', issue_record.project_identifier );
    htp.formHidden( 'issue_category', issue_record.issue_category );
```

```
    htp.formHidden( 'issue_status', issue_record.issue_status );
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( htf.formSubmit( null, 'Cancel' ), ccolspan=>4 );
    htp.tableRowClose;
--
    htp.formClose;
  end loop;
--
  htp.tableClose;
--
  pits_utility.page_footer( 'ISSUE_DELETE_VERIFICATION', pits_issue_upd.version );
--
END delete_verification;
--
/*****************************************************************************/
--
PROCEDURE delete_issue( issue_rowid in varchar2 default null )
/*
 *   Delete a issue record and associated issue_activity records.
 *
 *    Input:  issue_rowid
 *
 *   Output:  database data
 *
 */
IS
--
  this_project_identifier pits_issues.project_identifier%type := null;
  this_issue_category     pits_issues.issue_category%type := null;
  this_issue_status       pits_issues.issue_status%type := null;
--
BEGIN
--
  for issue_record in
    ( select project_identifier, issue_category, issue_status
        from pits_issues
        where rowid = delete_issue.issue_rowid )
  loop
    this_project_identifier := issue_record.project_identifier;
    this_issue_category := issue_record.issue_category;
    this_issue_status := issue_record.issue_status;
  end loop;
--
  begin
    delete pits_issue_activity
      where issue_id = (select issue_id
                          from pits_issues
                          where rowid = delete_issue.issue_rowid);
  exception
    when others then
      pits_utility.report_error( 'pits_issue_upd.delete_issue issue_activity' );
  end;
--
  begin
    delete pits_issues
      where rowid = delete_issue.issue_rowid;
  exception
    when others then
      pits_utility.report_error( 'pits_issue_upd.delete_issue issue' );
  end;
--
  owa_util.redirect_url( owa_util.get_owa_service_path ||
                           'pits_issue_upd.display_issues' ||
                           '?project_identifier=' || this_project_identifier ||
                           '&issue_category=' || this_issue_category ||
                           '&issue_status=' || this_issue_status, TRUE );
--  owa_util.redirect_url( owa_util.get_owa_service_path ||
--                          'pits_issue_upd.main', TRUE );
--
EXCEPTION
  when others then
    pits_utility.report_error( 'pits_issue_upd.delete_issue' );
```

```
--
END delete_issue;
--
/*****************************************************************************/
/*****************************************************************************/
--
END pits_issue_upd;
/
show errors
```

Package Example 2

```
/*****************************************************************************/
--
PROCEDURE current_sessions
/*
 *   Display the available SIDs for display of the current sessions.
 *
 *    Input:  (none)
 *
 *   Output:  (none)
 */
IS
--
  anchor varchar2(200);
--
BEGIN
--
--  set up the HTML page
--
  pat_utility.page_header( 'CURRENT_SESSIONS', 'Display', 'Current Sessions' );
--
  htp.tableOpen( cattributes=>'WIDTH = 600 NOBORDER' );
--
  htp.tableRowOpen;
  htp.tableData( ' ', cattributes=>'WIDTH=50' );
  htp.tableData( ' ', cattributes=>'WIDTH=50' );
  htp.tableRowClose;
--
  for validation_record in
    ( select validation_text
        from pat_validations
    where validation_field = 'INSTANCE_SID'
    order by validation_text )
  loop
    anchor := owa_util.get_owa_service_path || 'pat_reports.display_sessions?sid=' ||
              validation_record.validation_text;
    htp.tableRowOpen;
    htp.tableData( htf.img( curl=>'/pat_images/pat_menu_item.gif', calign=>'RIGHT',
                            calt=>'' ), cattributes=>'WIDTH=100' );
    htp.tableData( htf.anchor( anchor, validation_record.validation_text ) );
    htp.tableRowClose;
--
    htp.tableRowOpen;
    htp.tableRowClose;
--
  end loop;
--
  htp.TableClose;
--
  htp.nl;
--
  pat_utility.page_footer( 'CURRENT_SESSIONS', 'DISPLAY', pat_reports.version );
--
EXCEPTION
  when others then
    pat_utility.report_error( 'pat_reports.current_sessions' );
--
END current_sessions;
--
/*****************************************************************************/
```

```
--
PROCEDURE display_sessions( sid in varchar2 default null )
/*
 *   Show all database sessions/processes.
 *
 *    Input:  sid
 *
 *   Output:  (none)
 */
IS
--
  process_count NUMBER := 0;
  session_count NUMBER := 0;
  user_session_count NUMBER := 0;
  this_username varchar2(30) := null;
--
  type  ref_cursor_type is REF CURSOR;
  process_cursor           ref_cursor_type;
  process_query            varchar2(2000) := null;
  process_spid             v$process.spid%type;
  process_pid              v$process.pid%type;
  process_process          v$session.process%type;
  process_addr             v$process.addr%type;
  user_cursor              ref_cursor_type;
  user_query               varchar2(2000) := null;
  user_user_name           apps.fnd_user.user_name%type;
  session_cursor           ref_cursor_type;
  session_query            varchar2(2000) := null;
  session_sid              v$session.sid%type;
  session_status           v$session.status%type;
  session_serial_number    v$session.serial#%type;
  session_program          v$session.program%type;
  session_process          v$session.process%type;
  session_osuser           v$session.osuser%type;
  session_logon_time       varchar2(22);
--
  pat_sid pat_parameters.parameter_text%type := null;
--
BEGIN
--
  pat_sid := pat_utility.read_text_parameter( 'PAT_SID' );
--
  if display_sessions.sid = pat_sid then
    process_query := 'select distinct p.spid, p.pid, s.process, p.addr ' ||
                     'from v$process p, v$session s ' ||
                     'where p.addr = s.paddr ' ||
                     'order by to_number(p.spid)';
  else
    process_query := 'select distinct p.spid, p.pid, s.process, p.addr ' ||
                     'from v$process@' || display_sessions.sid ||
                     ' p, v$session@' || display_sessions.sid ||' s ' ||
                     'where p.addr = s.paddr ' ||
                     'order by to_number(p.spid)';
  end if;
--
--  set up the HTML page
--
  pat_utility.page_header( 'display_sessions', 'Display', 'Current Sessions' );
--
  htp.tableOpen( cattributes=>'WIDTH = 600 BORDER' );
--
  open process_cursor for process_query;
  loop
    fetch process_cursor
      into process_spid,
           process_pid,
           process_process,
           process_addr;
    exit when process_cursor%notfound;
--
    if trunc(process_count/50)*50 = process_count then
      htp.tableRowOpen;
      htp.tableData( htf.bold('OS PID') );
```

```
      htp.tableData( htf.bold('Oracle PID') );
      htp.tableData( htf.bold('User') );
      htp.tableData( htf.bold(' ') );
      htp.tableRowClose;
    end if;
    process_count := process_count + 1;
--
    htp.tableRowOpen( 'left', 'top' );
    htp.tableData( process_spid );
    htp.tableData( process_pid );
    this_username := null;
    if display_sessions.sid = pat_sid then
      user_query := 'select u.user_name ' ||
                    'from apps.fnd_logins l,apps.fnd_user u ' ||
                    'where l.pid = ''' || process_pid || ''' ' ||
                    'and l.spid = ''' || process_process || ''' ' ||
                    'and l.end_time is null ' ||
                    'and l.user_id = u.user_id';
    else
      user_query := 'select u.user_name ' ||
                    'from apps.fnd_logins@' || display_sessions.sid ||
                    ' l, apps.fnd_user@' || display_sessions.sid || ' u ' ||
                    'where l.pid = ''' || process_pid || ''' ' ||
                    'and l.spid = ''' || process_process || ''' ' ||
                    'and l.end_time is null ' ||
                    'and l.user_id = u.user_id';
    end if;
    open user_cursor for user_query;
    loop
      fetch user_cursor
        into user_user_name;
      this_username := user_user_name;
      exit when user_cursor%notfound;
    end loop;
    close user_cursor;
    htp.tableData( nvl(this_username,' ') );
    htp.print( '<td>' );
    htp.tableOpen( cattributes=>'BORDER="1"' );
    if display_sessions.sid = pat_sid then
      session_query := 'select s.sid, s.status, s.serial# serial_number,s.program, ' ||
                       's.process, s.osuser, ' ||
                       'to_char(s.logon_time,''mm/dd/yyyy hh24:mi'') logon_time ' ||
                       'from v$session s ' ||
                       'where s.paddr = ''' || process_addr || ''' ' ||
                       'order by sid, s.serial#';
    else
      session_query := 'select s.sid, s.status, s.serial# serial_number,s.program, ' ||
                           's.process, s.osuser, ' ||
                           'to_char(s.logon_time,''mm/dd/yyyy hh24:mi'') logon_time ' ||
                           'from v$session@' || display_sessions.sid || ' s ' ||
                           'where s.paddr = ''' || process_addr || ''' ' ||
                           'order by sid, s.serial#';
    end if;
    session_count := 0;
    open session_cursor for session_query;
    loop
      fetch session_cursor
        into session_sid,
             session_status,
             session_serial_number,
             session_program,
             session_process,
             session_osuser,
             session_logon_time;
      exit when session_cursor%notfound;
--
      session_count := session_count + 1;
      user_session_count := user_session_count + 1;
      if session_count = 1 then
        htp.tableRowOpen( 'left', 'top' );
        htp.tableData( htf.bold( 'SID' ) );
        htp.tableData( htf.bold( 'Serial' ) );
        htp.tableData( htf.bold( 'Status' ) );
```

```
        htp.tableData( htf.bold( 'OS User' ) );
        htp.tableData( htf.bold( 'Logon' ) );
        htp.tableData( htf.bold( 'Program' ) );
        htp.tableRowClose;
      end if;
      htp.tableRowOpen( 'left', 'top' );
      htp.tableData( session_sid );
      htp.tableData( session_serial_number );
      htp.tableData( session_status );
      htp.tableData( session_osuser );
      htp.tableData( session_logon_time );
      htp.tableData( session_program );
      htp.tableRowClose;
--
    end loop;
    close session_cursor;
--
    htp.tableClose;
    htp.tableRowClose;
    htp.print( '</td>' );
--
  end loop;
  close process_cursor;
--
  htp.tableRowOpen( 'left', 'top' );
  htp.tableData( 'Process Count: ' || to_char(process_count) );
  htp.tableData( 'Session Count: ' || to_char(user_session_count) );
  htp.tableRowClose;
--
  pat_utility.page_footer( 'display_sessions', 'DISPLAY', pat_reports.version );
--
EXCEPTION
  when others then
    pat_utility.report_error( 'pat_reports.display_sessions' );
--
END display_sessions;
--
--
```