```
--
--  ask_utility.sql
--
set scan off
--
CREATE or REPLACE PACKAGE ask_utility IS
/*
 *   Package:  ask_utility
 *
 *    Module:  ask_utility.sql
 *
 *   Purpose:  Ask! - Survey Tool
 *             Utility function and procedure headers.
 *
 *   1. 29 Jun 2006  Jeunnette  Initial implementation.
 *
 ************************************************************************
 *  Copyright, (C) 2006 by Prairie Systems Group, Limited               *
 ************************************************************************
 */
--
--  package global data
--
  ask_version             constant  varchar2(3) := '1';
  version                 constant  varchar2(3) := '1';
--
  table_head_bgcolor       CONSTANT   varchar2(10) := 'white';
  table_even_line_bgcolor  CONSTANT   varchar2(10) := 'aliceblue';
  table_odd_line_bgcolor   CONSTANT   varchar2(10) := '#DDDDDD'; -- light gray
--
  page_background         constant  varchar2(50) := 'ask_background.jpg';
  datetime_format         constant  varchar2(20) := 'mm/dd/yyyy hh24mi';
  date_format             constant  varchar2(20) := 'mm/dd/yyyy';
--
/* Page Heading Procedure ****************************************/
  PROCEDURE page_header( page_name             IN VARCHAR2 DEFAULT 'DEFAULT' ,
                         page_title            in varchar2 default null,
                         session_id            in number default 0,
                         user_id               in number default 0,
                         organization_id       in number default 0,
                         main_menu_button_flag in boolean default FALSE,
                         about_button_flag     in boolean default FALSE,
                         close_button_flag     in boolean default FALSE,
                         print_button_flag     in boolean default FALSE,
                         focus_field           in varchar2 default null );
--
/* Page Footer Procedure *****************************************/
  PROCEDURE page_footer( page_name        IN VARCHAR2 DEFAULT NULL ,
                         page_version     IN VARCHAR2 DEFAULT null,
                         organization_id  in number default 0 );
```

```
--
/* Local URL Function ************************************/
  FUNCTION local_URL( package_procedure_parameters in varchar2 )
           RETURN VARCHAR2;
--
/* Standard Error Reporting Procedure ******************************/
  PROCEDURE report_error( organization_id     in number default 0,
                          routine_name IN VARCHAR2 DEFAULT NULL,
                          context in varchar2 default null );
--
/* Validation Error Display Procedure ******************************/
  PROCEDURE display_validation_message( validation_message IN VARCHAR2 DEFAULT NULL );
--
/* Display Help Procedure ************************************************/
  PROCEDURE display_help( help_text_id in number default 0 );
--
/* Read Parameter Function *********************************/
  FUNCTION read_parameter( organization_id  in number,
                           parameter in varchar2 )
           RETURN VARCHAR2;
--
/* Check Parameter Flag Function ***********************************/
  FUNCTION check_parameter_flag( organization_id in number,
                                 parameter in varchar2 )
           RETURN BOOLEAN;
--
/* Is_Number Function ******************************************/
  FUNCTION is_number( numeric_string in varchar2 )
           RETURN boolean;
--
/* ValidateDate Function ***************************************/
  FUNCTION ValidateDate( in_date in varchar2,
                         out_date out varchar2 )
           RETURN BOOLEAN;
--
/* Expand_Text Function ****************************************/
  FUNCTION expand_text( text_string in varchar2 )
           RETURN varchar2;
--
/* Trim_string Function ****************************************/
  FUNCTION trim_string( in_string in varchar2 )
           RETURN VARCHAR2;
--
/* next_sequence Function  *************************************/
  FUNCTION next_sequence
           RETURN number;
--
/* send_mail Procedure *****************************************/
  PROCEDURE send_mail( from_address in varchar2,
                       to_address   in varchar2,
```

```
                      cc_address   in varchar2,
                      subject      in varchar2,
                      message_text in varchar2 default null );
--
/* Encrypt Function ************************************************/
  FUNCTION dbms_encrypt( input_string in varchar2 )
           RETURN VARCHAR2;
--
/* Decrypt Function ************************************************/
  FUNCTION dbms_decrypt( input_string in varchar2 )
           RETURN VARCHAR2;
--
END;
/
show errors
--
CREATE or REPLACE PACKAGE BODY ask_utility IS
/*
 *  Package:  ask_utility
 *
 *  Purpose:  Survey Tool utility procedures/functions
 *
 ********************************************************************
 *  Copyright, (C) 2006 by Prairie Systems Group, Limited           *
 ********************************************************************
 */
/*****************************************************************************/
/*--- Private Functions/Procedures ------------------------------------------*/
/* style        - produces the style page                                    */
/* javascript   - some standard javascript functions                         */
/*---------------------------------------------------------------------------*/
--
  PROCEDURE style( organization_id in number default 0 )
  IS
  --
    css_text ask_organization_parameters.parameter_text%type := null;
  --
  BEGIN
    css_text := ask_utility.read_parameter( style.organization_id, 'STYLE_SHEET' );
--
    if css_text is not null then
      htp.print( css_text );
    else
      htp.print( '<STYLE TYPE="text/css">' );
      htp.print( '<!-- ' );
      htp.print( 'A img { border: 0px none; }' );
      htp.print( 'B { font-size: 10pt; font-family: Verdana,Arial,Helvetica,sans-serif }' );
      htp.print( 'BODY {  background: white; color: black; font-size: 10pt; font-family:
      Verdana,Arial,Helvetica,sans-serif }' );
      htp.print( 'CAPTION { font-size: 16pt; color: blue; font-family: Verdana,Arial,Helvetica,sans-serif
```

```
      }' );
      htp.print( 'H1 { font-family: Georgia, Palatino, Times New Roman, Times, serif; font-size: 170%;
      color: #336699; border: solid #CCCC99; width: 100%; border-width: 0px 0px 2px 0px; }' );
      htp.print( 'H2 { font-family: Georgia, Palatino, Times New Roman, Times, serif; font-size: 130%;
      color: #336699; border: solid #CCCC99; width: 100%; border-width: 0px 0px 2px 0px; }' );
      htp.print( 'H3 { font-family: Georgia, Palatino, Times New Roman, Times, serif; font-size: 110%;
      color: #336699; border: solid #CCCC99; width: 100%; border-width: 0px 0px 2px 0px; }' );
      htp.print( 'H4 { font-family: Georgia, Palatino, Times New Roman, Times, serif; font-size: 92% }' );
      htp.print( 'H5 { font-size: 12pt; font-family: Georgia, Palatino, Times New Roman, Times, serif }' );
      htp.print( 'H6 { font-size: 10pt; font-family: Georgia, Palatino, Times New Roman, Times, serif }' );
      htp.print( 'TD { color: black; font-size: 9pt; font-family: Verdana,Arial,Helvetica,sans-serif }' );
--      htp.print( 'TH { color: black; background: #CCCCCC; font-size: 10pt; font-family:
Verdana,Arial,Helvetica,sans-serif }' );
--      htp.print( 'TH { color: white; background: #2F5378; font-size: 10pt; font-family: Georgia,
Palatino, Times New Roman, Times, serif }' );
      htp.print( 'TH { color: white; background: #2F5378; font-size: 10pt; font-family:
      Verdana,Arial,Helvetica,sans-serif }' );
      htp.print( 'TD { font-size: 10pt; font-family: Verdana,Arial,Helvetica,sans-serif }' );
      htp.print( 'INPUT { font-size: 10pt; font-family: Verdana,Arial,Helvetica,sans-serif }' );
      htp.print( 'OPTION { font-size: 10pt; font-family: Verdana,Arial,Helvetica,sans-serif }' );
      htp.print( 'SELECT { font-size: 10pt; font-family: Verdana,Arial,Helvetica,sans-serif }' );
      htp.print( 'TEXTAREA { font-size: 10pt; font-family: Verdana,Arial,Helvetica,sans-serif }' );
      htp.print( '.aliceblue { background: aliceblue } ' );
      htp.print( '.blue { color: blue } ' );
      htp.print( '.calendar_even_cell { background: #E8E8E8; font-size: 8pt; border-left:1px dotted #666;
      border-top:1px dotted #666; border-right:1px dotted #666; border-bottom:1px dotted #666; width:
      100px; height: 25px; vertical-align: top; horizontal-align: left }' );
      htp.print( '.calendar_odd_cell { background: #white; font-size: 8pt; border-left:1px dotted #666;
      border-top:1px dotted #666; border-right:1px dotted #666; border-bottom:1px dotted #666; width:
      100px; height: 25px; vertical-align: top; horizontal-align: left }' );
      htp.print( '.dataentry { background: #DDDDDD; font-size: 10pt }' );
      htp.print( '.eighteen { font-size: 16pt; font-family: Verdana,Arial,Helvetica,sans-serif }' );
      htp.print( '.fourteen { font-size: 14pt }' );
      htp.print( '.green { font-size: 8pt; color: green }' );
      htp.print( '.instructions { color: blue; font: italic 10pt sans-serif }' );
      htp.print( '.lightcoral { background: lightcoral } ' );
      htp.print( '.lightgreen { background: lightgreen } ' );
      htp.print( '.lightseagreen { background: lightseagreen } ' );
      htp.print( '.light_blue { background: #D0E0F0 } ');
      htp.print( '.light_gray { background: #DDDDDD } ');
      htp.print( '.nine { font-size: 8pt }' );
      htp.print( '.optional { background: #DDDDDD; font-size: 8pt }' );
      htp.print( '.palegreen { background: palegreen } ');
      htp.print( '.red { font-size: 8pt; color: red }' );
      htp.print( '.required { background: #D0E0FF; font-size: 8pt }' );
      htp.print( '.sixteen { font-size: 14pt; font-family: Verdana,Arial,Helvetica,sans-serif }' );
      htp.print( '.small { font-size: 8pt } ');
      htp.print( '.springgreen { background: springgreen } ' );
      htp.print( '.table_even_line { background: #E8E8E8; } ' );
      htp.print( '.table_odd_line { background: white; } ' );
```

```
      htp.print( '.ten { font-size: 8pt }' );
      htp.print( '.tiny { font-size: 6pt }' );
      htp.print( '.twelve { font-size: 10pt }' );
      htp.print( '.ten { font-size: 8pt }' );
      htp.print( '.text { font-family: Courier }' );
      htp.print( '.white { background: white } ');
      htp.print( ' -->');
      htp.print( '</STYLE>' );
    end if;
  END style;
--
  PROCEDURE javascript
  IS
  BEGIN
    htp.print( '<SCRIPT LANGUAGE="JavaScript">' );
    htp.print( 'function upperCase(this_field) { var tfValue = this_field.value; this_field.Value =
    tfValue.toUpperCase(); }' );
    htp.print( 'function setFocus(this_form, this_field) { document.this_form.this_field.focus(); }' );
    htp.print( 'function popWindow( pURL, name, features ) { new_window = window.open( pURL, name, features
    ); }' );
    htp.print( '</SCRIPT>' );
  END javascript;
--
/*--- Public Functions/Procedures ------------------------------------------------------*/
/*--- Public Functions/Procedures ------------------------------------------------------*/
--
/*****************************************************************************************/
--
  PROCEDURE page_header( page_name             in VARCHAR2 default 'DEFAULT' ,
                         page_title            in varchar2 default null,
                         session_id            in number default 0,
                         user_id               in number default 0,
                         organization_id       in number default 0,
                         main_menu_button_flag in boolean default FALSE,
                         about_button_flag     in boolean default FALSE,
                         close_button_flag     in boolean default FALSE,
                         print_button_flag     in boolean default FALSE,
                         focus_field           in varchar2 default null )
  IS
    url         VARCHAR2(100);
    ip varchar2(20) := owa_util.get_cgi_env('REMOTE_ADDR');
--
    organization_background   ask_organization_parameters.parameter_text%type := null;
    organization_page_header  ask_organization_parameters.parameter_text%type := null;
--
  BEGIN
    if page_header.organization_id != 0 then
      organization_background := ask_utility.read_parameter( page_header.organization_id,
                                                             'PAGE_BACKGROUND' );
      organization_page_header := ask_utility.read_parameter( page_header.organization_id,
```

```
                                                              'PAGE_HEADER' );
    end if;
--
    htp.print( '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">' );
    htp.htmlOpen;
    htp.HeadOpen;
    htp.Title( 'Ask! Survey Tool - '|| page_header.page_title );
    style( page_header.organization_id );
    htp.HeadClose;
--
    if organization_background is not null then
      if substr(organization_background,1,1) = '#' then
        if focus_field is not null then
          htp.BodyOpen(cattributes=>'BACKGROUND="' || organization_background || '" ' ||
                                    'onLoad="document.' || focus_field || '.focus()"' );
        else
          htp.BodyOpen(cattributes=>'BACKGROUND="' || organization_background || '"' );
        end if;
      else
        if focus_field is not null then
          htp.BodyOpen(cattributes=>'BACKGROUND="' || owa_util.get_owa_service_path || 'ask_images/' ||
          organization_background || '" ' ||
                                    'onLoad="document.' || focus_field || '.focus()"' );
        else
          htp.BodyOpen(cattributes=>'BACKGROUND="' || owa_util.get_owa_service_path || 'ask_images/' ||
          organization_background || '"' );
        end if;
      end if;
    else
      if focus_field is not null then
        htp.BodyOpen(cattributes=>'BGCOLOR="#ffffff" onLoad="document.' || focus_field || '.focus()"' );
      else
        htp.BodyOpen(cattributes=>'BGCOLOR="#ffffff"');
      end if;
    end if;
--
    htp.tableOpen( calign=>'LEFT', cclear=>'CLEAR', cattributes=>'WIDTH=600 BORDER=0' );
--
    htp.tableRowOpen( 'left', 'top' );
    htp.print( '<td>' );
    htp.tableOpen( cattributes=>'WIDTH=600 BORDER=0' );
--
    htp.tableRowOpen( calign=>'LEFT' );
    if organization_page_header is not null then
      htp.tableData( organization_page_header );
    else
      htp.tableData( htf.img( owa_util.get_owa_service_path || 'ask_images/ask_page_header.jpg',
      calt=>'Header' ) );
    end if;
    htp.tableRowClose;
```

```
--
    htp.tableRowOpen;
    htp.tableData( htf.header( nsize=>3, calign=>'LEFT',
                               cheader=>ask_utility.expand_text(page_header.page_title) ) );
    htp.tableRowClose;
--
    htp.tableClose;
    htp.print( '</td>' );
    htp.tableRowClose;
--
    if page_header.main_menu_button_flag or
       page_header.about_button_flag or
       page_header.print_button_flag or
       page_header.close_button_flag or
       page_header.session_id != 0 then
      htp.tableRowOpen( calign=>'center' );
      htp.print( '<td>' );
      htp.tableOpen( calign=>'CENTER', cclear=>'CLEAR', cattributes=>'NOBORDER WIDTH=600
      CLASS="light_gray"' );
      htp.tableRowOpen( 'top', 'center', cattributes=>'CLASS="light_gray"' );
      if main_menu_button_flag then
        if page_header.session_id != 0 and page_header.user_id != 0 then
          htp.tableData( htf.anchor2( curl=>owa_util.get_owa_service_path || 'ask.menu' ||
                                           '?session_id=' || to_char(page_header.session_id) ||
                                           '&user_id=' || to_char(page_header.user_id),
                                      ctext=>'<img src="' || owa_util.get_owa_service_path ||
                                             'ask_images/ask_main_menu_button.jpg' ||
                                             '" border="0" alt="Main Menu">' ),
                         'left' );
        end if;
      end if;
--
      if page_header.about_button_flag then
        if page_header.session_id != 0 or page_header.user_id != 0 then
          htp.tableData( htf.anchor2( curl=>owa_util.get_owa_service_path || 'ask.about',
                                      ctext=>'<img src="' || owa_util.get_owa_service_path ||
                                             'ask_images/ask_about_button.jpg' ||
                                             '" border="0" alt="About">',
                                      ctarget=>'about' ),
                         'center' );
        else
          htp.tableData( htf.anchor2( curl=>owa_util.get_owa_service_path || 'ask.about',
                                      ctext=>'<img src="' || owa_util.get_owa_service_path ||
                                             'ask_images/ask_about_button.jpg' ||
                                             '" border="0" alt="About">',
                                      ctarget=>'about' ),
                         'left' );
        end if;
      end if;
--
```

```
      if page_header.print_button_flag then
        htp.tableData( htf.anchor2( curl=>'javascript:window.print()',
                                    ctext=>'<img src="' || owa_util.get_owa_service_path ||
                                           'ask_images/ask_print_button.jpg' ||
                                           '" border="0" alt="Print this page.">' ),
                       'right' );
      end if;
--
      if page_header.close_button_flag then
        htp.tableData( htf.anchor2( curl=>'javascript:window.close()',
                                    ctext=>'<img src="' || owa_util.get_owa_service_path ||
                                           'ask_images/ask_close_button.jpg' ||
                                           '" border="0" alt="Close this window.">' ),
                       'right' );
      end if;
--
      for help_text_record in
        ( select help_text_id
            from ask_help_text
            where help_identifier = upper(page_header.page_name) )
      loop
        htp.print( '<SCRIPT LANGUAGE="JavaScript">' );
        htp.print( 'function popWindow( pURL, name, features ) { new_window = window.open( pURL, name,
        features ); new_window.focus(); }' );
        htp.print( '</SCRIPT>' );
--
        htp.tableData( htf.anchor2( curl=>replace(htf.escape_url(owa_util.get_owa_service_path ||
                                                  'ask_utility.display_help' ||
                                                  '?help_text_id=' ||
                                                  to_char(help_text_record.help_text_id)),
                                                  ' ','%20'),
                                    ctext=>'<img src="' || owa_util.get_owa_service_path ||
                                           'ask_images/ask_help_button.jpg" border="0" alt="Help">',
                                    ctarget=>'Help',
                                    cattributes=>'onClick="popWindow(''' ||
                                                 replace(htf.escape_url(owa_util.get_owa_service_path ||
                                                      'ask_utility.display_help' ||
                                                      '?help_text_id=' ||
                                                      to_char(help_text_record.help_text_id)),
                                                      ' ','%20') ||
                                                ''', ''Help'', ''width=650,height=600,left=300,top=
                                                0,resizable,scrollbars'')"' ),
                       'center' );
      end loop;
--
      if page_header.session_id != 0 then
        htp.tableData( htf.anchor2( curl=>owa_util.get_owa_service_path || 'ask.logout' ||
                                         '?session_id=' || to_char(page_header.session_id),
                                    ctext=>'<img src="' || owa_util.get_owa_service_path ||
                                           'ask_images/ask_logout_button.jpg' ||
```

```
                                              '" border="0" alt="Logout">' ),
                        'right' );
      end if;
      htp.tableRowClose;
--
      htp.tableClose;
      htp.print( '</td>' );
      htp.tableRowClose;
--
    end if;
--
    htp.tableClose;
--
    htp.nl( cclear=>'LEFT' );
--
  EXCEPTION
    when others then
      ask_utility.report_error( page_header.organization_id,
                                'ask_utility.page_header', 'procedure' );
--
  END page_header;
--
/*******************************************************************************************/
--
  PROCEDURE page_footer( page_name        in VARCHAR2 DEFAULT NULL ,
                         page_version     IN VARCHAR2 DEFAULT null,
                         organization_id  in number default 0 )
  IS
    user_name     VARCHAR2(30);
    system_date   VARCHAR2(20);
--
    organization_page_footer  ask_organization_parameters.parameter_text%type := null;
--
  BEGIN
    if page_footer.organization_id != 0 then
      organization_page_footer := ask_utility.read_parameter( page_footer.organization_id,
                                                              'PAGE_FOOTER' );
    end if;
--
    htp.nl( cclear=>'LEFT' );
--
    htp.tableOpen( cattributes=>'WIDTH=600 CELLPADDING=0 CELLSPACING=0 BORDER=0' );
--
    if organization_page_footer is not null then
      htp.tableRowOpen;
      htp.tableData( organization_page_footer );
      htp.tableRowClose;
    end if;
--
    htp.tableRowOpen;
```

```
--    htp.tableData( htf.img2( curl=> owa_util.get_owa_service_path || 'ask_images/ask_curve.gif',
--                             cattributes=>'WIDTH=600 HEIGHT=17' ) );
    htp.tableData( htf.img2( curl=> owa_util.get_owa_service_path || 'ask_images/ask_psg_footer.jpg' ) );
    htp.tableRowClose;
--
    htp.tableRowOpen;
    htp.tableData( htf.bold('Ask! Survey Tool', cattributes=>'CLASS="SMALL"') ||
                  ' | Copyright &copy; ' || to_char(sysdate,'yyyy') ||
                  ', Prairie Systems Group, Limited  ',
                  'right', cattributes=>'WIDTH=600 CLASS="SMALL"' );
    htp.tableRowClose;
--
    htp.tableRowOpen;
    htp.tableData( 'Page Name: ' || page_name ||
                   ' Version: ' || ask_utility.ask_version || '.' || page_version || '  ',
                   'right', cattributes=>'WIDTH=600 CLASS="SMALL"' );
    htp.tableRowClose;
--
    htp.tableClose;
--
    htp.BodyClose;
    htp.HtmlClose;
--
  EXCEPTION
    when others then
      ask_utility.report_error( page_footer.organization_id,
                                'ask_utility.page_footer', 'procedure' );
--
  END page_footer;
--
/*****************************************************************************************/
--
FUNCTION local_URL( package_procedure_parameters in varchar2 )
         RETURN VARCHAR2
/*
 *   Create a "local" URL .
 *
 *    Input:  package_procedure_parameters
 *
 *   Output:  URL
 *
 */
IS
--
BEGIN
  return owa_util.get_owa_service_path || local_URL.package_procedure_parameters;
--
EXCEPTION
  when others then
    return null;
```

```
--
END local_URL;
--
/*****************************************************************************/
--
PROCEDURE report_error( organization_id    in number default 0,
                        routine_name       in varchar2 default null,
                        context            in varchar2 default null )
/*
 *   Let the user know about an unexpected Oracle error.
 *
 *    Input:  organization_id
 *            routine_name
 *            context - additional information about what the routine was
 *                      askng.
 *
 *   Output:  none
 *
 */
IS
--
  this_error_code  number := SQLCODE;
  this_error_message  varchar2(300) := SQLERRM;
--
BEGIN
--
  ask_utility.page_header( page_name => 'ERROR_REPORT',
                           print_button_flag => TRUE,
                           organization_id => report_error.organization_id );
--
  htp.nl;
  htp.print( htf.bold( 'Unexpected error in routine ' || routine_name ) );
--
  if context is not null then
    htp.nl;
    htp.nl;
    htp.print( context );
  end if;
--
  if sqlerrm is not null then
    htp.nl;
    htp.nl;
    htp.print( 'Oracle error: ' || this_error_message );
  end if;
--
  htp.nl;
  htp.nl;
  htp.print( '<kbd>' ||
             htf.nobr(replace(dbms_utility.format_call_stack,chr(10),'<br>')) ||
             '</kbd>' );
```

```
--
  htp.nl;
  htp.line( cattributes=>'WIDTH=600 ALIGN=LEFT' );
--
--  htp.formOpen( owa_util.get_owa_service_path || 'ask.main' );
--  htp.formSubmit( null, 'Continue' );
--  htp.formClose;
--
  ask_utility.page_footer( 'ERROR_REPORT',
                           organization_id => report_error.organization_id );
--
  begin
    insert into ask_errors
      ( error_id,
        organization_id,
        error_timestamp,
        error_message,
        routine_name,
        error_context )
      values ( ask_utility.next_sequence,
               report_error.organization_id,
               sysdate,
               this_error_message,
               substr(routine_name,1,200),
               substr(context,1,4000) );
  exception
    when others then
      null;
  end;
--
END report_error;
--
/*****************************************************************************/
--
PROCEDURE display_validation_message( validation_message in varchar2 default null )
/*
 *   Let the user know about a validation message.  This code is centralized
 *   in ask_utility so we change what it looks like without having to change
 *   all the places it is referenced.
 *
 *    Input:  validation_message
 *
 *   Output:  HTML
 *
 */
IS
--
BEGIN
--
  htp.nl;
```

```
  htp.tableOpen( cattributes=>'WIDTH=600' );
  htp.tableRowOpen;
  htp.tableData( htf.img( curl=>owa_util.get_owa_service_path || 'ask_images/ask_error.gif',
  calign=>'RIGHT',
                          calt=>'Validation Error(s):' ), cattributes=>'WIDTH = 125' );
  htp.tableData( validation_message, cattributes=>'CLASS="instructions"' );
  htp.tableRowClose;
  htp.tableClose;
  htp.nl;
  htp.line( cattributes=>'WIDTH=600 ALIGN=LEFT' );
--
END display_validation_message;
--
/*****************************************************************************/
--
PROCEDURE display_help( help_text_id in number default 0 )
/*
 *   Display a specified help screen.
 *
 *   Input:  help_text_id
 *
 */
IS
--
  help_record_count number := 0;
--
BEGIN
--
  ask_utility.page_header( 'DISPLAY_HELP', 'Help' );
--
  htp.nl;
  htp.tableOpen( cattributes=>'WIDTH=600 NOBORDER' );
--
  for help_text_record in
    ( select help_identifier, help_text
        from ask_help_text
        where help_text_id = display_help.help_text_id )
  loop
    help_record_count := help_record_count + 1;
--
    htp.tableRowOpen( 'left', 'top', cattributes=>'BORDER=1' );
    htp.tableData( replace(help_text_record.help_text,chr(10),'<br>') );
    htp.tableRowClose;
  end loop;
--
  if help_record_count = 0 then
    htp.tableRowOpen( 'left', 'top', cattributes=>'BORDER=1' );
    htp.tableData( htf.bold( 'No Help defined for ' || to_char(display_help.help_text_id)) );
    htp.tableRowClose;
  end if;
```

```
--
  htp.tableRowOpen( 'left', 'top', cattributes=>'BORDER=1' );
  htp.tableData( ' ' );
  htp.tableRowClose;
--
  htp.tableRowOpen( 'left', 'top', cattributes=>'BORDER=1' );
  htp.tableData( htf.anchor2( curl=>'javascript:window.close()',
                              ctext=>'<img src="' || owa_util.get_owa_service_path ||
                                     'ask_images/ask_close_button.jpg" border="0" alt="Close this
                                     window.">' ) );
  htp.tableRowClose;
--
  htp.TableClose;
  htp.nl;
--
  ask_utility.page_footer( 'DISPLAY_HELP', ask_utility.version );
--
END display_help;
--
/*****************************************************************************************/
--
FUNCTION read_parameter( organization_id in number,
                         parameter in varchar2 )
         RETURN VARCHAR2
/*
 *   Read a parameter value from the parameer table.
 *
 *    Input:  organization_id
 *            parameter
 *
 *   Output:  parameter value (null if error)
 *
 */
IS
  cursor parameter_cursor is
         select parameter_text
           from ask_organization_parameters
           where organization_id = read_parameter.organization_id
             and parameter = upper(read_parameter.parameter);
  parameter_record parameter_cursor%ROWTYPE;
--
BEGIN
  if not parameter_cursor%isopen then
    open parameter_cursor;
  end if;
  fetch parameter_cursor into parameter_record;
  if parameter_cursor%notfound then
    close parameter_cursor;
    return null;
  else
```

```
    close parameter_cursor;
    return parameter_record.parameter_text;
  end if;
--
EXCEPTION
  when others then
    return null;
--
END read_parameter;
--
/*****************************************************************************/
--
FUNCTION check_parameter_flag( organization_id in number,
                               parameter in varchar2 )
         RETURN BOOLEAN
/*
 *   Check parameter flag.
 *
 *    Input:  organization_id
 *            parameter
 *
 *   Output:  TRUE if parameter = 'Y' or 'YES'
 *            otherwise FALSE
 *
 */
IS
  cursor parameter_cursor is
         select ask_utility.trim_string(parameter_text) parameter_text
           from ask_organization_parameters
           where organization_id = check_parameter_flag.organization_id
             and parameter = upper(check_parameter_flag.parameter);
  parameter_text ask_organization_parameters.parameter_text%type := null;
--
BEGIN
--
  FOR parameter_record IN parameter_cursor
  LOOP
    parameter_text := ltrim(ltrim(parameter_record.parameter_text,CHR(10)),CHR(13));
  END LOOP;
--
  if substr(parameter_text,1,1) = 'Y' then
    RETURN TRUE;
  else
    RETURN FALSE;
  end if;
--
EXCEPTION
  when others then
    return FALSE;
--
```

```
END check_parameter_flag;
--
/*****************************************************************************************/
--
FUNCTION is_number( numeric_string in varchar2 )
         RETURN BOOLEAN
/*
 *   Determine if the passed string will convert to a number without error.
 *
 *    Input:  numeric_string
 *
 *   Output:  boolean TRUE or FALSE
 *
 *   History:
 *
 *   20 Aug 1998  Jeunnette  Initial implementation.
 */
IS
  test_number number;
--
BEGIN
  test_number := to_number(numeric_string);
--
  return TRUE;
--
EXCEPTION
  when others then
    return FALSE;
--
END is_number;
--
/*****************************************************************************************/
--
FUNCTION ValidateDate( in_date in varchar2,
                       out_date out varchar2 )
         RETURN BOOLEAN
/*
 *   Check for a valid entered or generated date.
 *
 *   After an article in the July, 1998 edition of "Exploring Oracle" from The Cobb Group.
 *
 *    Input:  in_date - the date to validate
 *
 *   Output:  out_date - a real date
 *            Returns boolean TRUE if date valid.
 *
 *   History:
 *
 *   27 Aug 1998  Jeunnette  Initial implementation.
 *    7 Aug 2004  Jeunnette  Add 'YYYYMMDD' format.
```

```
 */
IS
  date_value varchar2(32) := ltrim(rtrim(upper(in_date)));
  date_format_valid exception;
--  pragma exception_init( date_format_valid, -20001,
--                         'Date Converted Successfully');
  last_day_of_the_week constant varchar2(32) := 'FRIDAY';
  first_day_of_the_week constant varchar2(32) := 'MONDAY';
--
  PROCEDURE ConvertToDate( date_char in varchar2, date_format in varchar2 )
  IS
  BEGIN
    select to_char(to_date( date_char, date_format ),'mm/dd/yyyy')
      into out_date
      from dual;
    raise date_format_valid;
  EXCEPTION
    when date_format_valid then
      raise;  -- propogate to the calling module
    when others then
      null;
  END;
--
BEGIN
--
--  handle standard abbreviations
--
  BEGIN
    if date_value = 'EW' then
      out_date := to_char( next_day( sysdate, last_day_of_the_week ), 'mm/dd/yyyy');
      return TRUE;
    elsif date_value = 'BW' then
      out_date := to_char( next_day( sysdate, first_day_of_the_week ), 'mm/dd/yyyy');
      return TRUE;
    elsif date_value = 'BM' then
      out_date := to_char( trunc( sysdate, 'MONTH' ), 'mm/dd/yyyy');
      return TRUE;
    elsif date_value = 'EM' then
      out_date := to_char( last_day( sysdate ), 'mm/dd/yyyy');
      return TRUE;
    elsif date_value = 'TODAY' then
      out_date := to_char( sysdate, 'mm/dd/yyyy');
      return TRUE;
    elsif date_value = 'YESTERDAY' then
      out_date := to_char( sysdate-1, 'mm/dd/yyyy');
      return TRUE;
    elsif date_value = 'TOMORROW' then
      out_date := to_char( sysdate+1, 'mm/dd/yyyy');
      return TRUE;
    else
```

```
      out_date := to_char( next_day( sysdate, date_value ), 'mm/dd/yyyy' );
      return TRUE;
    end if;
  EXCEPTION
    when others then null;
  END;
--
--  we must have a date so now we try to convert using various
--  format strings.
--
  ConvertToDate( date_value, 'MM/DD/YYYY' );
  ConvertToDate( date_value, 'MM-DD-YYYY' );
  ConvertToDate( date_value, 'DD/MON/YYYY' );
  ConvertToDate( date_value, 'DD-MON-YYYY' );
  ConvertToDate( date_value, 'YYYYMMDD' );
  ConvertToDate( date_value, 'MMDD' );
  ConvertToDate( date_value, 'MON' );
  ConvertToDate( date_value, 'MONTH' );
  ConvertToDate( date_value, 'DD' );
  ConvertToDate( date_value, 'MM/DD' );
  ConvertToDate( date_value, 'MM-DD' );
  ConvertToDate( date_value, 'MM/DD/RR' );
  ConvertToDate( date_value, 'MM-DD-RR' );
  ConvertToDate( date_value, 'MON/DD' );
  ConvertToDate( date_value, 'MON-DD' );
  ConvertToDate( date_value, 'MON/DD/YYYY' );
  ConvertToDate( date_value, 'MON-DD-YYYY' );
  ConvertToDate( date_value, 'MON/RR' );
  ConvertToDate( date_value, 'MON-RR' );
  ConvertToDate( date_value, 'DD/MON' );
  ConvertToDate( date_value, 'DD-MON' );
  ConvertToDate( date_value, 'DD/MON/RR' );
  ConvertToDate( date_value, 'DD-MON-RR' );
--
  return FALSE;  -- invalid date.
--
EXCEPTION
  when date_format_valid then
    return TRUE;
  when others then
    return FALSE;
--
END ValidateDate;
--
/*****************************************************************************/
--
FUNCTION expand_text( text_string in varchar2 )
         RETURN varchar2
/*
 *   Add spaces to a text string to   s p r e a d   it out.
```

```
 *
 *    Input:  text_string
 *
 *   Output:  varchar2
 *
 */
IS
--
  return_string  varchar2(2000) := null;
--
BEGIN
  for charasker_loop in 1..length(text_string)
  loop
    if charasker_loop >= 1000 then
      exit;
    end if;
--
    return_string := return_string || substr(text_string,charasker_loop,1) || ' ';
  end loop;
--
  return return_string;
--
EXCEPTION
  when others then
    return null;
--
END expand_text;
--
/*****************************************************************************/
--
FUNCTION trim_string( in_string in varchar2 )
         RETURN VARCHAR2
/*
 *   Trim off leading and trailing spaces, line feeds, and carriage returns.
 *
 *    Input:  in_string
 *
 *   Output:  cleaned up string
 *
 */
IS
  out_string        varchar2(4000) := null;
  in_string_length  binary_integer := 0;
  out_string_length binary_integer := 0;
--
BEGIN
  in_string_length := nvl(length(in_string),0);
  out_string_length := in_string_length;
--
  for charasker_loop in reverse 1..in_string_length
```

```
  loop
    if substr(in_string,charasker_loop,1) in (chr(10),chr(13),chr(32)) then
      out_string_length := charasker_loop - 1;
    else
      exit;  -- bail out of the loop if we hit a non-space or terminator charasker...
    end if;
  end loop;
--
  if out_string_length > 4000 then
    out_string_length := 4000;
  end if;
  out_string := substr(in_string,1,out_string_length);
  out_string_length := 0;
--
  for charasker_loop in 1..nvl(length(out_string),0)
  loop
    if substr(out_string,charasker_loop,1) in (chr(10),chr(13),chr(32)) then
      out_string_length := charasker_loop;
    else
      exit;  -- bail out of the loop if we hit a non-space or terminator charasker...
    end if;
  end loop;
--
  return substr(out_string,out_string_length+1);
--
EXCEPTION
  when others then
    return null;
--
END trim_string;
--
/*****************************************************************************/
--
FUNCTION next_sequence return number
/*
 *   Get the next sequence value from ask_sequence.
 *
 *    Input:  (none)
 *
 *   Output:  ask_sequence.nextval
 *
 */
IS
--
  next_value           number := 0;
--
BEGIN
--
  begin
    select ask_sequence.nextval
```

```
      into next_value
      from dual;
  exception
    when others then
      report_error( organization_id => 0,
                    routine_name => 'ask_utility.next_sequence' );
  end;
--
  return next_value;
--
END next_sequence;
--
/*****************************************************************************/
--
PROCEDURE send_mail( from_address in varchar2,
                     to_address   in varchar2,
                     cc_address   in varchar2,
                     subject      in varchar2,
                     message_text in varchar2 default null )
--
/*
 *   Send a simple, non-attached email message to a specified address.
 *
 *    Input:  from_address
 *            to_address
 *            cc_address
 *            subject
 *            message_text (up to 32767 bytes)
 *
 *   Output:  a SMTP mail message
 *
 *   History:
 *
 *   18 Jul 2002  Jeunnette  Initial implementation from a procedure
 *                           written by Dave Wotton.  See
 *                           http://home.clara.net/dwotton/dba/oracle_smtp.htm
 */
IS
--
  v_smtp_server      varchar2(100) := 'localhost';
--  This may need to be the IP address of a SMTP service if there is not a local
--  SMTP service running (particularly relevant to Windows 2000 servers).
  v_smtp_server_port number  := 25;
--
  crlf               varchar2(2):= chr(13) || chr(10);
--
  conn               UTL_SMTP.CONNECTION;
--
  mesg               varchar2(32767);
  mesg_len           number;
```

```
  max_size           number := 32767;
--
  mesg_too_long      exception;
  invalid_path       exception;
--
  mesg_length_exceeded boolean := false;
  comma              binary_integer := 0;
  next_comma         binary_integer := 0;
  moretodo           boolean := FALSE;
--
BEGIN
--
-- Open the SMTP connection ...
--
  conn:= utl_smtp.open_connection( v_smtp_server, v_smtp_server_port );
--
-- Initial handshaking ...
--
  utl_smtp.helo( conn, v_smtp_server );
  utl_smtp.mail( conn, from_address );
--
  comma := nvl(instr( to_address, ',' ),0);
  if comma = 0 then
    utl_smtp.rcpt( conn, to_address );
  else
    moretodo := TRUE;
    comma := 0;
    while moretodo
    loop
      next_comma := nvl(instr(to_address,',',comma+1),0);
      if next_comma = 0 then
        next_comma := nvl(length(to_address),0) + 1;
        moretodo := FALSE;
      end if;
      utl_smtp.rcpt( conn, substr(to_address,comma+1,next_comma-comma-1) );
      comma := next_comma;
    end loop;
  end if;
--
  if cc_address is not null then
    comma := nvl(instr( cc_address, ',' ),0);
    if comma = 0 then
--      utl_smtp.rcpt( conn, 'cc:' || cc_address );
      utl_smtp.rcpt( conn, cc_address );
    else
      moretodo := TRUE;
      comma := 0;
      while moretodo
      loop
        next_comma := nvl(instr(cc_address,',',comma+1),0);
```

```
        if next_comma = 0 then
          next_comma := nvl(length(cc_address),0) + 1;
          moretodo := FALSE;
        end if;
--      utl_smtp.rcpt( conn, 'cc:' || substr(cc_address,comma+1,next_comma-comma-1) );
        utl_smtp.rcpt( conn, substr(cc_address,comma+1,next_comma-comma-1) );
        comma := next_comma;
      end loop;
    end if;
  end if;
--
  utl_smtp.open_data ( conn );
--
-- build the start of the mail message ...
--
  mesg:= 'Date: ' || TO_CHAR( SYSDATE, 'dd Mon yy hh24:mi:ss' ) || crlf ||
         'From: ' || from_address || crlf ||
         'Subject: ' || subject || crlf ||
         'To: ' || to_address || crlf ||
         'Cc: ' || cc_address || crlf ||
         '' || crlf;
  mesg_len := nvl(length(mesg),0);
  if mesg_len + nvl(length(message_text),0)+2 > max_size then
     mesg_length_exceeded := true;
  end if;
  mesg := mesg || message_text || crlf ;
--
  utl_smtp.write_data ( conn, mesg );
--
-- and close the SMTP connection  ...
--
  utl_smtp.close_data( conn );
--
  utl_smtp.quit( conn );
--
--
EXCEPTION
  when others then
    ask_utility.report_error( organization_id => 0,
                              routine_name => 'ask_utility.send_mail',
                              context => 'Unexpected error in ask_utility.send_mail ' ||
                                         ' to=' || to_address ||
                                         ' subject=' || subject ||
                                         ' message=' || mesg );
--
END send_mail;
--
/*****************************************************************************/
--
FUNCTION dbms_encrypt( input_string in varchar2 )
```

```
         RETURN VARCHAR2
/*
 *   Encrypt a string using the dbms_obsfuscation_tookkit.
 *
 *    Input:  input_string
 *
 *   Output:  encrypted string (hex)
 *
 */
IS
  key_string varchar2(8) := 'ASK ask ';
  encrypted_string  varchar2(2048);
  adjusted_string varchar2(2048);
  adjusted_string_length binary_integer := 0;
--
BEGIN
  adjusted_string_length := trunc(length(dbms_encrypt.input_string)/8)*8;
  if adjusted_string_length < length(dbms_encrypt.input_string) then
    adjusted_string_length := adjusted_string_length + 8;
  end if;
  adjusted_string := substr(dbms_encrypt.input_string||'        ',1,adjusted_string_length);
--
  dbms_obfuscation_toolkit.DESEncrypt( input_string=>adjusted_string,
                                       key_string=>key_string,
                                       encrypted_string=>encrypted_string );
--
  return rawtohex(utl_raw.cast_to_raw(encrypted_string));
--  return encrypted_string;
--
EXCEPTION
  when others then
    return null;
--
END dbms_encrypt;
--
/*******************************************************************************/
--
FUNCTION dbms_decrypt( input_string in varchar2 )
         RETURN VARCHAR2
/*
 *   Decrypt a string using the dbms_obsfuscation_tookkit.
 *
 *    Input:  input_string
 *
 *   Output:  decrypted string
 *
 */
IS
  key_string varchar2(8) := 'ASK ask ';
  decrypted_string  varchar2(2048);
```

```
  adjusted_string varchar2(2048);
  adjusted_string_length binary_integer := 0;
BEGIN
  adjusted_string := utl_raw.cast_to_varchar2(hextoraw(dbms_decrypt.input_string));
  dbms_obfuscation_toolkit.DESDecrypt( input_string=>adjusted_string,
                                       key_string=>key_string,
                                       decrypted_string=>decrypted_string );
--
  return decrypted_string;
--
EXCEPTION
  when others then
    return null;
--
END dbms_decrypt;
--
/*****************************************************************************************/
/*****************************************************************************************/
--
END ask_utility;
/
show errors
```